Proximal Stochastic Gradient Descent
Randomized Coordinate Gradient Descent

Pontus Giselsson

Learning goals

- Know the coordinate gradient and stochastic gradient methods
  - Understand when one or the other can/should be used
  - Understand conditions for convergence
  - Know how the analyses differ and what step-sizes work
  - Relate analysis to the one for proximal gradient method
  - Know result that convergence of stochastic methods is based on

Why stochastic methods?

Randomized selections

- Stochastic proximal gradient descent solves *finite sum* problems

$$\underset{x}{\text{minimize}}\, \tfrac{1}{N}\left(\sum_{i=1}^{N} f_i(x)\right) + g(x)$$

  where gradient is taken w.r.t. *randomly chosen* f_i instead of f

- Coordinate proximal gradient descent solves separable problems

$$\underset{x}{\text{minimize}}\, f(x) + \sum_{i=1}^{n} g_i(x_i)$$

  where one *randomly chosen* coordinate is updated every iteration

- Deterministic (cyclic) selection rules can also be used
- Random selection gives better convergence guarantees

Stochastic algorithm analysis

- Stochastic algorithms generate *realizations* of stochastic process
- Stochastic algorithm analysis:
  - (i) analyze the generated stochastic process of random variables
  - (ii) draw conclusion on (almost) all realizations
- More specifically:
  - (i) constructing (almost) supermartingale inequality for the algorithm
  - (ii) applying Robbins Siegmund supermartingale theorem
- Typically strong guarantees for (almost) all realizations

Robbins Siegmund supermartingale theorem

Suppose that:

(i) $(x_k)_{k\in\mathbb{N}}$: sequence of $\mathbb{R}^n$-valued random variables

(ii) $(v_k)_{k\in\mathbb{N}}$: sequence of $\mathbb{R}_{\geq 0}$-valued random variables

(iii) $(w_k)_{k\in\mathbb{N}}$: sequence of $\mathbb{R}_{\geq 0}$-valued random variables

(iv) $V : \mathbb{R}^n \to \mathbb{R}_{\geq c}$ is lower bounded function ($V(x) \geq c$) with $c \in \mathbb{R}$

(v) the following (almost) supermartingale inequality holds a.s. $\forall k$:

$$\mathbb{E}[V(x_{k+1})|\mathcal{F}_k] \leq V(x_k) + v_k - w_k,$$

where $\mathbb{E}$ conditioned on $\mathcal{F}_k$: "information known until iterate k"

Then, whenever $(v_k)_{k\in\mathbb{N}}$ is summable:

- $V(x_k)$ converges a.s. to a $\mathbb{R}_{\geq c}$-valued *random* variable
- $\mathbb{E}V(x_k)$ converges a.s. to a $\mathbb{R}_{\geq c}$-valued number
- $w_{k\in\mathbb{N}}$ is summable and $w_k \to 0$ as $k \to \infty$ a.s.

a.s. means almost surely; for "all" realizations (except 0-measure)

Stochastic Proximal Gradient Method

Proximal gradient method

- Proximal gradient method solves problems of the form

$$\underset{x}{\text{minimize}}\, f(x) + g(x)$$

  where (at least in our analysis)

  - $f : \mathbb{R}^n \to \mathbb{R}$ is β-smooth (not necessarily convex)
  - $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is closed convex

- For large problems, gradient is expensive to compute
  $\Rightarrow$ replace by unbiased stochastic approximation of gradient

Unbiased stochastic gradient approximation

- Stochastic gradient:
  - estimator $\widehat{\nabla} f(x)$ outputs $\mathbb{R}^n$-valued random variable
  - realization $\widetilde{\nabla} f(x) : \mathbb{R}^n \to \mathbb{R}^n$ outputs a realization in $\mathbb{R}^n$
- An unbiased stochastic gradient approximator $\widehat{\nabla} f$ satisfies

$$\mathbb{E}\widehat{\nabla} f(x) = \nabla f(x)$$

- If x is random variable (as in SGD) an unbiased estimator satisfies

$$\mathbb{E}[\widehat{\nabla} f(x)|x] = \nabla f(x)$$

Stochastic gradient descent (SGD)

- The following iteration generates $(x_k)_{k\in\mathbb{N}}$ of *random* variables:

$$x_{k+1} = \mathrm{prox}_{\gamma_k g}(x_k - \gamma_k \widehat{\nabla} f(x_k))$$

  since $\widehat{\nabla} f$ outputs random $\mathbb{R}^n$-valued variables

- Stochastic gradient descent finds a *realization* of this sequence:

$$x_{k+1} = \mathrm{prox}_{\gamma_k g}(x_k - \gamma_k \widetilde{\nabla} f(x_k))$$

  where $(x_k)_{k\in\mathbb{N}}$ here is a realization which is different every time

- Sloppy in notation for when x_k is *random variable* vs *realization*
- Efficient if realizations $\widetilde{\nabla} f$ much cheaper to evaluate than ∇f
- Analyze former and draw conclusions of (almost) all realizations

Stochastic gradients – Finite sum problems

- Consider *finite sum problems* of the form

$$\underset{x}{\text{minimize}}\, \underbrace{\tfrac{1}{N}\left(\sum_{i=1}^{N} f_i(x)\right)}_{f(x)} + g(x)$$

  where ($\frac{1}{N}$ is for convenience and)
  - all $f_i : \mathbb{R}^n \to \mathbb{R}$ are β_i-smooth (not necessarily convex)
  - $f : \mathbb{R}^n \to \mathbb{R}$ is β-smooth (not necessarily convex)
  - $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is closed convex
- Training problems of this form, where sum over training data
- Stochastic gradient: select f_i at random and take gradient step

