Modern Optimization Methods for Big Data Problems

MATH11146
The University of Edinburgh
Spring 2017

Peter Richtárik

Lecture 6

Randomized Subspace Descent for Convex Minimization: Algorithm and Convergence Theory

February 8, 2017

The Problem

We now 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 n is **very large** and that f is:

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

This is **unconstrained minimization of a smooth strongly convex function.**

Remark: We can write $x = \sum_{i=1}^n x_i e_i$, where e_i is the ith unit coordinate vector in $\mathbb{R}^n$. This is the vector whose ith coordinate is equal to 1, and all other coordinates are zero.

What's Ahead?

In Lecture 5, we described SDSA: stochastic dual subspace ascent. The method applies to an unconstrained convex quadratic minimization problem (equivalently: unconstrained concave quadratic maximization problem):

$$\max_{y \in \mathbb{R}^m} \underbrace{D(y)}_{\text{concave}} \quad \Leftrightarrow \quad \underbrace{\min_{y \in \mathbb{R}^m} -D(y)}_{\text{convex}}$$

Think of (1) as a generalized version of the dual problem from last lecture, with the following changes:

- We call the "dual" function variable x instead of y
- The dual variable lives in $\mathbb{R}^n$ instead of $\mathbb{R}^m$
- We call the "dual" objective f instead of $-D$
- The new objective is more general: **smooth** instead of just **quadratic**
- The new objective is less general: **strongly convex** as opposed to just **convex**
- From now on, we will denote iterates of an algorithm by superscripts as opposed to subscripts. That is, we will have a sequence $x^0, x^1, x^2, \dots$

Algorithmic Idea - I

How to solve the problem now that the objective is **not quadratic?**

Answer: A method of the same structure as SDSA, i.e.,

$$y_{k+1} = y_k + \mathbf{S}_k \lambda_k$$

will work, but we need to define the stepsize parameter λ_k differently as it is no longer easy to choose it greedily (i.e., to optimize for it by maximizing the dual objective). Using the new notation, the method will have the form

$$x^{k+1} = x^k + \mathbf{S}_k \lambda^k.$$

For a quadratic objective, all we had to do to decide on the stepsize was to solve a small (small if $\mathbf{S}_k$ had a few columns only) linear system. For a general convex function, we can't do this. In order to solve the problem, we will focus on a specific class of matrices $\mathbf{S}_k \sim \mathcal{D}$: **random column submatrices of the $n \times n$ identity matrix.** In this case, $\mathbf{S}_k \lambda^k$ is a linear

Algorithmic Idea - II

combination of a random subset of the coordinate (i.e., unit basis) vectors in $\mathbb{R}^n$. Therefore, the update can be written in the form

$$\mathbf{S}_k \lambda^k = \sum_{i \in S_k} h_i e_i,$$

where $\{h_i\}$ are the coefficients defining the linear combination.

Because of this, we do not have to talk about a random matrix $\mathbf{S}_k$. It is enough to talk about a random subset S_k of $[n] \overset{\text{def}}{=} \{1, 2, \ldots, n\}$. One leads to the other and vice versa.

The Plan

In summary, the method we will analyze in this lecture has the form

$$\boxed{x^{k+1} = x^k + \sum_{i \in S_k} h_i e_i}$$

where

- S_k is a random subset of $[n]$ (called **sampling**), and
- h_i is the stepsize in direction e_i

Main goals of this lecture:

- How to choose the random set S_k?
- How to choose the stepsizes h_i?
- Analyze the convergence of the method

Randomized Coordinate Descent with Arbitrary Sampling

NSync Algorithm (R. and Takáč 2013, [2])

Input: initial point $x^0 \in \mathbb{R}^n$

subset probabilities $\{p_S\}$ for each $S \subseteq [n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$

stepsize parameters $v_1, \ldots, v_n > 0$

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

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

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

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 in $\mathbb{R}^n$)

end for

Remark: The **NSync algorithm** was introduced in 2013. This is 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

$$h_S \stackrel{\text{def}}{=} \sum_{i \in S} h_i e_i, \tag{2}$$

then

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

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

$$\nabla_i f(x) = e_i^T \nabla f(x) = (\nabla f(x))_i \tag{4}$$

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

$$p_i = \mathrm{Prob}(i \in \hat{S}) \tag{5}$$

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**

$$p_S \stackrel{\text{def}}{=} \mathrm{Prob}(\hat{S} = S), \quad S \subseteq [n]; \tag{6}$$

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

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

$$p_{ij} \stackrel{\text{def}}{=} \mathrm{Prob}(i \in \hat{S}, j \in \hat{S}) = \sum_{S : \{i,j\} \subseteq S} p_S. \tag{7}$$

