Modern Optimization Methods for Big Data Problems

MATH11146
The University of Edinburgh

Peter Richtárik

Week 3
Randomized Coordinate Descent With Arbitrary Sampling
January 27, 2016

The Problem

We first consider the following problem:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x = (x_1, \dots, x_n) \in \mathbb{R}^n \end{aligned} \tag{1}$$

We will assume that f is:

- **"smooth"** (will be made precise later)
- **strongly convex** (will be made precise later)

So, this is unconstrained minimization of a smooth convex function.

Randomized Coordinate Descent with Arbitrary Sampling

NSync Algorithm (R. and Takáč 2014, [4])

Input: initial point $x_0 \in \mathbb{R}^n$
subset probabilities $\{p_S\}$ for each $S \subseteq [n] \overset{\text{def}}{=} \{1, 2, \dots, n\}$
stepsize parameters $v_1, \dots, v_n > 0$

for $k = 0, 1, 2, \dots$ **do**

a) **Select a random set of coordinates** $S_k \subseteq [n]$ following the law

$$\mathbf{P}(S_k = S) = p_S, \qquad S \subseteq [n]$$

b) **Update (possibly in parallel) selected coordinates:**

$$x_{k+1} = x_k - \sum_{i \in S_k} \frac{1}{v_i} (e_i^T \nabla f(x_k)) e_i$$

(e_i is the ith unit coordinate vector)

end for

Remark: The **NSync algorithm** was introduced in 2013. The first coordinate descent algorithm using **arbitrary sampling.**

Two More Ways of Writing the Update Step

1. **Coordinate-by-coordinate:**

$$x_i^{k+1} = \begin{cases} x_i^k, & i \notin S_k, \\ x_i^k - \frac{1}{v_i} (\nabla f(x_k))_i, & i \in S_k. \end{cases}$$

2. **Via projection to a subset of blocks:** If for $h \in \mathbb{R}^n$ and $S \subseteq [n]$ we write

$$h_S \overset{\text{def}}{=} \sum_{i \in S} h_i e_i, \qquad (2)$$

then

$$x^{k+1} = x^k + h_{S_k} \qquad \text{for} \qquad h = -(\mathrm{Diag}(v))^{-1} \nabla f(x_k). \qquad (3)$$

Depending on context, we shall interchangeably denote the ith partial derivative of f at x by

$$\nabla_i f(x) = e_i^T \nabla f(x) = (\nabla f(x))_i$$

Samplings

Definition 1 (Sampling)

By the name **sampling** we refer to a set valued random mapping with values being subsets of $[n] = \{1, 2, \ldots, n\}$. For sampling $\hat{S}$ we define the **probability vector** $p = (p_1, \ldots, p_n)^T$ by

$$p_i = \mathbf{P}(i \in \hat{S}) \tag{4}$$

We say that $\hat{S}$ is **proper**, if $p_i > 0$ for all i.

- A sampling $\hat{S}$ is uniquely characterized by the **probability mass function**

$$p_S \overset{\text{def}}{=} \mathbf{P}(\hat{S} = S), \quad S \subseteq [n]; \tag{5}$$

  that is, by assigning probabilities to all subsets of $[n]$.

- Later on it will be useful to also consider the **probability matrix** $P = P(\hat{S}) = (p_{ij})$ given by

$$p_{ij} \overset{\text{def}}{=} \mathbf{P}(i \in \hat{S}, j \in \hat{S}) = \sum_{S:\{i,j\}\subseteq S} p_S. \tag{6}$$

Samplings: A Basic Identity

Lemma 2 ([3])

For any sampling $\hat{S}$ we have

$$\sum_{i=1}^{n} p_i = \mathbf{E}[|\hat{S}|]. \tag{7}$$

Proof.

$$\sum_{i=1}^{n} p_i \overset{(4)+(5)}{=} \sum_{i=1}^{n} \sum_{S\subseteq[n]:i\in S} p_S = \sum_{S\subseteq[n]} \sum_{i:i\in S} p_S = \sum_{S\subseteq[n]} p_S|S| = \mathbf{E}[|\hat{S}|].$$

□

Sampling Zoo - Part I

Why consider different samplings?

