Algorithmic Challenges of Big Data

Yurii Nesterov, CORE/INMA (UCL)

August 13-15, 2014

Max Planck Institute
Outline

Lecture 1: *Huge-Scale Optimization by Coordinate Updates*
- Problems with sparse data
- Implementation of coordinate moves
- Worst-case efficiency bounds
- Page-rank problem (Google problem)
- Numerical experiments

Lecture 2: *Subgradient methods for Huge-Scale Optimization Problems*

Lecture 3: *Finding primal-dual solutions of Huge-Scale Problems*

Reason for success: intelligent use of problem structure

Exercises 1,2: Training on implementation details
## Nonlinear Optimization: problems sizes

<table>
<thead>
<tr>
<th>Class</th>
<th>Operations</th>
<th>Dimension</th>
<th>Iter.Cost</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small-size</td>
<td>All</td>
<td>$10^0 - 10^2$</td>
<td>$n^4 \rightarrow n^3$</td>
<td>Kilobyte: $10^3$</td>
</tr>
<tr>
<td>Medium-size</td>
<td>$A^{-1}$</td>
<td>$10^3 - 10^4$</td>
<td>$n^3 \rightarrow n^2$</td>
<td>Megabyte: $10^6$</td>
</tr>
<tr>
<td>Large-scale</td>
<td>$Ax$</td>
<td>$10^5 - 10^7$</td>
<td>$n^2 \rightarrow n$</td>
<td>Gigabyte: $10^9$</td>
</tr>
<tr>
<td>Huge-scale</td>
<td>$x + y$</td>
<td>$10^8 - 10^{12}$</td>
<td>$n \rightarrow \log n$</td>
<td>Terabyte: $10^{12}$</td>
</tr>
</tbody>
</table>

### Sources of Huge-Scale problems

- Internet (New)
- Telecommunications (New)
- Finite-element schemes (Old)
- PDE, Weather prediction (Old)

### Main hope: Sparsity.
Our plans for today

- Take a very old optimization method.
- Explain why it is very bad.
- Prove that (sometimes) it is very good.
- Check this by numerical experiments.

**NB:** This will work for two other lectures too.
Very old optimization idea: Coordinate Search

**Problem:** \[ \min_{x \in \mathbb{R}^n} f(x) \quad (f \text{ is convex and differentiable}). \]

**Coordinate relaxation algorithm**

For \( k \geq 0 \) iterate

1. Choose active coordinate \( i_k \).
2. Update \( x_{k+1} = x_k - h_k \nabla_{i_k} f(x_k)e_{i_k} \) ensuring \( f(x_{k+1}) \leq f(x_k) \).
   \((e_i \text{ is } i\text{th coordinate vector in } \mathbb{R}^n.\))

**Main advantage:** Very simple implementation.
Possible strategies

1. Cyclic moves. (Difficult to analyze.)
2. Random choice of coordinate (Why?)
3. Choose coordinate with the maximal directional derivative.

**Complexity estimate:** assume

\[ \| \nabla f(x) - \nabla f(y) \| \leq L \| x - y \|, \quad x, y \in \mathbb{R}^n. \]

Let us choose \( h_k = \frac{1}{L} \). Then

\[
f(x_k) - f(x_{k+1}) \geq \frac{1}{2L} |\nabla_{i_k} f(x_k)|^2 \geq \frac{1}{2nL} \| \nabla f(x_k) \|^2
\]

\[
\geq \frac{1}{2nLR^2} (f(x_k) - f^*)^2.
\]

Hence, \( f(x_k) - f^* \leq \frac{2nLR^2}{k} \), \( k \geq 1 \). (For pure GM, drop \( n \).)

This was the only known theoretical result known for CDM!
Theoretical justification:

- Complexity bounds are not known for the most of the schemes.
- The only justified scheme needs computation of the whole gradient. (Why don’t use GM?)

Computational complexity:

- **Fast differentiation:** if function is defined by a sequence of operations, then $C(\nabla f) \leq 4C(f)$.
- Can we do anything without computing the function’s values?