Single function stochastic gradient

- Let I be a $\{1, \ldots, N\}$-valued random variable
- Let, as before, $\widehat{\nabla} f$ denote stochastic gradient estimator
- Realization: let i be drawn from probability distribution of I

$$\widetilde{\nabla} f(x) = \nabla f_i(x)$$

where we will use uniform probability distribution

$$p_i = p(I = i) = \tfrac{1}{N}$$

- Stochastic gradient is unbiased:

$$\mathbb{E}[\widehat{\nabla} f(x)|x] = \sum_{i=1}^{N} p_i \nabla f_i(x) = \tfrac{1}{N} \sum_{i=1}^{N} \nabla f_i(x) = \nabla f(x)$$

Mini-batch stochastic gradient

- Let $\mathcal{B}$ be set of K-sample mini-batches to choose from:
  - Example: 2-sample mini-batches and $N = 4$:

$$\mathcal{B} = \{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\}\}$$

  - Number of mini batches $\binom{N}{K}$, each item in $\binom{N-1}{K-1}$ batches
- Let $\mathbb{B}$ be $\mathcal{B}$-valued random variable
- Let, as before, $\widehat{\nabla} f$ denote stochastic gradient estimator
- Realization: let B be drawn from probability distribution of $\mathbb{B}$

$$\widetilde{\nabla} f(x) = \tfrac{1}{K} \sum_{i \in B} \nabla f_i(x)$$

where we will use uniform probability distribution

$$p_B = p(\mathbb{B} = B) = \tfrac{1}{|\mathcal{B}|}$$

- Stochastic gradient is unbiased:

$$\mathbb{E}\widehat{\nabla} f(x) = \tfrac{1}{\binom{N}{K}} \sum_{B \in \mathcal{B}} \tfrac{1}{K} \sum_{i \in B} \nabla f_i(x) = \tfrac{\binom{N-1}{K-1}}{\binom{N}{K}K} \sum_{i=1}^{N} \nabla f_i(x) = \tfrac{1}{N} \sum_{i=1}^{N} \nabla f_i(x) = \nabla f(x)$$

Stochastic gradient descent for finite sum problems

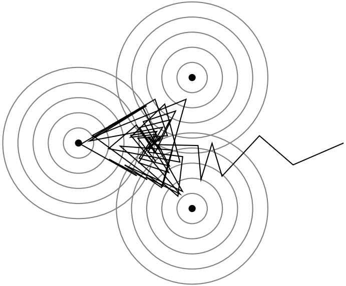
- The algorithm, choose $x_0 \in \mathbb{R}^n$ and iterate:
  1. Sample a mini-batch $B_k \in \mathcal{B}$ of indices uniformly (prob. $\frac{1}{|\mathcal{B}|}$)
  2. Run

$$x_{k+1} = \operatorname{prox}_{\gamma_k g}(x_k - \tfrac{\gamma_k}{|B_k|} \sum_{j \in B_k} \nabla f_j(x_k))$$

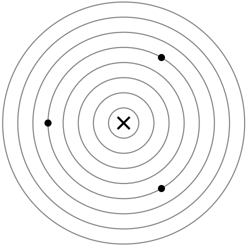
- Of course, can have $\mathcal{B} = \{1, \ldots, N\}$ and sample only one function
- Gives realization of underlying stochastic process
- How about convergence?

SGD – Example

- Let $c_1 + c_2 + c_3 = 0$
- Solve $\text{minimize}_x(\frac{1}{2}(\|x - c_1\|_2^2 + \|x - c_2\|_2^2 + \|x - c_3\|_2^2) = \frac{3}{2}\|x\|_2^2 + c$
- Stochastic gradient method with $\gamma_k = 1/3$



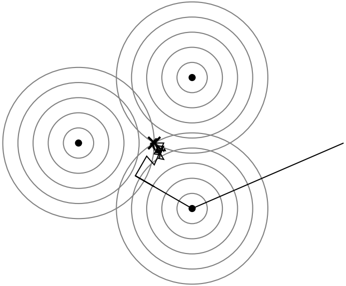
Levelsets of summands



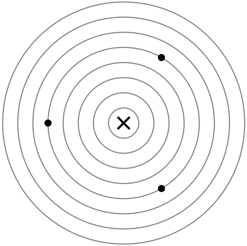
Levelset of sum

SGD – Example

- Let $c_1 + c_2 + c_3 = 0$
- Solve $\operatorname{minimize}_x(\frac{1}{2}(\|x - c_1\|_2^2 + \|x - c_2\|_2^2 + \|x - c_3\|_2^2) = \frac{3}{2}\|x\|_2^2 + c$
- Stochastic gradient method with $\gamma_k = 1/k$



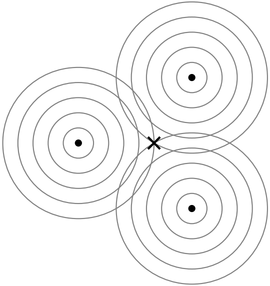
Levelsets of summands



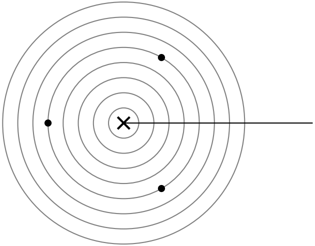
Levelset of sum

SGD – Example

- Let $c_1 + c_2 + c_3 = 0$
- Solve $\text{minimize}_x(\frac{1}{2}(\|x - c_1\|_2^2 + \|x - c_2\|_2^2 + \|x - c_3\|_2^2) = \frac{3}{2}\|x\|_2^2 + c$
- Gradient method with $\gamma_k = 1/3$



Levelsets of summands



Levelset of sum

- SGD will not converge for constant steps (unlike gradient method)