1. **Basic Considerations.** It is important that each block i has a positive probability of being chosen, otherwise NSync will not be able to update some blocks and hence will not converge to optimum. For technical/sanity reasons, we define:
    - **Proper sampling.** $p_i = \mathbf{P}(i \in \hat{S}) > 0$ for all $i \in [n]$
    - **Nil sampling:** $\mathbf{P}(\hat{S} = \emptyset) = 1$
    - **Vacuous sampling:** $\mathbf{P}(\hat{S} = \emptyset) > 0$
2. **Parallelism.** Choice of sampling affects the level of parallelism:
    - $\mathbf{E}[|\hat{S}|]$ is the average number of updates performed in parallel in one iteration; and is hence closely related to the number of iterations.
    - **serial sampling:** picks one block:

    $$\mathbf{P}(|\hat{S}| = 1) = 1$$

    We call this sampling serial although nothing prevents us from computing the actual update to the block, and/or to apply he update in parallel.

Sampling Zoo - Part II

- **fully parallel sampling:** always picks all blocks:

$$\mathbf{P}(\hat{S} = \{1, 2, \dots, n\}) = 1$$

3. **Processor reliability.** Sampling may be induced/informed by the computing environment:
    - **Reliable/dedicated processors.** If one has reliable processors, it is sensible to choose sampling $\hat{S}$ such that $\mathbf{P}(|\hat{S}| = \tau) = 1$ for some τ related to the number of processors.
    - **Unreliable processors.** If processors given a computing task are busy or unreliable, they return answer later or not at all - it is then sensible to ignore such updates and move on. This then means that $|\hat{S}|$ varies from iteration to iteration.
4. **Distributed computing.** In a distributed computing environment it is sensible:
    - to allow each compute node as much autonomy as possible so as to **minimize communication cost**,
    - to make sure **all nodes are busy** at all times

Sampling Zoo - Part III

This suggests a strategy where the set of blocks is partitioned, with each node owning a partition, and independently picking a "chunky" subset of blocks at each iteration it will update, ideally from local information.

5. **Uniformity.** It may or may not make sense to update some blocks more often than others:
   - **uniform samplings:**

     $$\mathbf{P}(i \in \hat{S}) = \mathbf{P}(j \in \hat{S}) \quad \text{for all} \quad i, j \in [n]$$

     - **doubly uniform (DU):** These are samplings characterized by:

       $$|S'| = |S''| \;\Rightarrow\; \mathbf{P}(\hat{S} = S') = \mathbf{P}(\hat{S} = S'') \quad \text{for all} \quad S', S'' \subseteq [n]$$

     - **τ-nice:** DU sampling with the additional property that

       $$\mathbf{P}(|\hat{S}| = \tau) = 1$$

     - **distributed τ-nice:** will define later
     - **independent sampling:** union of independent uniform serial samplings
   - **nonuniform samplings**

Sampling Zoo - Part IV

6. **Complexity of generating a sampling.** Some samplings are computationally more efficient to generate than others: the potential benefits of a sampling may be completely ruined by the difficulty to generate sets according to the sampling's distribution.
   - a τ-nice sampling can be well approximated by an independent sampling, which is easy to generate...
   - in general, many samplings will be hard to generate

Assumption: Strong convexity

Assumption 1 (Strong convexity)

Function f is differentiable and λ-strongly convex (with $\lambda > 0$) with respect to the standard Euclidean norm

$$\|h\| \stackrel{def}{=} \left(\sum_{i=1}^{n} h_i^2\right)^{1/2}.$$

That is, we assume that for all $x, h \in \mathbb{R}^n$,

$$f(x+h) \geq f(x) + \langle \nabla f(x), h\rangle + \frac{\lambda}{2}\|h\|^2. \tag{8}$$

Assumption: Expected Separable Overapproximation

Assumption 2 (ESO)

Assume $\hat{S}$ is proper and that for some vector of positive weights $v = (v_1, \dots, v_n)$ and all $x, h \in \mathbb{R}^n$,

$$\mathbf{E}[f(x + h_{\hat{S}})] \leq f(x) + \langle \nabla f(x), h\rangle_p + \tfrac{1}{2}\|h\|^2_{p \bullet v}. \tag{9}$$