Result: CDM were almost out of computational practice during decades.
Let $E \in R^{n\times n}$ be an incidence matrix of a graph. Denote $e = (1, \ldots, 1)^T$ and

$$E = E \cdot \text{diag} \left( (E^T e)^{-1} \right).$$

Thus, $E^T e = e$. Our problem is as follows:

Find $x^* \geq 0 : \quad E x^* = x^*$.

**Optimization formulation:**

$$f(x) \overset{\text{def}}{=} \frac{1}{2} \| E x - x \|^2 + \gamma \left[ \langle e, x \rangle - 1 \right]^2 \rightarrow \min_{x \in R^n}$$
Huge-scale problems

Main features

- The size is very big \((n \geq 10^7)\).
- The data is distributed in space.
- The requested parts of data are not always available.
- The data is changing in time.

Consequences

Simplest operations are expensive or infeasible:

- Update of the full vector of variables.
- Matrix-vector multiplication.
- Computation of the objective function’s value, etc.
Structure of the Google Problem

Let us look at the gradient of the objective:

$$\nabla_i f(x) = \langle a_i, g(x) \rangle + \gamma[\langle e, x \rangle - 1], \; i = 1, \ldots, n,$$

$$g(x) = \bar{E}x - x \in \mathbb{R}^n, \; (\bar{E} = (a_1, \ldots, a_n)).$$

Main observations:

- The coordinate move $x_+ = x - h_i \nabla_i f(x)e_i$ needs $O(p_i)$ a.o. ($p_i$ is the number of nonzero elements in $a_i$.)
- $d_i \overset{\text{def}}{=} \text{diag} \left( \nabla^2 f \overset{\text{def}}{=} \bar{E}^T\bar{E} + \gamma ee^T \right)_i = \gamma + \frac{1}{p_i}$ are available. We can use them for choosing the step sizes ($h_i = \frac{1}{d_i}$).

Reasonable coordinate choice strategy? Random!
Random coordinate descent methods (RCDM)

\[
\min_{x \in \mathbb{R}^N} f(x), \quad (f \text{ is convex and differentiable})
\]

Let us decompose the space: \( R^N = \bigotimes_{i=1}^{n} R^{n_i} \), \( N = \sum_{i=1}^{n} n_i \),

\[
I_N = (U_1, \ldots, U_n) \in R^{N \times N}, \quad U_i \in R^{N \times n_i},
\]

\[
x = (x^{(1)}, \ldots, x^{(n)})^T = \sum_{i=1}^{n} U_i x^{(i)}, \quad x^{(i)} \in R^{n_i}.
\]

**Partial gradient** of \( f(x) \) in \( x^{(i)} \) is \( f'_i(x) = U_i^T \nabla f(x) \in R^{n_i} \).

For \( R^{n_i} \), we fix norms \( \|x\|_{(i)} \), \( \|s\|_{(i)}^\ast = \max_{\|h\|_{(i)} = 1} \langle s, h \rangle \).

If \( h(s) \) is the optimal solution, then \( s^* \overset{\text{def}}{=} \|s\|_{(i)}^\ast \cdot h(s) \).
Main Assumption:

\[ \| f_i'(x + U_i h_i) - f_i'(x) \|_{(i)}^* \leq L_i \| h_i \|_{(i)}, \quad h_i \in \mathbb{R}^{n_i}, \ i = 1, \ldots, n. \]

Then

\[ f(x + U_i h_i) \leq f(x) + \langle f_i'(x), h_i \rangle + \frac{L_i}{2} \| h_i \|_{(i)}^2, \quad x \in \mathbb{R}^N, \ h_i \in \mathbb{R}^{n_i}. \]

Define the coordinate steps: \( T_i(x) \overset{\text{def}}{=} x - \frac{1}{L_i} U_i f_i'(x)*. \) Then,

\[ f(x) - f(T_i(x)) \geq \frac{1}{2L_i} \left( \| f_i'(x) \|_{(i)}^* \right)^2, \quad i = 1, \ldots, n. \]

Proof: Minimize the upper bound.
Random coordinate choice

We need a special random counter $R_\alpha$, $\alpha \in [0, 1]$:

$$\text{Prob} [i] = p^{(i)}_\alpha = L^\alpha_i \cdot \left[ \sum_{j=1}^{n} L^\alpha_j \right]^{-1}, \quad i = 1, \ldots, n.$$  

Note: $R_0$ generates uniform distribution.