Fixed step-size SGD does not converge to solution

- We can at most hope for finding point $\bar{x}$ such that

$$0 \in \partial g(\bar{x}) + \nabla f(\bar{x})$$

  i.e., the proximal gradient fixed-point characterization

- Consider setting $g = 0$ and assume x_k such that $0 = \nabla f(x_k)$
  - That $0 = \nabla f(x_k)$ does *not* imply $0 = \nabla f_i(x_k)$ for all f_i, hence

$$x_{k+1} = x_k - \gamma_k \nabla f_i(x_k) \neq x_k$$

    i.e., will move away from prox-grad fixed-point for fixed $\gamma_k > 0$
  - Need diminishing step-size rule

Assumptions for convergence

Assumptions:

(i) No nonsmooth term[1], i.e., $g = 0$

(ii) $f : \mathbb{R}^n \to \mathbb{R}$ is β-smooth, for all $x, y \in \mathbb{R}^n$:

$$f(y) \leq f(x) + \nabla f(x)^T(y - x) + \tfrac{\beta}{2}\|y - x\|_2^2$$

(iii) Stochastic gradient of f is unbiased: $\mathbb{E}[\widehat{\nabla} f(x)|x] = \nabla f(x)$

(iv) Variance $\mathbb{E}[\|\widehat{\nabla} f(x) - \nabla f(x)\|_2^2|x] \leq \sigma^2$ is bounded

(v) Step-sizes satisfy $\sum_{k=1}^{\infty} \gamma_k = \infty$ and $\sum_{k=1}^{\infty} \gamma_k^2 < \infty$

[1] Simplifies analysis in nonconvex setting, in convex setting easier to incluce

Convergence proof – Roadmap

- Stochastic gradient descent

$$x_{k+1} = x_k - \gamma_k \widetilde{\nabla} f(x_k) \tag{1}$$

gives realization of stochastic process generated by

$$x_{k+1} = x_k - \gamma_k \widehat{\nabla} f(x_k) \tag{SGD}$$

where

  - (1): $x_k \in \mathbb{R}^n$, can be implemented
  - (SGD): x_k are $\mathbb{R}^n$-valued random variables, not implementable

- Will analyze:
  - stochastic process generated by (SGD) via supermartingale
  - gives results of "all" (except 0-measure) realizations given by (1)

Convergence – SGD martingale inequality

Expectation and variance satisfy (also when conditioned):

- (a) monotonicity: if $X \leq Y$, then $\mathbb{E}[X] \leq \mathbb{E}[Y]$
- (b) linearity: $\mathbb{E}[\alpha X + \beta Y] = \alpha\mathbb{E}[X] + \beta\mathbb{E}[Y]$ for $\alpha, \beta \in \mathbb{R}$
- (c) $\mathbb{E}[\|Z\|_2^2] = \mathbb{E}[\|Z - \mathbb{E}[Z]\|_2^2] + \|\mathbb{E}[Z]\|_2^2$

where X, Y are $\mathbb{R}$-valued, Z is $\mathbb{R}^n$-valued random variables. Therefore,

$$\begin{aligned}
&\mathbb{E}[f(x_{k+1})|x_k] \\
(ii),(a) &\leq \mathbb{E}[f(x_k) + \nabla f(x_k)^T(x_{k+1} - x_k) + \tfrac{\beta}{2}\|x_{k+1} - x_k\|_2^2|x_k] \\
(b),(\textsf{SGD}) &= f(x_k) - \gamma_k \nabla f(x_k)^T \mathbb{E}[\widehat{\nabla} f(x_k)|x_k] + \tfrac{\beta\gamma_k^2}{2}\mathbb{E}[\|\widehat{\nabla} f(x_k)\|_2^2|x_k] \\
(iii),(c) &= f(x_k) - \gamma_k \nabla f(x_k)^T \nabla f(x_k) \\
&\quad + \tfrac{\beta\gamma_k^2}{2}(\mathbb{E}[\|\widehat{\nabla} f(x_k) - \mathbb{E}[\widehat{\nabla} f(x_k)|x]\|_2^2|x_k] + \|\mathbb{E}[\widehat{\nabla} f(x_k)|x_k]\|_2^2) \\
(iii) &= f(x_k) - \gamma_k(1 - \tfrac{\beta\gamma_k}{2})\|\nabla f(x_k)\|_2^2 + \tfrac{\beta\gamma_k^2}{2}(\mathbb{E}[\|\widehat{\nabla} f(x_k) - \nabla f(x_k)\|_2^2|x_k]) \\
(iv) &\leq f(x_k) - \gamma_k(1 - \tfrac{\beta\gamma_k}{2})\|\nabla f(x_k)\|_2^2 + \tfrac{\beta\gamma_k^2}{2}\sigma^2
\end{aligned}$$

SGD – Matching with supermartingale theorem

- SGD satisfies (almost) supermartingale inequality

$$\mathbb{E}[f(x_{k+1})|x_k] \leq f(x_k) - \gamma_k(1 - \tfrac{\beta\gamma_k}{2})\|\nabla f(x_k)\|_2^2 + \tfrac{\beta\gamma_k^2}{2}\sigma^2$$

- After, say m, iterations, $\gamma_k \leq \frac{1}{\beta}$ (diminishing γ_k), hence $\forall k \geq m$:

$$\mathbb{E}[f(x_{k+1})|x_k] \leq f(x_k) - \tfrac{\gamma_k}{2}\|\nabla f(x_k)\|_2^2 + \tfrac{\beta\gamma_k^2}{2}\sigma^2$$

  we consider this sequence and let $m = 1$ (w.l.o.g.)

