Coordinate Descent Methods

Goran Banjac

Large-Scale Convex Optimization
ETH Zurich

April 7, 2020

Coordinate minimization

- we want to minimize a convex function $f\colon \mathbb{R}^n \mapsto \overline{\mathbb{R}}$
- in coordinate descent, we optimize over one variable at a time
- consider
$$f(x) = f(x_1, x_2, x_3, \ldots, x_n)$$
- the *coordinate minimization* (Gauss-Seidel) algorithm is

$$\begin{aligned}
x_1^{k+1} &\in \operatorname*{argmin}_{x_1} f(x_1, x_2^k, x_3^k, \ldots, x_n^k) \\
x_2^{k+1} &\in \operatorname*{argmin}_{x_2} f(x_1^{k+1}, x_2, x_3^k, \ldots, x_n^k) \\
x_3^{k+1} &\in \operatorname*{argmin}_{x_3} f(x_1^{k+1}, x_2^{k+1}, x_3, \ldots, x_n^k) \\
&\vdots \\
x_n^{k+1} &\in \operatorname*{argmin}_{x_n} f(x_1^{k+1}, x_2^{k+1}, x_3^{k+1}, \ldots, x_n)
\end{aligned}$$

Coordinatewise optimality

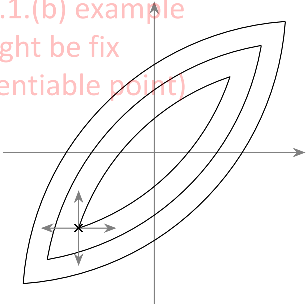
- assume f is differentiable and $\bar{x}$ is a fixed-point of the coordinate minimization algorithm, *i.e.*, $0 = \frac{\partial f}{\partial x_i}(\bar{x})$ for all i
- then $\bar{x}$ is a minimizer of f since

$$\nabla f(\bar{x}) = \left(\frac{\partial f}{\partial x_1}(\bar{x}), \ldots, \frac{\partial f}{\partial x_n}(\bar{x}) \right) = 0$$

Nondifferentiable case

- coordinate optimality does not necessarily imply optimality if f is nondifferentiable
- **example:** $f(x_1, x_2) = |x_1 - x_2| + \frac{1}{2}(x_1^2 + x_2^2)$

see the homework 6.1.(b) example
also, some points might be fix
points(the nondifferentiable point)
but is not optimal



Separable case

- consider the problem

$$\text{minimize} \quad f(x) = g(x) + h(x)$$

- g is convex and differentiable
- h is convex, not necessarily differentiable, and has the form

$$h(x) = \sum_{i=1}^{n} h_i(x_i)$$

- is every fixed-point of the algorithm a minimizer of f?

Separable case – optimality

- let $\mathbf{x}_i^k = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_n^k)$
- we first show that x_i optimizes ith update if

$$\langle \nabla_i g(\mathbf{x}_i^k), y_i - x_i \rangle + h_i(y_i) - h_i(x_i) \geq 0, \quad \forall y_i \in \mathbb{R}$$

- for $\mathbf{y}_i^k = (x_1^{k+1}, \ldots, x_{i-1}^{k+1}, y_i, x_{i+1}^k, \ldots, x_n^k)$ we have

$$g(\mathbf{y}_i^k) - g(\mathbf{x}_i^k) \geq \langle \nabla g(\mathbf{x}_i^k), \mathbf{y}_i^k - \mathbf{x}_i^k \rangle = \langle \nabla_i g(\mathbf{x}_i^k), y_i - x_i \rangle$$

- then, if condition holds, we have for all $\mathbf{y}_i^k$

$$\begin{aligned} f(\mathbf{y}_i^k) - f(\mathbf{x}_i^k) &= g(\mathbf{y}_i^k) - g(\mathbf{x}_i^k) + h_i(y_i) - h_i(x_i) \\ &\geq \langle \nabla_i g(\mathbf{x}_i^k), y_i - x_i \rangle + h_i(y_i) - h_i(x_i) \\ &\geq 0 \end{aligned}$$