Note that **the ESO parameters v, p depend on both f and $\hat{S}$.** *For simplicity, we will often instead of* (9) *use the compact notation*

$$(f, \hat{S}) \sim ESO(v).$$

Notation used above:

$$h_S \stackrel{\text{def}}{=} \sum_{i \in S} h_i e_i \in \mathbb{R}^n \qquad \text{(projection of } h \in \mathbb{R}^n \text{ onto coordinates } i \in S)$$

$$\langle g, h\rangle_p \stackrel{\text{def}}{=} \sum_{i=1}^{n} p_i g_i h_i \in \mathbb{R} \qquad \text{(weighted inner product)}$$

$$p \bullet v \stackrel{\text{def}}{=} (p_1 v_1, \dots, p_n v_n) \in \mathbb{R}^n \qquad \text{(Hadamard product)}$$

Assumption: Expected Separable Overapproximation

Here the ESO inequality again, now without the simplifying notation:

$$\underbrace{\mathbf{E}\left[f\left(x+\sum_{i\in\hat{S}} h_i e_i\right)\right]}_{\text{complicated}} \leq f(x) + \underbrace{\sum_{i=1}^{n} p_i \nabla_i f(x) h_i}_{\text{linear in } h} + \underbrace{\frac{1}{2}\sum_{i=1}^{n} p_i v_i h_i^2}_{\text{quadratic and separable in } h}$$

Complexity of NSync

Theorem 3 (R. and Takáč 2013, [4])

Let x^* be a minimizer of f. Let Assumptions 1 and 2 be satisfied for a proper sampling $\hat{S}$ (that is, $(f, \hat{S}) \sim ESO(v)$). Choose

- *starting point $x^0 \in \mathbb{R}^n$,*
- *error tolerance $0 < \epsilon < f(x^0) - f(x^*)$ and*
- *confidence level $0 < \rho < 1$.*

If $\{x^k\}$ are the random iterates generated by **NSync***, where the random sets S_k are iid following the distribution of $\hat{S}$, then*

$$\mathbf{K} \geq \frac{\boldsymbol{\Omega}}{\lambda} \log\left(\frac{f(x^0) - f(x^*)}{\epsilon\rho}\right) \quad \Rightarrow \quad \mathbf{P}(f(x^{\mathbf{K}}) - f(x^*) \leq \epsilon) \geq 1 - \rho, \quad (10)$$

where

$$\boldsymbol{\Omega} \overset{def}{=} \max_{i=1,\dots,n} \frac{v_i}{p_i} \geq \frac{\sum_{i=1}^n v_i}{\mathbf{E}[|\hat{S}|]}. \quad (11)$$

What does this mean?

- **Linear convergence.** NSync converges linearly (i.e., logarithmic dependence on ϵ)
- **High confidence is not a problem.** ρ appears inside the logarithm, so it easy to achieve high confidence (by running the method longer; there is no need to restart)
- **Focus on the leading term.** The leading term is Ω; and we have a closed-form expression for it in terms of
  - parameters $v_1, \ldots, v_n$ (which depend on f and $\hat{S}$)
  - parameters $p_1, \ldots, p_n$ (which depend on $\hat{S}$)
- **Parallelization speedup.** The lower bound suggests that *if it was the case that* the parameters v_i did not grow with increasing $\tau \overset{\text{def}}{=} \mathbf{E}[|\hat{S}|]$, then we could potentially be getting linear speedup in τ (average number of updates per iteration).
  - So we shall **study the dependence of v_i on τ** (this will depend on f and $\hat{S}$)
  - As we shall see, speedup is often guaranteed for **sparse or well-conditioned problems.**

Question: How to **design** sampling $\hat{S}$ so that $\mathbf{\Omega}$ is minimized?

Analysis of the Algorithm (Proof of Theorem 3)

Tool: Markov's Inequality

Theorem 4 (Markov's Inequality)

Let X be a nonnegative random variable. Then for any $\epsilon > 0$,

$$\mathbf{P}(X \geq \epsilon) \leq \frac{\mathbf{E}[X]}{\epsilon}.$$

Proof.