- Matching sequence with Robbins-Siegmund theorem:

$$V = f, \qquad w_k = \tfrac{\gamma_k}{2}\|\nabla f(x_k)\|_2^2, \qquad v_k = \tfrac{\beta\gamma_k^2}{2}\sigma^2$$

  where $(v_k)_{k\in\mathbb{N}}$ must be summable to apply theorem:

  - γ_k cannot be fixed for all k (“converges to noise ball”)
  - instead $\sum_{k=1}^{\infty}\gamma_k^2 < \infty$ (Assumption (v)) to have v_k summable

SGD – Supermartingale theorem consequence

Since $(\gamma_k^2)_{k\in\mathbb{N}}$ summable, hence $(v_k)_{k\in\mathbb{N}}$ summable:

- $(\mathbb{E}f(x_k))_{k\in\mathbb{N}}$ converges a.s. (not very useful)
- $(w_k)_{k\in\mathbb{N}} = (\frac{\gamma_k}{2}\|\nabla f(x_k)\|_2^2)_{k\in\mathbb{N}}$ is a.s. summable:
  - Even though $\sum_{k=1}^{\infty}\gamma_k = \infty$, cannot conclude $\nabla f(x_k) \to 0$
  - However, $\min\limits_{j=1,\ldots,k}\|\nabla f(x_j)\|_2 \to 0$ as $k \to \infty$ (next slide)

Minimum gradient convergence

- We concluded that $(\frac{\gamma_k}{2}\|\nabla f(x_k)\|_2^2)_{k\in\mathbb{N}}$ is summable a.s.
- Therefore, the following holds at every iteration K a.s.

$$\min_{k=1,\ldots,K}\|\nabla f(x_k)\|_2^2\sum_{k=1}^{K}\tfrac{\gamma_k}{2}\leq\sum_{k=1}^{K}\tfrac{\gamma_k}{2}\|\nabla f(x_k)\|_2^2\leq C$$

  where C is sum of $(\frac{\gamma_k}{2}\|\nabla f(x_k)\|_2^2)_{k\in\mathbb{N}}$, hence finite
- Hence, for "all" realizations (except a 0-measure), i.e., a.s.

$$\min_{k=1,\ldots,K}\|\nabla f(x_k)\|_2^2\leq\tfrac{2C}{\sum_{k=1}^{K}\gamma_k}\to 0$$

  as $K\to\infty$, since $\sum_{k=1}^{\infty}\gamma_k=\infty$ (Assumption (v))

A step-length choice

The requirement to conclude that for “all” realizations $(x_k)_{k\in\mathbb{N}}$

$$\min_{k=1,\dots,K} \|\nabla f(x_k)\|_2^2 \to 0$$

is γ_k not summable but square summable, this is satisfied, e.g., for

$$\gamma_k = \tfrac{M}{k}$$

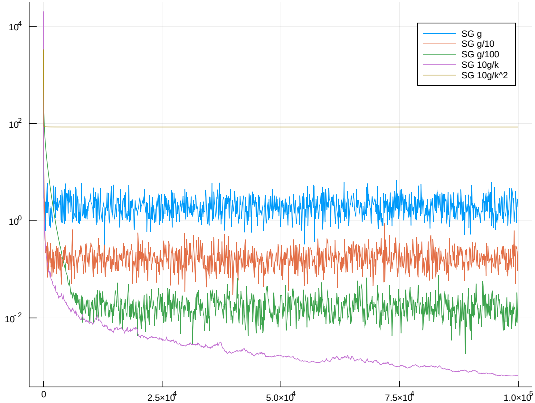
for a positive $M \in \mathbb{R}_{>0}$

Convex setting

- Difficult to prove sequence convergence also in convex setting
- Reason; algorithm moves away from prox-grad fixed-point set

Example – SGD with different step-lengths

- Problem $\text{minimize}_x \|Ax - b\|_2^2$
- with $A \in \mathbb{R}^{40\times20}$ and $b \in \mathbb{R}^{40}$ randomly generated
- x axis: iteration, y axis: function value



Randomized Coordinate Proximal Gradient Descent

Composite problem format

- Consider composite problems of the form

$$\underset{x}{\text{minimize}}\, f(x) + \sum_{i=1}^{n} g_i(x_i)$$

  where

  - $f : \mathbb{R}^n \to \mathbb{R}$ is β-smooth (not necessarily convex)
  - $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is closed convex and separable

- Problem structure includes:
  - Training problems with $\|x\|_1$ or $\|x\|_2^2$ regularization
  - Dual SVM problem formulation

Coordinate proximal gradient descent

- Compute proximal gradient step, update random coordinate j:

$$\begin{aligned} & j \in \{1, \ldots, n\} \text{ is randomly chosen} \\ x_j^{k+1} &= \operatorname{prox}_{\gamma_k g_j}(x_j^k - \gamma_k \nabla f(x^k)_j) \\ x_i^{k+1} &= x_i^k \text{ for all } i \neq j \end{aligned}$$

- Comments:
  - We use super-scripts for iteration and sub-script for coordinate
  - Full gradient computed, inefficient? Sometimes very efficient!
  - Algorithm analysis very similar to proximal gradient descent
  - Can take blocks of coordinates instead
- Recall prox is separable since g is, so algorithm can be seen as

$$\begin{bmatrix} x_1^{k+1} \\ \vdots \\ x_j^{k+1} \\ \vdots \\ x_n^{k+1} \end{bmatrix} = \begin{bmatrix} x_1^k \\ \vdots \\ (\operatorname{prox}_{\gamma g}(x^k - \gamma \nabla f(x^k)))_j \\ \vdots \\ x_n^k \end{bmatrix},$$

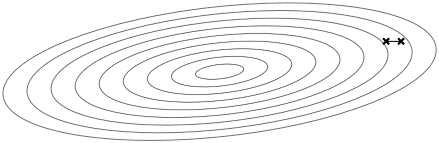
i.e., take full prox-grad step, update only one variable

Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \; \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$