- therefore, $f(\mathbf{x}_i^k)$ has the lowest value along ith coordinate

Separable case – optimality

- assume that we have reached a fixed-point of the algorithm, *i.e.*, $\mathbf{x}_i^k = \mathbf{x}_j^k$ for all $i \neq j$
- then, for any y and all $\mathbf{x}_j^k$, we have

$$\begin{aligned} f(y) - f(\mathbf{x}_j^k) &= g(y) - g(\mathbf{x}_j^k) + \sum_{i=1}^n (h_i(y_i) - h_i(x_i)) \\ &\geq \langle \nabla g(\mathbf{x}_j^k), y - \mathbf{x}_j^k \rangle + \sum_{i=1}^n (h_i(y_i) - h_i(x_i)) \\ &= \sum_{i=1}^n \big(\nabla_i g(\mathbf{x}_j^k), y_i - x_i \rangle + h_i(y_i) - h_i(x_i) \big) \\ &\geq 0 \end{aligned}$$

- therefore, $\mathbf{x}_i^k$ minimizes f

Convergence

- strong convergence results require additional assumptions
- we know that the function value is nonincreasing, *i.e.*,

$$f(\mathbf{x}_{i+1}^k) \leq f(\mathbf{x}_i^k)$$

- note that the minimizers in the updates may not be unique
- therefore, arguments for convergence of iterates become tricky
- **variations:**
  - *block coordinate descent*: extension to the case where $x_i \in \mathbb{R}^{n_i}$ are subvectors of x
  - order of updates can be changed (random schemes exist as well)

Parallelization

- the *parallel coordinate minimization* (Jacobi) algorithm is

$$x_i^{k+1} \in \underset{x_i}{\operatorname{argmin}}\, f(x_1^k, \ldots, x_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_n^k)$$

- each component can be updated simultaneously
- unfortunately, the algorithm does not necessarily converge, even when f is differentiable
- *regularized Jacobi algorithm* can be used instead

$$x_i^{k+1} \in \underset{x_i}{\operatorname{argmin}}\, f(x_1^k, \ldots, x_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_n^k) + \tfrac{c}{2}\|x_i - x_i^k\|_2^2$$

- requires Lipschitz smoothness of f and appropriate choice of the regularization parameter $c > 0$ to converge
- there exist *asynchronous* variants

Coordinate gradient descent

- in coordinate gradient descent we solve

$$\text{minimize} \quad f(x)$$

- assume f is block-smooth
  - let

$$\begin{aligned}\mathbf{x}_i &= (x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) \\ \mathbf{y}_i &= (x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)\end{aligned}$$

  - f satisfies

$$f(\mathbf{y}_i) \leq f(\mathbf{x}_i) + \langle \nabla f(\mathbf{x}_i), \mathbf{y}_i - \mathbf{x}_i \rangle + \tfrac{L_i}{2} \|\mathbf{y}_i - \mathbf{x}_i\|_2^2$$

  for some $L_i \geq 0$, all $\mathbf{x}_i, \mathbf{y}_i$ and all $i = \{1, \ldots, n\}$

- equivalent condition:

$$f(\mathbf{y}_i) \leq f(\mathbf{x}_i) + \langle \nabla_i f(\mathbf{x}_i), y_i - x_i \rangle + \tfrac{L_i}{2} \|y_i - x_i\|_2^2$$

- if f is L-smooth, then $L_i \leq L$

Coordinate gradient descent

step size, also notice the minus sign, because this is a minimization problem. The step size is decided by the smoothness of the function, better smooth property, the bigger the smooth is ,the smaller the step size can be chosen. Sounds not intuitive. Can think of quadratic functions, better smooth, smaller the open angle, therefore smaller step size (graphically)

- the algorithm performs the following updates (*e.g.*, in a cyclic fashion):

$$x_i^{k+1} \in \underset{x_i}{\operatorname{argmin}} \left\{ f(\mathbf{x}_i^k) + \langle \nabla_i f(\mathbf{x}_i^k), x_i - x_i^k \rangle + \frac{L_i}{2} \|x_i - x_i^k\|_2^2 \right\}$$
$$= x_i^k - \boxed{\frac{1}{L_i}} \nabla_i f(\mathbf{x}_i^k)$$