Let $1_{X \geq \epsilon}$ be the random variable which is equal to 1 if $X \geq \epsilon$ and 0 otherwise. Then

$$1_{X \geq \epsilon} \quad \leq \quad \frac{X}{\epsilon}.$$

By taking expectations of all terms, we obtain

$$\mathbf{P}(X \geq \epsilon) = \mathbf{E}\left[1_{X \geq \epsilon}\right] \leq \mathbf{E}\left[\frac{X}{\epsilon}\right] = \frac{\mathbf{E}[X]}{\epsilon}.$$

□

Tool: Tower Property of Expectations (Motivation)

Example 5

Consider discrete random variables $\mathbf{X}$ and $\mathbf{Y}$:

- $\mathbf{X}$ has 2 outcomes: x_1 and x_2
- $\mathbf{Y}$ has 3 outcomes: y_1, y_2 and y_3

Their joint probability mass function if given in this table:

	y_1	y_2	y_3	
x_1	0.05	0.20	0.03	**0.28**
x_2	0.25	0.30	0.17	**0.72**
	0.30	**0.50**	**0.20**	1

Obviously, $\mathbf{E}[\mathbf{X}] = \mathbf{0.28}x_1 + \mathbf{0.72}x_2$. But we can also write:

$$\begin{aligned} \mathbf{E}[\mathbf{X}] &= (0.05x_1 + 0.25x_2) + (0.20x_1 + 0.30x_2) + (0.03x_1 + 0.17x_2) \\ &= \underbrace{\mathbf{0.30}}_{\mathbf{P}(\mathbf{Y}=y_1)} \underbrace{\left(\tfrac{0.05}{0.30}x_1 + \tfrac{0.25}{0.30}x_2\right)}_{\mathbf{E}[\mathbf{X} \mid \mathbf{Y}=y_1]} + \mathbf{0.50}\left(\tfrac{0.20}{0.50}x_1 + \tfrac{0.30}{0.50}x_2\right) + \mathbf{0.20}\left(\tfrac{0.03}{0.20}x_1 + \tfrac{0.17}{0.20}x_2\right) \\ &= \mathbf{E}[\mathbf{E}[\mathbf{X} \mid \mathbf{Y}]]. \end{aligned}$$

Tower Property

Lemma 6 (Tower Property / Iterated Expectation)

For any random variables X and Y, we have $\mathbf{E}[X] = \mathbf{E}[\mathbf{E}[X \mid Y]]$.

Proof.

We shall only prove this for discrete random variables; the proof is more technical in the continuous case.

$$\begin{aligned}\mathbf{E}[X] = \sum_x x\mathbf{P}(X=x) &= \sum_x x \sum_y \mathbf{P}(X=x \,\&\, Y=y) \\ &= \sum_y \sum_x x\mathbf{P}(X=x \,\&\, Y=y) \\ &= \sum_y \sum_x \mathbf{P}(Y=y) x \frac{\mathbf{P}(X=x \,\&\, Y=y)}{\mathbf{P}(Y=y)} \\ &= \sum_y \mathbf{P}(Y=y) \underbrace{\sum_x x\mathbf{P}(X=x \mid Y=y)}_{\mathbf{E}[X \mid Y=y]} \\ &= \mathbf{E}[\mathbf{E}[X \mid Y]].\end{aligned}$$

□

Proof of Theorem 3 - Part I

- If we let $\mu \stackrel{\text{def}}{=} \lambda/\Omega$, then

$$\begin{aligned} f(x+h) &\stackrel{(8)}{\geq} f(x) + \langle \nabla f(x), h\rangle + \tfrac{\lambda}{2}\|h\|^2 \\ &\geq f(x) + \langle \nabla f(x), h\rangle + \tfrac{\mu}{2}\|h\|^2_{v\bullet p^{-1}}. \end{aligned} \tag{12}$$

  Indeed, one can easily verify that μ is defined to be the largest number for which

$$\lambda\|h\|^2 \geq \mu\|h\|^2_{v\bullet p^{-1}}$$

  holds for all h. Hence, f is μ-strongly convex with respect to the norm $\|\cdot\|_{v\bullet p^{-1}}$.
- Let x^* be a minimizer of f, i.e., an optimal solution of (1). Minimizing both sides of (12) in h, we get

$$\begin{aligned} f(x^*) - f(x) &\stackrel{(12)}{\geq} \min_{h\in\mathbb{R}^n} \langle \nabla f(x), h\rangle + \tfrac{\mu}{2}\|h\|^2_{v\bullet p^{-1}} \\ &= -\tfrac{1}{2\mu}\|\nabla f(x)\|^2_{p\bullet v^{-1}}. \end{aligned} \tag{13}$$