Method $RCDM(\alpha, x_0)$

For $k \geq 0$ iterate:

1) Choose $i_k = R_\alpha$.

2) Update $x_{k+1} = T_{i_k}(x_k)$.
Complexity bounds for RCDM

We need to introduce the following norms for $x, g \in \mathbb{R}^N$:

$$\|x\|_\alpha = \left[ \sum_{i=1}^{n} L^\alpha (\|x(i)\|_2^2) \right]^{1/2}, \quad \|g\|_\alpha^* = \left[ \sum_{i=1}^{n} \frac{1}{L^\alpha} \left( \|g(i)\|_\alpha^* \right)^2 \right]^{1/2}.$$ 

After $k$ iterations, $RCDM(\alpha, x_0)$ generates random output $x_k$, which depends on $\xi_k = \{i_0, \ldots, i_k\}$. Denote $\phi_k = E_{\xi_{k-1}} f(x_k)$.

**Theorem.** For any $k \geq 1$ we have

$$\phi_k - f^* \leq \frac{2}{k} \cdot \left[ \sum_{j=1}^{n} L_j^\alpha \right] \cdot R_{1-\alpha}^2(x_0),$$

where $R_\beta(x_0) = \max_x \left\{ \max_{x_* \in X^*} \|x - x_*\|_\beta : f(x) \leq f(x_0) \right\}$. 
1. $\alpha = 0$. Then $S_0 = n$, and we get

$$\phi_k - f^* \leq \frac{2n}{k} \cdot R_1^2(x_0).$$

Note

- We use the metric $\|x\|_1^2 = \sum_{i=1}^{n} L_i \|x^{(i)}\|_2^2$.

- For matrix with diagonal $\{L_i\}_{i=1}^{n}$ its norm can reach $n$.

- Hence, for GM we can guarantee the same bound. But its cost of iteration is much higher!
2. $\alpha = \frac{1}{2}$. Let $n_i = 1, i = 1, \ldots, n$. Denote

$$D_\infty(x_0) = \max_x \left\{ \max_{y \in X^*} \max_{1 \leq i \leq n} |x^{(i)} - y^{(i)}| : f(x) \leq f(x_0) \right\}.$$ 

Then, $R_{1/2}^2(x_0) \leq S_{1/2}D_\infty^2(x_0)$, and we obtain

$$\phi_k - f^* \leq \frac{2}{k} \cdot \left[ \sum_{i=1}^{n} L_i^{1/2} \right]^2 \cdot D_\infty^2(x_0).$$

Note:

- For the first order methods, the worst-case complexity of minimizing over a box depends on $n$.
- Since $S_{1/2}$ can be bounded, RCDM can be applied in situations where the usual GM fail.
3. $\alpha = 1$. Let all norms $\| \cdot \|_{(i)}$ are standard Euclidean. Then $R_0(x_0)$ is the size of the initial level set, and

$$
\phi_k - f^* \leq \frac{2}{k} \cdot \left[ \sum_{i=1}^{n} L_i \right] \cdot R_0^2(x_0) \equiv \frac{2n}{k} \cdot \left[ \frac{1}{n} \sum_{i=1}^{n} L_i \right] \cdot R_0^2(x_0).
$$

Rate of convergence of GM can be estimated as

$$
f(x_k) - f^* \leq \frac{\gamma}{k} R_0^2(x_0),
$$

where $\gamma$ satisfies condition $f''(x) \preceq \gamma \cdot I$, $x \in R^N$.