- can be extended to the case where $f(x) = g(x) + h(x)$, where g is block-smooth and h is separable
- the updates have the following form:

$$x_i^{k+1} \in \underset{x_i}{\operatorname{argmin}} \left\{ g(\mathbf{x}_i^k) + \langle \nabla_i g(\mathbf{x}_i^k), x_i - x_i^k \rangle + \frac{L_i}{2} \|x_i - x_i^k\|_2^2 + h_i(x_i) \right\}$$
$$= \underset{x_i}{\operatorname{argmin}} \left\{ \frac{L_i}{2} \|x_i - x_i^k + \frac{1}{L_i} \nabla_i g(\mathbf{x}_i^k)\|_2^2 + h_i(x_i) \right\}$$

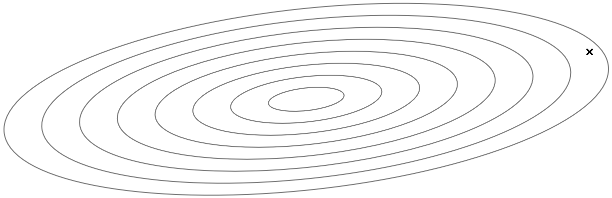
Does coordinate gradient descent use Gaussian-Seidal fashion?
yes

Coordinate gradient descent – example

- consider the following L-smooth problem

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

- choose $L_i = L$ for all i

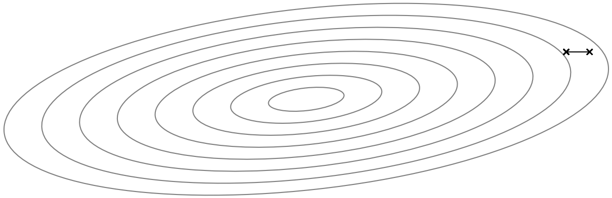


Coordinate gradient descent – example

- consider the following L-smooth problem

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

- choose $L_i = L$ for all i

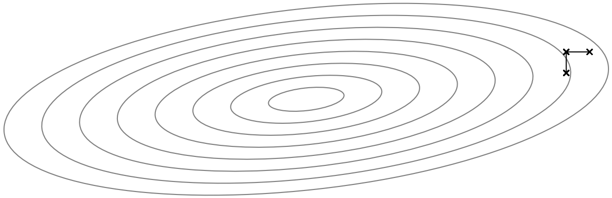


Coordinate gradient descent – example

- consider the following L-smooth problem

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

- choose $L_i = L$ for all i

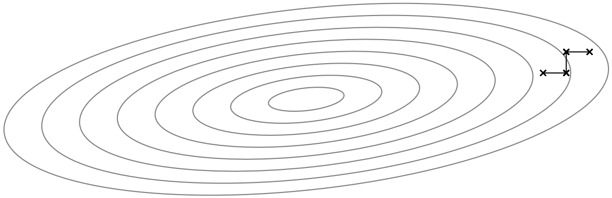


Coordinate gradient descent – example

- consider the following L-smooth problem

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

- choose $L_i = L$ for all i

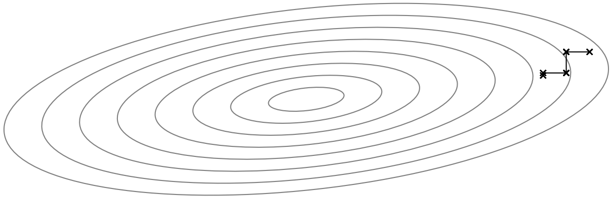


Coordinate gradient descent – example

- consider the following L-smooth problem

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

- choose $L_i = L$ for all i

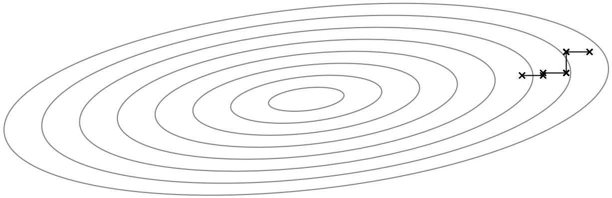


Coordinate gradient descent – example

- consider the following L-smooth problem