Samplings: A Basic Identity

Lemma 2 ([1])

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

$$\sum_{i=1}^{n} p_i = \mathrm{E}\left[|\hat{S}|\right]. \tag{8}$$

Proof.

$$\sum_{i=1}^{n} p_i \overset{(5)+(6)}{=} \sum_{i=1}^{n} \sum_{S\subseteq[n]:i\in S} p_S = \sum_{S\subseteq[n]} \sum_{i:i\in S} p_S = \sum_{S\subseteq[n]} p_S|S| = \mathrm{E}\left[|\hat{S}|\right].$$

□

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.** $p_i = \mathrm{Prob}(i \in \hat{S}) > 0$ for all $i \in [n]$
   - **Nil sampling:** $\mathrm{Prob}(\hat{S} = \emptyset) = 1$
   - **Vacuous sampling:** $\mathrm{Prob}(\hat{S} = \emptyset) > 0$
2. **Parallelism.** Choice of sampling affects the level of parallelism:
   - $\mathrm{E}\left[|\hat{S}|\right]$ 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:

$$\mathrm{Prob}(|\hat{S}| = 1) = 1$$

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:

$$\text{Prob}(\hat{S} = \{1, 2, \ldots, n\}) = 1$$

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 $\text{Prob}(|\hat{S}| = \tau) = 1$ 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:**

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

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

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

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

$$\text{Prob}(|\hat{S}| = \tau) = 1$$

     - **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

$$\|h\| \overset{def}{=} \sqrt{\sum_{i=1}^n h_i^2}.$$

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

$$f(x+h) \geq f(x) + \langle \nabla f(x), h \rangle + \frac{\lambda}{2}\|h\|^2. \tag{9}$$

Definition 3 (Weighted Euclidean norm)

We shall often need to work with **weighted Euclidean norms** defined as

$$\|h\|_w \overset{\text{def}}{=} \sqrt{\sum_{i=1}^n w_i h_i^2}, \tag{10}$$

where $w = (w_1, \ldots, w_n) \in \mathbb{R}^n$ is a vector of positive numbers ("weights"). The main reason for this is the ESO assumption we introduce on the next slide.

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 $x, h \in \mathbb{R}^n$,

$$\mathrm{E}\left[f(x + h_{\hat{S}})\right] \leq f(x) + \langle \nabla f(x), h \rangle_p + \tfrac{1}{2}\|h\|^2_{p \bullet v}. \tag{11}$$

Note that **the ESO parameters v, p depend on both f and $\hat{S}$.** *For simplicity, we will often instead of* (11) *use the compact notation*

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

Notation used above (allows us to write everything in a condensed form):

$$h_S \overset{\text{def}}{=} \sum_{i \in S} h_i e_i \in \mathbb{R}^n \tag{12}$$
("projection" of $h \in \mathbb{R}^n$ onto coordinates $i \in S$)

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

$$\langle g, h \rangle_p \overset{\text{def}}{=} \sum_{i=1}^n p_i g_i h_i \in \mathbb{R} \tag{13}$$
(weighted inner product with weights p)

Assumption: Expected Separable Overapproximation

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

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

Complexity of NSync

Theorem 4 (R. and Takáč 2013, [2])

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*

$$\mathbf{K} \geq \frac{\boldsymbol{\Omega}}{\lambda} \log \left(\frac{f(x^0) - f(x^*)}{\epsilon \rho} \right) \;\Rightarrow\; \mathrm{Prob}(f(x^{\mathbf{K}}) - f(x^*) \leq \epsilon) \geq 1 - \rho, \quad (14)$$

where

$$\boldsymbol{\Omega} \overset{def}{=} \max_{i=1,\ldots,n} \frac{v_i}{p_i} \geq \frac{\sum_{i=1}^n v_i}{\mathrm{E}\left[|\hat{S}|\right]}. \quad (15)$$

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 $\boldsymbol{\Omega}$; 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 $\tau \overset{\text{def}}{=} \mathrm{E}\left[|\hat{S}|\right]$, 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 $\boldsymbol{\Omega}$ is minimized?