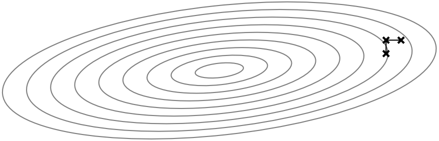


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$

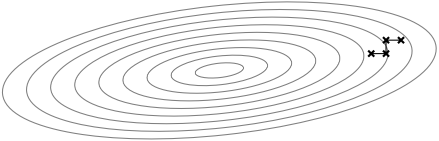


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$

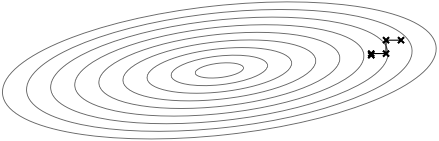


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \; \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T \begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$

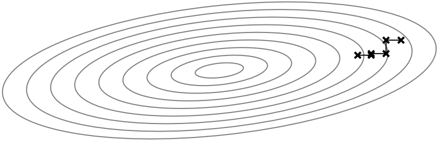


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \;\; \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$

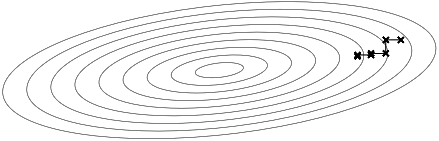


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$

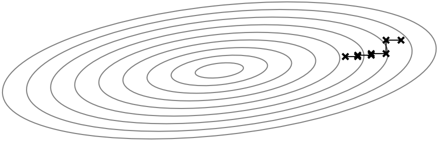


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \;\; \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$

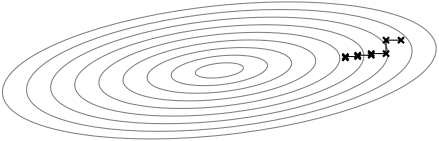


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \;\; \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T \begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$

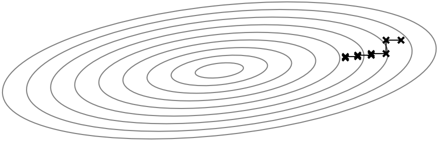


Coordinate descent – Example

- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$

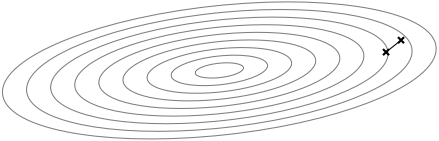


Example – Comparison to gradient descent

- Gradient descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$, similar progress

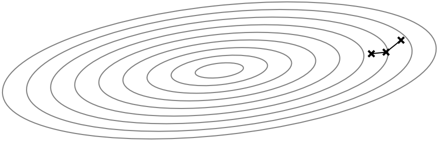


Example – Comparison to gradient descent

- Gradient descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$, similar progress

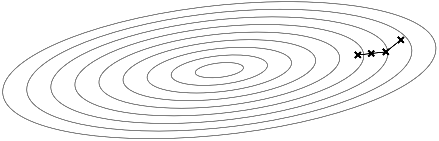


Example – Comparison to gradient descent

- Gradient descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$, similar progress

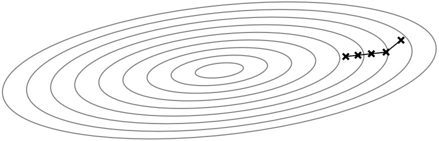


Example – Comparison to gradient descent

- Gradient descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$, similar progress

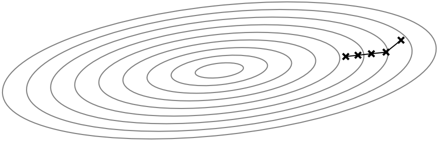


Example – Comparison to gradient descent

- Gradient descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma = \frac{1}{\beta}$, similar progress



Coordinate descent – Another formulation

- We will formulate algorithm differently for analysis
- Introduce coordinate-selection set

$$C_j^x = \{y \in \mathbb{R}^n : y_l = x_l \text{ for all } l \neq j\}$$

  i.e., $y_l = x_l$ for all coordinates $l \neq j$, only y_j is free
- The coordinate descent update is, select j at random and:

$$\begin{aligned}
x^{k+1} &= \underset{y}{\operatorname{argmin}}(f(x^k) + \nabla f(x^k)^T(y - x^k) + \tfrac{1}{2\gamma_k}\|y - x^k\|_2^2 + g(y) + \iota_{C_j^{x^k}}(y)) \\
&= \underset{y}{\operatorname{argmin}}(g(y) + \tfrac{1}{2\gamma_k}\|y - (x^k - \gamma_k \nabla f(x_k))\|_2^2 + \iota_{C_j^{x^k}}(y)) \\
&= \underset{y_j, y_l = x_l^k}{\operatorname{argmin}}(g_j(y_j) + \tfrac{1}{2\gamma_k}\|y_j - (x_j^k - \gamma_k \nabla f(x_k)_j)\|_2^2) \\
&= \begin{cases} \operatorname{prox}_{\gamma_k g_j(y)}(x_j^k - \gamma_k \nabla f(x^k)_j) & \text{for coordinate } j \\ x_l^k & \text{for all coordinates } l \neq j \end{cases}
\end{aligned}$$

Block-coordinate descent

- Let $\mathcal{B}$ be set of block of variables, e.g.,
  - Overlapping blocks with each coordinate in K elements

    $$\mathcal{B} = \{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\}\}$$

    requires single-coordinate separable g
  - Nonoverlapping blocks

    $$\mathcal{B} = \{\{1,2\},\{3,4\},\{5,6\}\}$$

    can have block-separable g
- Draw block j uniformly from $\mathcal{B}$ and define the set C_j^x as before:

  $$C_j^x = \{y \in \mathbb{R}^n : y_l = x_l \text{ for all } l \notin j\}$$

  i.e., $y_l = x_l$ for all coordinates $l \notin j$ only block j is free
- The coordinate descent update is, select block j at random and:

$$x^{k+1} = \underset{y}{\operatorname{argmin}}(f(x^k) + \nabla f(x^k)^T(y - x^k) + \tfrac{1}{2\gamma_k}\|y - x^k\|_2^2 + g(y) + \iota_{C_j^{x^k}}(y))$$

$$= \begin{cases} \operatorname{prox}_{\gamma_k g_j(y)}(x_j^k - \gamma_k \nabla f(x^k)_j) & \text{for block } j \\ x_l^k & \text{for all coordinates } l \notin j \end{cases}$$

where notation x_j is vector of coordinates in j

Expected value of residual

- For convergence, we will need[1] for some $\xi > 0$

$$\mathbb{E}[\|x^{k+1} - x^k\|_2^2 | x^k] = \xi \|\text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) - x_k\|_2^2 \quad (2)$$