Proof of Theorem 3 - Part II

- Let $h^k \overset{\text{def}}{=} -v^{-1} \bullet \nabla f(x^k)$. Then in view of (3), we have $x^{k+1} = x^k + h^k_{S_k}$. Utilizing Assumption 2, we get

$$\begin{aligned} \mathbf{E}[f(x^{k+1}) \mid x^k] &= \mathbf{E}\left[f(x^k + h^k_{S_k}) \mid x^k\right] \\ &\overset{(9)}{\leq} f(x^k) + \langle \nabla f(x^k), h^k \rangle_p + \tfrac{1}{2}\|h^k\|^2_{p\bullet v} \\ &= f(x^k) - \tfrac{1}{2}\|\nabla f(x^k)\|^2_{p\bullet v^{-1}} \\ &\overset{(13)}{\leq} f(x^k) - \mu(f(x^k) - f(x^*)). \end{aligned}$$

- Taking expectations in the last inequality, using the **tower property,** and subtracting $f(x^*)$ from both sides of the inequality, we get

$$\mathbf{E}[f(x^{k+1}) - f(x^*)] \leq (1-\mu)\mathbf{E}[f(x^k) - f(x^*)].$$

Unrolling the recurrence, we get

$$\mathbf{E}[f(x^k) - f(x^*)] \leq (1-\mu)^k (f(x^0) - f(x^*)). \tag{14}$$

Proof of Theorem 3 - Part III

- Using **Markov's inequality,** (14) and the definition of K, we get

$$\begin{aligned} \mathbf{P}(f(x^K) - f(x^*) \geq \epsilon) &\leq \mathbf{E}[f(x^K) - f(x^*)]/\epsilon \\ &\overset{(14)}{\leq} (1-\mu)^K (f(x^0) - f(x^*))/\epsilon \overset{(10)}{\leq} \rho. \end{aligned}$$

- Finally, let us now establish the lower bound on Ω. Letting

$$\Delta \overset{\text{def}}{=} \left\{ p' \in \mathbb{R}^n : p' \geq 0, \sum_{i=1}^n p'_i = \mathbf{E}[|\hat{S}|] \right\},$$

we have

$$\Omega \overset{(11)}{=} \max_i \frac{v_i}{p_i} \overset{(7)}{\geq} \min_{p' \in \Delta} \max_i \frac{v_i}{p'_i} = \frac{1}{\mathbf{E}[|\hat{S}|]} \sum_{i=1}^n v_i,$$

where the last equality follows since optimal p'_i is proportional to v_i.

How to compute the ESO "stepsize" parameters $v_1, \ldots, v_n$?

$C^1(A)$ Functions

By definition, the ESO parameters v depend on both f and $\hat{S}$. This is also highlighted by the notation we use:

$$(f, \hat{S}) \sim ESO(v).$$

Hence, in order to compute these parameters, we need to specify f and $\hat{S}$. The following definition describes a wide class of functions, often appearing in computational practice, for which we will be able to do this.

Definition 7 ($C^1(A)$ functions)

Let $A \in \mathbb{R}^{m \times n}$. By $C^1(A)$ we denote the set of continuously differentiable functions $f : \mathbb{R}^n \to \mathbb{R}$ satisfying the following inequality for all $x, h \in \mathbb{R}^n$:

$$f(x+h) \leq f(x) + (\nabla f(x))^T h + \frac{1}{2} h^T A^T A h. \tag{15}$$

Example 8 (Least Squares)

The function $f(x) = \frac{1}{2}\|Ax - b\|^2$ satisfies (15) with an equality. Hence, $f \in C^1(A)$.

Are There More $C^1(A)$ Functions?

- Functions we wish to minimize in **machine learning** often have a **"finite sum" structure:**

$$f(x) \overset{\text{def}}{=} \sum_j \phi_j(M_j x), \tag{16}$$

  where $M_j \in \mathbb{R}^{m\times n}$ are **data matrices** and $\phi_j : \mathbb{R}^m \to \mathbb{R}$ are **loss functions.**
- The next result says that under certain assumptions on the loss functions, $f \in C^1(A)$ for some A, and describes what this A is.