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

- choose $L_i = L$ for all i

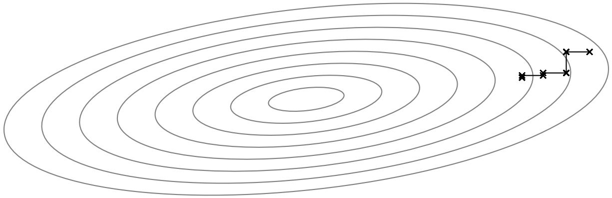


Coordinate gradient descent – example

- consider the following L-smooth problem

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

- choose $L_i = L$ for all i

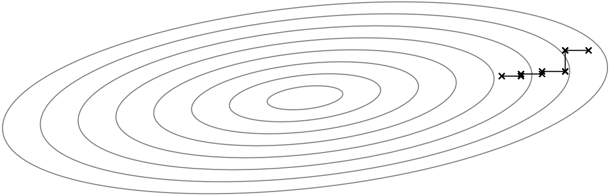


Coordinate gradient descent – example

- consider the following L-smooth problem

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

- choose $L_i = L$ for all i

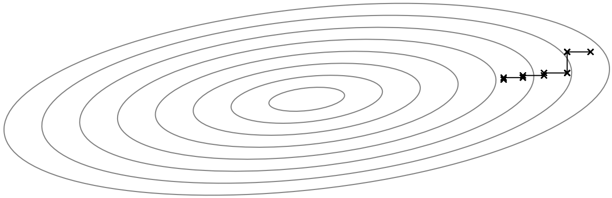


Coordinate gradient descent – example

- consider the following L-smooth problem

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

- choose $L_i = L$ for all i

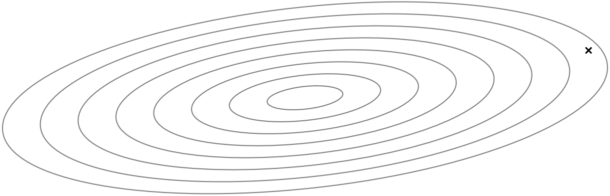


Coordinate gradient descent – example

- consider the following L-smooth problem:

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

- now choose $L_1 = 0.1$ and $L_2 = 1$

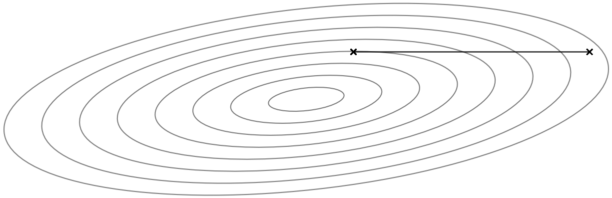


Coordinate gradient descent – example

- consider the following L-smooth problem:

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

- now choose $L_1 = 0.1$ and $L_2 = 1$

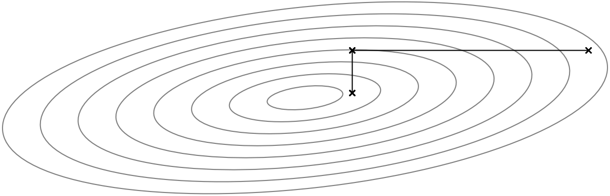


Coordinate gradient descent – example

- consider the following L-smooth problem:

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

- now choose $L_1 = 0.1$ and $L_2 = 1$

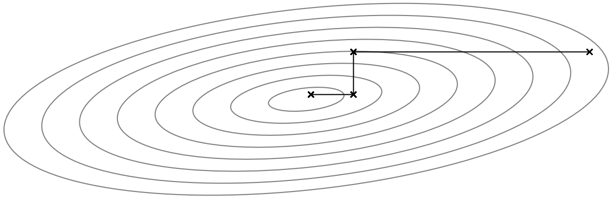


Coordinate gradient descent – example

- consider the following L-smooth problem:

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

- now choose $L_1 = 0.1$ and $L_2 = 1$

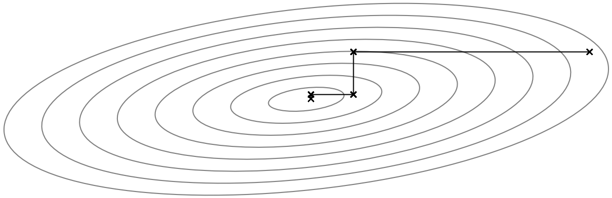


Coordinate gradient descent – example

- consider the following L-smooth problem:

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

- now choose $L_1 = 0.1$ and $L_2 = 1$

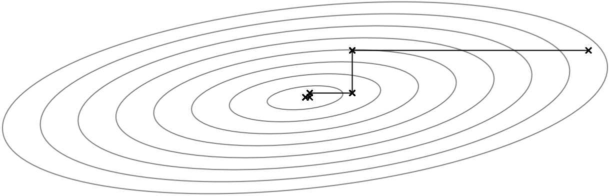


Coordinate gradient descent – example

- consider the following L-smooth problem:

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

- now choose $L_1 = 0.1$ and $L_2 = 1$



Finite sum problems

- consider *finite sum problems* of the form:

$$\text{minimize} \quad f(x) = \tfrac{1}{N} \sum_{i=1}^{N} f_i(x)$$

  where all f_i are differentiable
- for large problems gradient can be expensive to compute
- can be replaced 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 $\widetilde{\nabla} f(x)$ 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, then an unbiased estimator satisfies

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

interesting result from Homework 6. 2(c). The equivalence of stochastic gradient descent in a cyclic fashion with minimum norm problem.

Stochastic gradient descent

$$\boxed{\begin{array}{ll} \min\limits_{x} & \frac{1}{2}\|x\|_2^2, \\ \text{s.t.} & Ax = b, \end{array}} = \boxed{\min_{y} \quad \frac{1}{2}\|Ay-b\|_2^2 = \frac{1}{2}\sum_{j=1}^{m}(a_j^\top y - b_j)^2.}$$

- the following iteration generates a sequence of *random* variables:

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

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

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

- sloppy notation 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 gradient for finite sum problems

$$\text{minimize} \quad f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

- select f_i at random and take gradient step
- realization: let i be drawn from I:

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

hey see here.

  where I is the uniform probability distribution

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

- stochastic gradient is unbiased:

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

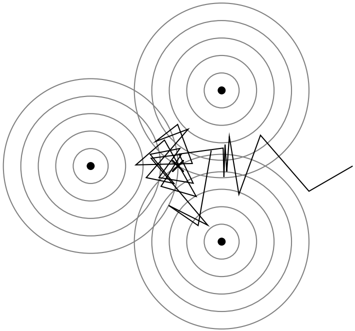
- *mini-batch stochastic gradient*: extension to the case where $\widetilde{\nabla} f(x)$ is obtained from K gradients ∇f_i

Stochastic gradient descent – example

- consider the following finite sum problem:

$$\text{minimize} \quad \tfrac{1}{2}\|x - c_1\|_2^2 + \tfrac{1}{2}\|x - c_2\|_2^2 + \tfrac{1}{2}\|x - c_3\|_2^2$$

- stochastic gradient descent with $\gamma_k = 1/3$

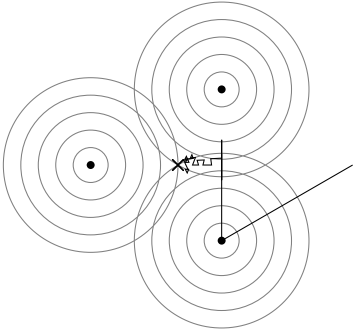


Stochastic gradient descent – example

- consider the following finite sum problem:

$$\text{minimize} \quad \frac{1}{2}\|x - c_1\|_2^2 + \frac{1}{2}\|x - c_2\|_2^2 + \frac{1}{2}\|x - c_3\|_2^2$$

- stochastic gradient descent with $\gamma_k = 1/k$



Assumptions for convergence

- f is L-smooth for all $x, y \in \mathbb{R}^n$
- stochastic gradient of f is unbiased: $\mathbb{E}\big[\widehat{\nabla} f(x) \mid x\big] = \nabla f(x)$
- bounded variance: $\mathbb{E}\big[\|\widehat{\nabla} f(x) - \nabla f(x)\|_2^2 \mid x\big] \leq \sigma^2$
- step sizes satisfy

$$\sum_{k=0}^{\infty} \gamma_k = +\infty, \qquad \sum_{k=0}^{\infty} \gamma_k^2 < +\infty$$

References

- these lecture notes are based to a large extent on the following courses developed by Pontus Giselsson at Lund:
  - Large-Scale Convex Optimization
  - Optimization for Learning

- the original slides can be downloaded from

  `https://archive.control.lth.se/ls-convex-2015/`

  `http://www.control.lth.se/education/engineering-program/frtn50-optimization-for-learning/`