- For single-valued coordinate descent, it holds with $\xi = \frac{1}{n}$, proof:

$$\mathbb{E}[\|x^{k+1} - x^k\|_2^2 | x^k] = \sum_{j=1}^{n} \tfrac{1}{n} \left\| \begin{bmatrix} x_1^{k+1} - x_1^k [= 0] \\ \vdots \\ \text{prox}_{\gamma_k g_j(y)}(x_j^k - \gamma_k \nabla f(x^k)_j) - x_j \\ \vdots \\ x_n^{k+1} - x_n^k [= 0] \end{bmatrix} \right\|_2^2$$

$$= \tfrac{1}{n} \sum_{j=1}^{n} \|\text{prox}_{\gamma_k g_j(y)}(x_j^k - \gamma_k \nabla f(x^k)_j) - x_j\|_2^2$$

$$= \tfrac{1}{n} \|\text{prox}_{\gamma_k g(y)}(x^k - \gamma_k \nabla f(x^k)) - x\|_2^2$$

Requires g to be separable!

- Similar thing (different constants) holds for block-coordinate

[1] Actually only need r.h.s. to 0 if l.h.s. to 0 in (2)

We will analyze

We will analyze: for every k, draw j from distribution and update

$$x^{k+1} = \underset{y}{\operatorname{argmin}}(f(x^k) + \nabla f(x^k)^T(y - x^k) + \tfrac{1}{2\gamma_k}\|y - x^k\|_2^2 + g(y) + \iota_{C_j^{x^k}}(y))$$
$$= \operatorname{prox}_{\gamma_{k(g+\iota_{C_j^{x^k}})}}(x_k - \gamma_k \nabla f(x^k))$$

with randomized (block) coordinate descent as special case, where

- coordinate-selector C_j^x depends on random variable j so that:

  $$\mathbb{E}[\|x^{k+1} - x^k\|_2^2 | x^k] = \xi \|\operatorname{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) - x_k\|_2^2$$

  for some $\xi > 0$ (satisfied for what have seen)
- g has separability structure compatible with C_j^x
- Optimality condition (consider $g + \iota_{C_j^{x^k}}$ as g):

  $$0 \in \partial(g + \iota_{C_j^{x^k}})(x^{k+1}) + \gamma_k^{-1}(x^{k+1} - (x^k - \gamma_k \nabla f(x^k)))$$

Assumptions for convergence

Essentially same assumptions and proof as for proximal gradient

(i) $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable (not necessarily convex)

(ii) f is β-smooth, i.e., for all $x, y \in \mathbb{R}^n$:

$$f(y) \leq f(x) + \nabla f(x)^T(y - x) + \tfrac{\beta}{2}\|x - y\|_2^2$$

(iii) $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ closed convex (structure compatible with C_j^x)

(iv) A minimizer exists

(v) Algorithm parameters $\gamma_k \in [\epsilon, \frac{2}{\beta} - \epsilon]$, where $\epsilon > 0$

(vi) C_j^x where j is random variable and g are such that

- $x^k \in C_j^k$ for all k
- $\mathbb{E}[\|x^{k+1} - x^k\|_2^2 | x^k] = \xi \|\mathrm{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) - x_k\|_2^2$, where $\xi > 0$ and expectation is over random variable j

A basic inequality

Using

(a) β-smoothness of f, i.e., Assumption (ii)

(b) Prox optimality condition: There exists $s_{k+1} \in \partial(g + \iota_{C_j^{x^k}})(x^{k+1})$

$$0 = s^{k+1} + \gamma_k^{-1}(x^{k+1} - (x^k - \gamma_k \nabla f(x^k)))$$

(c) Subgradient, $h = g + \iota_{C_j^{x^k}}$: $h(x^k) \geq h(x^{k+1}) + s_{k+1}^T(x^k - x^{k+1})$

it holds for every realization (since $\iota_{C_j^{x^k}}(x^{k+1}) = \iota_{C_j^{x^k}}(x^k) = 0$):

$$\begin{aligned}
f(x^{k+1}) + g(x^{k+1}) &= f(x^{k+1}) + g(x^{k+1}) + \iota_{C_j^{x^k}}(x^{k+1}) \\
{\scriptstyle (a)} &\leq f(x^k) + \nabla f(x^k)^T(x^{k+1} - x^k) + \tfrac{\beta}{2}\|x^{k+1} - x^k\|_2^2 + (g + \iota_{C_j^{x^k}})(x^{k+1}) \\
{\scriptstyle (c)} &\leq f(x^k) + \nabla f(x^k)^T(x^{k+1} - x^k) + \tfrac{\beta}{2}\|x^{k+1} - x^k\|_2^2 \\
&\quad + g(x^k) + \iota_{C_j^{x^k}}(x^k) - s_{k+1}^T(x^k - x^{k+1}) \\
{\scriptstyle (b)} &= f(x^k) + \nabla f(x^k)^T(x^{k+1} - x^k) + \tfrac{\beta}{2}\|x^{k+1} - x^k\|_2^2 \\
&\quad + g(x^k) + \gamma_k^{-1}(x^{k+1} - (x^k - \gamma_k \nabla f(x^k)))^T(x^k - x^{k+1}) \\
&= f(x^k) + g(x^k) + (\gamma_k^{-1} - \tfrac{\beta}{2})\|x^{k+1} - x^k\|_2^2
\end{aligned}$$