Analysis of the Algorithm (Proof of Theorem 4)

Tool: Markov's Inequality

Theorem 5 (Markov's Inequality)

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

$$\text{Prob}(X \geq \epsilon) \leq \frac{\text{E}[X]}{\epsilon}.$$

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

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

□

Tool: Tower Property of Expectations (Motivation)

Example 6

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

- $\mathbf{X}$ has 2 outcomes: $\mathbf{x}_1$ and $\mathbf{x}_2$
- $\mathbf{Y}$ has 3 outcomes: $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{y}_3$

Their joint probability mass function if given in this table:

	$\mathbf{y}_1$	$\mathbf{y}_2$	$\mathbf{y}_3$	
$\mathbf{x}_1$	0.05	0.20	0.03	**0.28**
$\mathbf{x}_2$	0.25	0.30	0.17	**0.72**
	0.30	**0.50**	**0.20**	1

Obviously, $\mathrm{E}[\mathbf{X}] = 0.28\mathbf{x}_1 + 0.72\mathbf{x}_2$. But we can also write:

$$\begin{aligned}
\mathrm{E}[\mathbf{X}] &= (0.05\mathbf{x}_1 + 0.25\mathbf{x}_2) + (0.20\mathbf{x}_1 + 0.30\mathbf{x}_2) + (0.03\mathbf{x}_1 + 0.17\mathbf{x}_2) \\
&= \underbrace{0.30}_{\mathrm{Prob}(\mathbf{Y}=\mathbf{y}_1)} \underbrace{\left(\tfrac{0.05}{0.30}\mathbf{x}_1 + \tfrac{0.25}{0.30}\mathbf{x}_2\right)}_{\mathrm{E}[\mathbf{X} \mid \mathbf{Y}=\mathbf{y}_1]} + 0.50\left(\tfrac{0.20}{0.50}\mathbf{x}_1 + \tfrac{0.30}{0.50}\mathbf{x}_2\right) + 0.20\left(\tfrac{0.03}{0.20}\mathbf{x}_1 + \tfrac{0.17}{0.20}\mathbf{x}_2\right) \\
&= \mathrm{E}[\mathrm{E}[\mathbf{X} \mid \mathbf{Y}]] .
\end{aligned}$$

Tower Property

Lemma 7 (Tower Property / Iterated Expectation)

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

Proof.

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

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

Proof of Theorem 4 - Part I

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

$$\begin{aligned} f(x+h) &\overset{(9)}{\geq} f(x) + \langle \nabla f(x), h \rangle + \tfrac{\lambda}{2}\|h\|^2 \\ &\geq f(x) + \langle \nabla f(x), h \rangle + \tfrac{\mu}{2}\|h\|^2_{v \bullet p^{-1}}. \end{aligned} \tag{16}$$

  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 (16) in h, we get

$$\begin{aligned} f(x^*) - f(x) &\overset{(16)}{\geq} \min_{h \in \mathbb{R}^n} \langle \nabla f(x), h \rangle + \tfrac{\mu}{2}\|h\|^2_{v \bullet p^{-1}} \\ &= -\tfrac{1}{2\mu}\|\nabla f(x)\|^2_{p \bullet v^{-1}}. \end{aligned} \tag{17}$$

Proof of Theorem 4 - 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

$$\begin{aligned} \mathrm{E}\left[f(x^{k+1}) \mid x^k\right] &= \mathrm{E}\left[f(x^k + h^k_{S_k}) \mid x^k\right] \\ &\overset{(11)}{\leq} f(x^k) + \langle \nabla f(x^k), h^k \rangle_p + \tfrac{1}{2}\|h^k\|^2_{p \bullet v} \\ &= f(x^k) - \tfrac{1}{2}\|\nabla f(x^k)\|^2_{p \bullet v^{-1}} \\ &\overset{(17)}{\leq} f(x^k) - \mu(f(x^k) - f(x^*)). \end{aligned}$$

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

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

  Unrolling the recurrence, we get

$$\mathrm{E}\left[f(x^k) - f(x^*)\right] \leq (1-\mu)^k (f(x^0) - f(x^*)). \tag{18}$$

Proof of Theorem 4 - Part III

- Using **Markov's inequality,** (18) and the definition of K, we get

$$\begin{aligned}\mathrm{Prob}(f(x^K)-f(x^*)\geq\epsilon) &\leq \mathrm{E}\left[f(x^K)-f(x^*)\right]/\epsilon \\ &\overset{(18)}{\leq} (1-\mu)^K(f(x^0)-f(x^*))/\epsilon \overset{(14)}{\leq} \rho.\end{aligned}$$

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

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

we have

$$\Omega \overset{(15)}{=} \max_i \frac{v_i}{p_i} \overset{(8)}{\geq} \min_{p'\in\Delta}\max_i \frac{v_i}{p'_i} = \frac{1}{\mathrm{E}\left[|\hat{S}|\right]}\sum_{i=1}^n v_i,$$

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

References

[1] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1), 433–484, 2016 (*arXiv:1212.0873*, 2012)

[2] Peter Richtárik and Martin Takáč. On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters* 10(6), 1233–1243, 2016 (*arXiv:1310.3438*)

[3] Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling II: expected separable overapproximation, *Optimization Methods and Software* 31(5), 858–884, 2016 (*arXiv:1412.8063*)