**Note:** maximal eigenvalue of symmetric matrix can reach its trace.

In the worst case, the rate of convergence of GM is the same as that of $RCDM$. 
Minimizing strongly convex functions

**Theorem.** Let \( f(x) \) be strongly convex with respect to \( \| \cdot \|_{1-\alpha} \) with convexity parameter \( \sigma_{1-\alpha} > 0 \). Then, for \( \{x_k\} \) generated by \( RCDM(\alpha, x_0) \) we have

\[
\phi_k - \phi^* \leq \left( 1 - \frac{\sigma_{1-\alpha}}{S_{\alpha}} \right)^k (f(x_0) - f^*).
\]

**Proof:** Let \( x_k \) be generated by \( RCDM \) after \( k \) iterations. Let us estimate the expected result of the next iteration.

\[
f(x_k) - E_{i_k}(f(x_{k+1})) = \sum_{i=1}^{n} p_{\alpha}^{(i)} \cdot [f(x_k) - f(T_i(x_k))]
\]

\[
\geq \sum_{i=1}^{n} \frac{p_{\alpha}^{(i)}}{2L_i} \left( \| f'_i(x_k) \|_{(i)}^* \right)^2 = \frac{1}{2S_{\alpha}} (\| f'(x_k) \|_{1-\alpha}^* )^2
\]

\[
\geq \frac{\sigma_{1-\alpha}}{S_{\alpha}} (f(x_k) - f^*).
\]

It remains to compute expectation in \( \xi_{k-1} \).
Confidence level of the answers

Note: We have proved that the expected values of random $f(x_k)$ are good.

*Can we guarantee anything after a single run?*

Confidence level: Probability $\beta \in (0, 1)$, that some statement about random output is correct.

Main tool: Markov inequality $(\xi, T > 0)$:

$$\text{Prob} [\xi \geq T] \leq \frac{E(\xi)}{T}.$$

Our situation:

$$\text{Prob} [f(x_k) - f^* \geq \epsilon] \leq \frac{1}{\epsilon} [\phi_k - f^*] \leq 1 - \beta.$$ We need $\phi_k - f^* \leq \epsilon \cdot (1 - \beta)$. Too expensive for $\beta \to 1$?
Consider $f_\mu(x) = f(x) + \frac{\mu}{2}\|x - x_0\|_1^2 - \alpha$. It is strongly convex. Therefore, we can obtain $\phi_k - f^* \leq \epsilon \cdot (1 - \beta)$ in

$$O\left(\frac{1}{\mu} S_\alpha \ln \frac{1}{\epsilon \cdot (1 - \beta)}\right)$$

iterations.

**Theorem.** Define $\alpha = 1$, $\mu = \frac{\epsilon}{4R_0^2(x_0)}$, and choose

$$k \geq 1 + \frac{8S_1 R_0^2(x_0)}{\epsilon} \left[ \ln \frac{2S_1 R_0^2(x_0)}{\epsilon} + \ln \frac{1}{1 - \beta} \right].$$

Let $x_k$ be generated by $RCDM(1, x_0)$ as applied to $f_\mu$. Then

$$\text{Prob} \left( f(x_k) - f^* \leq \epsilon \right) \geq \beta.$$

**Note:** $\beta = 1 - 10^{-p} \quad \Rightarrow \quad \ln 10^p = 2.3p.$
1. Problems with constraints:

\[
\min_{x \in Q} f(x),
\]

where \( Q = \bigotimes_{i=1}^{n} Q_i, Q_i \subseteq R^{n_i}, i = 1, \ldots, n, \) are closed and convex.

Define the constrained coordinate update:

\[
u^{(i)}(x) = \arg \min_{u^{(i)} \in Q_i} \left[ \left< f_i'(x), u^{(i)} - x^{(i)} \right> + \frac{L_i}{2} \| u^{(i)} - x^{(i)} \|_2^2 \right],
\]

\[
T_i(x) = x + U_i^T (u^{(i)} - x^{(i)}), \quad i = 1, \ldots, n.
\]

Then

\[
f(x) - f(T_i(x)) \geq \frac{L_i}{2} \| u^{(i)} - x^{(i)} \|_2^2, \quad i = 1, \ldots, n.
\]
For $k \geq 0$ iterate:

1) Choose randomly $i_k$ by uniform distribution on $\{1 \ldots n\}$.