Satisfies fixed-point characterization

- Inequality identical to in proximal gradient method
- For proximal gradient method

$$\|x^{k+1} - x^k\|_2 = \|\mathrm{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) - x_k\|_2$$

  hence fixed-point residual converges as consequence of inequality

- Take expectation of inequality conditioned on x^k to get that

$$\begin{aligned} &\mathbb{E}[f(x^{k+1}) + g(x^{k+1})|x^k] \\ &\qquad \leq f(x^k) + g(x^k) + (\gamma_k^{-1} - \tfrac{\beta}{2})\mathbb{E}[\|x^{k+1} - x^k\|_2^2 | x^k] \\ (vi), (v) &\qquad \leq f(x^k) + g(x^k) - \delta\|\mathrm{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) - x_k\|_2^2 \end{aligned}$$

  holds for the underlying stochastic process for some $\delta > 0$

- This is almost supermartingale, Robbins-Siegmund implies

$$\sum_{k=1}^{\infty} \delta \|\mathrm{prox}_{\gamma_k g}(x^k - \gamma_k \nabla f(x^k)) - x^k\|_2^2 < \infty,$$

  i.e., $\|\mathrm{prox}_{\gamma_k g}(x^k - \gamma_k \nabla f(x^k)) - x^k\|_2 \to 0$ as $k \to \infty$ a.s.

Convergence summary

- Analyzed

$$x^{k+1} = \mathrm{prox}_{\gamma_{k(g+\iota_{C_j^{x^k}})}}(x_k - \gamma_k \nabla f(x^k))$$

  and showed $\|\mathrm{prox}_{\gamma_k g}(x^k - \gamma_k \nabla f(x^k)) - x^k\|_2 \to 0$ a.s.

- This implies that a.s. (conclusion in proximal gradient lecture):

$$\partial g(x^{k+1}) + \nabla f(x^{k+1}) \ni \underbrace{\gamma_k^{-1}(x_k - x_{k+1}) + \nabla f(x_{k+1}) - \nabla f(x_k)}_{u_k} \to 0$$

  i.e., fixed-point characterization satisfied in limit

- Question: When is the algorithm efficient to implement?

Efficient implementation

- We consider 1-coordinate case (same applies to block setting)
- g must be separable to be compatible with C_j^x
- The update is

$$x_j^{k+1} = \mathrm{prox}_{\gamma_k g_j}(x_j^k - \gamma_k \nabla f(x^k)_j)$$

- $\mathrm{prox}_{\gamma_k g_j}$ efficient: 1D problem (often closed form solution)
- $\nabla f(x^k)_j$, i.e., element j of gradient:
  - requires in general to compute full gradient, then pick element
  - Will cover two cases when much cheaper
  - Efficient if cost roughly $\frac{1}{n}$ of full gradient cost

Example – Efficient coordinate gradient evaluations

- Let $f(x) = \frac{1}{2}x^T Hx + h^T x$ with $H \in \mathbb{R}^{n\times n}$, then:

  $$\nabla f(x)_j = (Hx)_j + h_j = (h_{j1}, \ldots, h_{jn})^T x + h_j$$

  i.e. updated at cost $\frac{1}{n}$ of full gradient (e.g., for dual SVM)
- Let $\nabla f(x) = L^T(\sigma(Lx) - b)$ with $L \in \mathbb{R}^{m\times n}$ and σ monotone
  - Covers least squares and logistic regression
  - Coordinate gradient

    $$(\nabla f(x))_j = (L^T(\sigma(Lx) - b))_j = (L^T)_j(\sigma(Lx) - b)$$

    where $(L^T)_j \in \mathbb{R}^m$ is j:th row in L^T
  - Assume we know $z = Ly$ at point $y = (x_1, \ldots, x_l, \ldots, x_n)$:

    $$Lx = Ly + L(x - y) = z + L_l(x_l - y_l)$$

    where $L_l \in \mathbb{R}^n$ is l:th row in L (note $x_l - y_l$ scalar) and gradient

    $$(\nabla f(x))_j = (L^T)_j(\sigma(z + L_l(x_l - y_l)) - b)$$

    can be updated at $\frac{1}{m}$ and $\frac{1}{n}$ of cost for the two steps

Convex case

- Assume, in addition to previous assumptions, that f is convex
- The following result can be shown to hold

A sequence $(x_k)_{k\in\mathbb{N}}$ converges a.s. to a fixed-point of

$$T_{\mathrm{PG}}^{\gamma} := \mathrm{prox}_{\gamma_k g}(I - \gamma_k \nabla f)$$

if the following conditions hold almost surely:

(i) $\|\mathrm{prox}_{\gamma g}(x_k - \gamma\nabla f(x_k)) - x_k\| \to 0$ as $k \to \infty$

(ii) $(\|x_k - z\|)_{k\in\mathbb{N}}$ converges for all $z \in \mathrm{fix} T_{\mathrm{PG}}^{\gamma}$

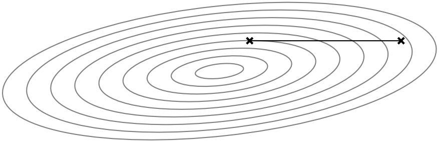
- Condition (i) already shown to hold for coordinate iteration
- Condition (ii) holds for convex problems (but not for nonconvex)
- Proof very similar to for proximal gradient method

Individual step-lengths

- Performance can be greatly improved with individual step-lengths
- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma_1 = \frac{1}{0.1}$, $\gamma_2 = 1$