Theorem 9 ([5])

Assume that for each j, function ϕ_j is γ_j-smooth:

$$\|\nabla\phi_j(s) - \nabla\phi_j(s')\| \le \gamma_j \|s - s'\|, \quad \textit{for all} \quad s, s' \in \mathbb{R}^m.$$

Then $f \in C^1(A)$, where A satisfies

$$A^T A = \sum_{j=1}^J \gamma_j M_j^T M_j.$$

Remark: The above theorem also says what $A^T A$ is. This is important, as will be clear from the next theorem.

ESO for $C_1(A)$ Functions and an Arbitrary Sampling

Theorem 10 ([5])

Assume $f \in C^1(A)$ for some real matrix $A \in \mathbb{R}^{m\times n}$, let $\hat{S}$ be an arbitrary sampling and let $P = P(\hat{S})$ be its probability matrix. If $v = (v_1, \dots, v_n)$ satisfies

$$P \bullet A^T A \preceq \mathrm{Diag}(p \bullet v),$$

then

$$(f, \hat{S}) \sim ESO(v).$$

Sampling Identity for a Quadratic

In in order to prove Theorem 10, we will need the following lemma.

Lemma 11 ([5])

Let G be any real $n \times n$ matrix and $\hat{S}$ an arbitrary sampling. Then for any $h \in \mathbb{R}^n$ we have

$$\mathbf{E}\left[h_{\hat{S}}^T G h_{\hat{S}}\right] = h^T \left(P(\hat{S}) \bullet G\right) h, \tag{17}$$

where $\bullet$ denotes the Hadamard (elementwise) product of matrices, and $P(\hat{S})$ is the probability matrix of $\hat{S}$.

Proof of Lemma 11

Proof.

Let 1_{ij} be the indicator random variable of the event $i \in \hat{S}$ & $j \in \hat{S}$:

$$1_{ij} = \begin{cases} 1 & \text{if } i \in \hat{S} \;\&\; j \in \hat{S}, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\mathbf{E}[1_{ij}] = \mathbf{P}(i \in \hat{S} \;\&\; j \in \hat{S}) = p_{ij}$. We now have

$$\begin{aligned} \mathbf{E}\left[h_{\hat{S}}^T G h_{\hat{S}}\right] &\overset{(2)}{=} \mathbf{E}\left[\sum_{i \in \hat{S}} \sum_{j \in \hat{S}} G_{ij} h_i h_j\right] = \mathbf{E}\left[\sum_{i=1}^n \sum_{j=1}^n 1_{ij} G_{ij} h_i h_j\right] \\ &= \sum_{i=1}^n \sum_{j=1}^n \mathbf{E}[1_{ij}] G_{ij} h_i h_j = \sum_{i=1}^n \sum_{j=1}^n p_{ij} G_{ij} h_i h_j \\ &= h^T \left(P(\hat{S}) \bullet G\right) h. \end{aligned}$$

$\square$

Proof of Theorem 10

Having established Lemma 11, we are now ready prove Theorem 10.

Proof.

Fixing any $x, h \in \mathbb{R}^n$, we have

$$\begin{aligned}
\mathbf{E}\left[f(x+h_{\hat{S}})\right] &\overset{(15)}{\leq} \mathbf{E}\left[f(x) + \langle \nabla f(x), h_{\hat{S}}\rangle + \tfrac{1}{2} h_{\hat{S}}^T A^T A h_{\hat{S}}\right] \\
&= f(x) + \langle \nabla f(x), h\rangle_p + \tfrac{1}{2}\mathbf{E}\left[h_{\hat{S}}^T A^T A h_{\hat{S}}\right] \\
&\overset{(\text{Lemma } 11)}{=} f(x) + \langle \nabla f(x), h\rangle_p + \tfrac{1}{2} h^T \left(P \bullet A^T A\right) h \\
&\leq f(x) + \langle \nabla f(x), h\rangle_p + \tfrac{1}{2} \underbrace{h^T \operatorname{Diag}(p \bullet v) h}_{=\|h\|_{p\bullet v}^2}.
\end{aligned}$$

$\square$

References I

[1] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341-362, 2012

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

[3] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 1–52, 2015

[4] Peter Richtárik and Martin Takáč. On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters*, 2015

[5] Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling II: expected separable overapproximation, *arXiv:1412.8063*, 2014