2) Update $x_{k+1} = T_{i_k}(x_k)$.

**Theorem.** For any $k \geq 0$ we have

$$\phi_k - f^* \leq \frac{n}{n+k} \cdot \left[ \frac{1}{2} R_1^2(x_0) + f(x_0) - f^* \right].$$

If $f$ is strongly convex in $\| \cdot \|_1$ with constant $\sigma$, then

$$\phi_k - f^* \leq \left( 1 - \frac{2\sigma}{n(1+\sigma)} \right)^k \cdot \left( \frac{1}{2} R_1^2(x_0) + f(x_0) - f^* \right).$$
Implementation details: Random Counter

Given the values $L_i$, $i = 1, \ldots, n$, generate efficiently random $i \in \{1, \ldots, n\}$ with probabilities $\text{Prob} [i = k] = L_k / \sum_{j=1}^{n} L_j$.

**Solution:**

a) Trivial $\Rightarrow O(n)$ operations.

b) Assume $n = 2^p$. Define $p + 1$ vectors $S_k \in \mathbb{R}^{2^{p-k}}$, $k = 0, \ldots, p$:

$$S_0^{(i)} = L_i, \ i = 1, \ldots, n.$$ 

$$S_k^{(i)} = S_{k-1}^{(2i)} + S_{k-1}^{(2i-1)}, \ i = 1, \ldots, 2^{p-k}, \ k = 1, \ldots, p.$$ 

**Algorithm:** Make the choice in $p$ steps, from top to bottom.

- If the element $i$ of $S_k$ is chosen, then choose in $S_{k-1}$ either $2i$ or $2i - 1$ in accordance to probabilities $\frac{S_{k-1}^{(2i)}}{S_k^{(i)}}$ or $\frac{S_{k-1}^{(2i-1)}}{S_k^{(i)}}$.

**Difference:** for $n = 2^{20} > 10^6$ we have $p = \log_2 n = 20$. 

Yu. Nesterov

Huge-scale optimization by CD-methods 23/25
Numerical experiments: Google problem

\[ f(x) \overset{\text{def}}{=} \frac{1}{2} \| \bar{E}x - x \|^2 + \frac{\gamma}{2} [\langle e, x \rangle - 1]^2 \rightarrow \min_{x \in \mathbb{R}^n}, \]

where \( \gamma > 0 \) is a penalty parameter, the norm is Euclidean.

**Termination criterion:** \( \| \bar{E}x - x \|_2(2) \leq \epsilon \cdot \| x \|_2 \) with \( \epsilon = 0.01 \).

**Computer:** Notebook Pentium-4 1.6GHz.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( p )</th>
<th>( \gamma )</th>
<th>( k )</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>65536</td>
<td>10</td>
<td>( \frac{1}{n} )</td>
<td>47</td>
<td>7.41</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>( \frac{1}{\sqrt{n}} )</td>
<td>65</td>
<td>10.5</td>
</tr>
<tr>
<td>262144</td>
<td>10</td>
<td>( \frac{1}{n} )</td>
<td>47</td>
<td>42.7</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>( \frac{1}{\sqrt{n}} )</td>
<td>72</td>
<td>76.5</td>
</tr>
<tr>
<td>1048576</td>
<td>10</td>
<td>( \frac{1}{n} )</td>
<td>49</td>
<td>247</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>( \frac{1}{\sqrt{n}} )</td>
<td>82</td>
<td>486</td>
</tr>
</tbody>
</table>

**NB:** Moderate growth of computational time.
1. We presented a technique for solving huge-scale *smooth* optimization problems with simple constraints.

2. Data can be distributed in space.

3. Data can be changing in time.

**Next lecture:** Huge-scale *nonsmooth* optimization problems.