- Achieved by using tighter upper bound

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \tfrac{1}{2}\|x-y\|_H^2$$

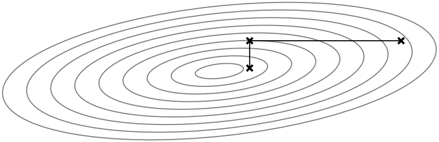
for some matrix H – next lecture

Individual step-lengths

- Performance can be greatly improved with individual step-lengths
- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \;\; \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size parameter $\gamma_1 = \frac{1}{0.1}$, $\gamma_2 = 1$



- Achieved by using tighter upper bound

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \tfrac{1}{2}\|x-y\|_H^2$$

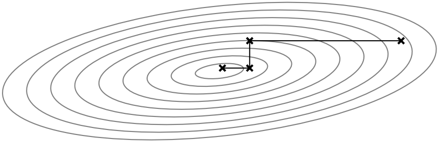
for some matrix H – next lecture

Individual step-lengths

- Performance can be greatly improved with individual step-lengths
- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \;\; \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma_1 = \frac{1}{0.1}$, $\gamma_2 = 1$



- Achieved by using tighter upper bound

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \tfrac{1}{2}\|x-y\|_H^2$$

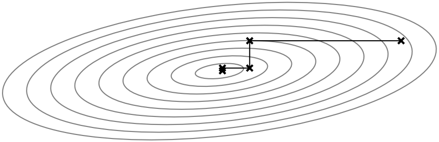
for some matrix H – next lecture

Individual step-lengths

- Performance can be greatly improved with individual step-lengths
- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \;\; \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size parameter $\gamma_1 = \frac{1}{0.1}$, $\gamma_2 = 1$



- Achieved by using tighter upper bound

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \tfrac{1}{2}\|x-y\|_H^2$$

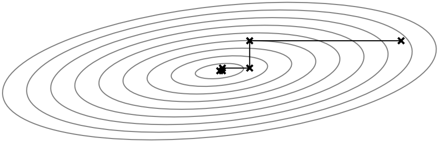
for some matrix H – next lecture

Individual step-lengths

- Performance can be greatly improved with individual step-lengths
- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \;\; \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T \begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma_1 = \frac{1}{0.1}$, $\gamma_2 = 1$



- Achieved by using tighter upper bound

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \tfrac{1}{2}\|x-y\|_H^2$$

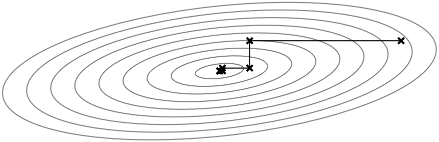
for some matrix H – next lecture

Individual step-lengths

- Performance can be greatly improved with individual step-lengths
- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma_1 = \frac{1}{0.1}$, $\gamma_2 = 1$



- Achieved by using tighter upper bound

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \tfrac{1}{2}\|x-y\|_H^2$$

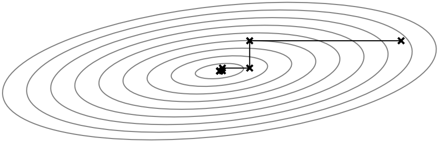
for some matrix H – next lecture

Individual step-lengths

- Performance can be greatly improved with individual step-lengths
- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma_1 = \frac{1}{0.1}$, $\gamma_2 = 1$



- Achieved by using tighter upper bound

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \tfrac{1}{2}\|x-y\|_H^2$$

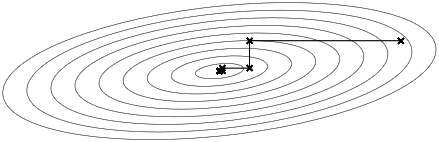
for some matrix H – next lecture

Individual step-lengths

- Performance can be greatly improved with individual step-lengths
- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}}\ \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^T\begin{bmatrix}0.1 & -0.1\\-0.1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

- Step-size parameter $\gamma_1 = \frac{1}{0.1}$, $\gamma_2 = 1$



- Achieved by using tighter upper bound

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \tfrac{1}{2}\|x-y\|_H^2$$

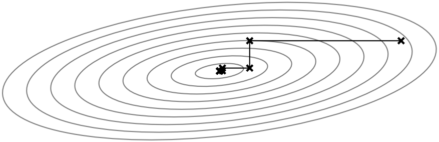
for some matrix H – next lecture

Individual step-lengths

- Performance can be greatly improved with individual step-lengths
- Coordinate descent on β-smooth quadratic problem

$$\underset{x}{\text{minimize}} \;\; \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 0.1 & -0.1 \\ -0.1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Step-size parameter $\gamma_1 = \frac{1}{0.1}$, $\gamma_2 = 1$



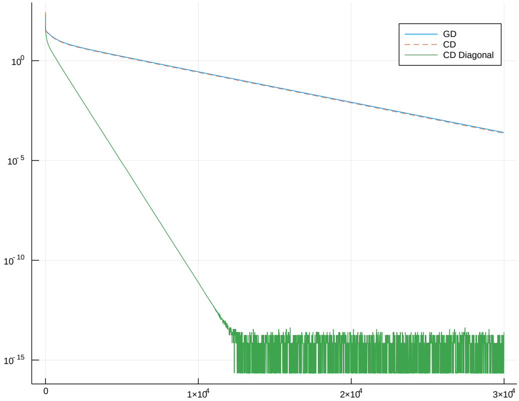
- Achieved by using tighter upper bound

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \tfrac{1}{2}\|x - y\|_H^2$$

for some matrix H – next lecture

Numerical example

- Same least squares construction as in stochastic example
- Compares: gradient, coordinate, and scaled coordinate descent
- x axis normalized for fair comparison, y-axis is function value
- Scaled version much faster


GD
CD
CD Diagonal
10^{0}
10^{-5}
10^{-10}
10^{-15}
1×10^{4}
2×10^{4}
3×10^{4}