LECTURE 13

LECTURE OUTLINE

- A taxonomy of algorithms for convex optimization
  - Iterative descent
  - Approximation
- A brief overview of approximation algorithms
- Focus on cost function descent
  - Gradient and subgradient methods
  - Gradient projection
  - Newton's method
- Incremental methods

APPROXIMATION

- Problem: Minimize convex $f : \Re^n \mapsto \Re$ over a closed convex set X.
- Approximation approach: Generate $\{x_k\}$ by

$$x_{k+1} \in \arg\min_{x \in X_k} F_k(x),$$

where:

F_k is a function that approximates f

X_k is a set that approximates X

- F_k and X_k may depend on the prior iterates $x_0, \ldots, x_k$, and other parameters.
- Key ideas:
  - Minimization of F_k over X_k should be easier than minimization of f over X
  - x_k should be a good starting point for obtaining x_{k+1}
  - Approximation of f by F_k and/or X by X_k should improve as k increases
- Major types of approximation algorithms:
  - Polyhedral approximation
  - Penalty, proximal, interior point methods
  - Smoothing

ITERATIVE DESCENT

- Generate $\{x_k\}$ such that

$$\phi(x_{k+1}) < \phi(x_k) \qquad \text{iff } x_k \text{ is not optimal}$$

- ϕ is a merit function (also called Lyapounov function)
  - Measures progress towards optimality
  - Is minimized only at optimal points, i.e.,

$$\arg\min_{x \in X} \phi(x) = \arg\min_{x \in X} f(x)$$

- Examples:

$$\phi(x) = f(x), \qquad \phi(x) = \inf_{x^* : \text{optimal}} \|x - x^*\|$$

- In some cases, iterative descent may be the primary idea, but modifications or approximations are introduced:
  - To make the method tolerant of random or nonrandom errors.
  - To make the method suitable for distributed asynchronous computation.

FOCUS ON COST FUNCTION DESCENT

- Consider the unconstrained problem: Minimize $f : \Re^n \mapsto \Re$ over $x \in \Re^n$.
- Generate $\{x_k\}$ by

$$x_{k+1} = x_k + \alpha_k d_k, \qquad k = 0, 1, \ldots$$

where d_k is a descent direction at x_k, i.e.,

$$f(x_k + \alpha d_k) < f(x_k), \qquad \forall \alpha \in (0, \bar{\alpha}]$$

- Many ways to choose the stepsize α_k.
- Sometimes a descent direction is used but the descent condition $f(x_k + \alpha_k d_k) < f(x_k)$ may not be strictly enforced in all iterations.
- Cost function descent is used primarily for differentiable f, with

$$d_k = -S_k \nabla f(x_k)$$

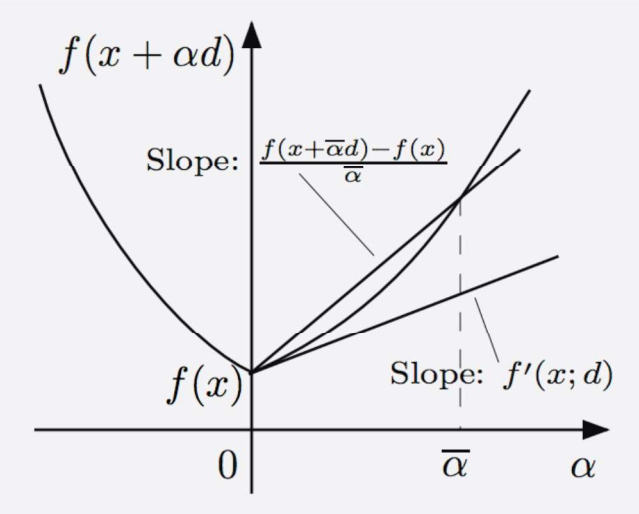
where S_k is positive definite (scaling) matrix.

- Encounters serious theoretical difficulties for nondifferentiable f.

DIRECTIONAL DERIVATIVES

- Directional derivative of a proper convex f:

$$f'(x;d) = \lim_{\alpha \downarrow 0} \frac{f(x+\alpha d) - f(x)}{\alpha}, \quad x \in \text{dom}(f), \; d \in \Re^n$$



- The ratio $\frac{f(x+\alpha d)-f(x)}{\alpha}$ is monotonically nonincreasing as $\alpha \downarrow 0$ and converges to $f'(x;d)$.

- d is a descent direction at x, i.e.,

$f(x+\alpha d) < f(x)$, for all $\alpha > 0$ sufficiently small

iff $f'(x;d) < 0$.

- If f is differentiable, $f'(x;d) = \nabla f(x)'d$, so if S is positive definite, $d = -S\nabla f(x)$ is a descent direction.

MANY ALGORITHMS BASED ON GRADIENT

- Consider unconstrained minimization of differentiable $f : \Re^n \mapsto \Re$ by

$$x_{k+1} = x_k - \alpha_k S_k \nabla f(x_k), \qquad k = 0, 1, \ldots$$

- Gradient or steepest descent method: $S_k = I$.
- Newton's method (fast local convergence):

$$S_k = \big(\nabla^2 f(x_k)\big)^{-1}$$

assuming $\nabla^2 f(x_k)$ is positive definite (otherwise modifications are needed).

- Many algorithms try to emulate Newton's method with less overhead (quasi-Newton, Gauss-Newton method, limited memory, conjugate direction, etc).
- Diagonal scaling: Choose S_k diagonal with inverse 2nd derivatives of f along the diagonal.
- Common stepsize rules:
  - Constant: $\alpha_k \equiv \alpha$
  - Diminishing: $\sum_{k=0}^{\infty} \alpha_k = \infty, \ \alpha_k \downarrow 0$
  - Minimization: $\alpha_k \in \arg\min_{\alpha>0} f(x + \alpha d)$

FAILURE FOR NONDIFFERENTIABLE COST

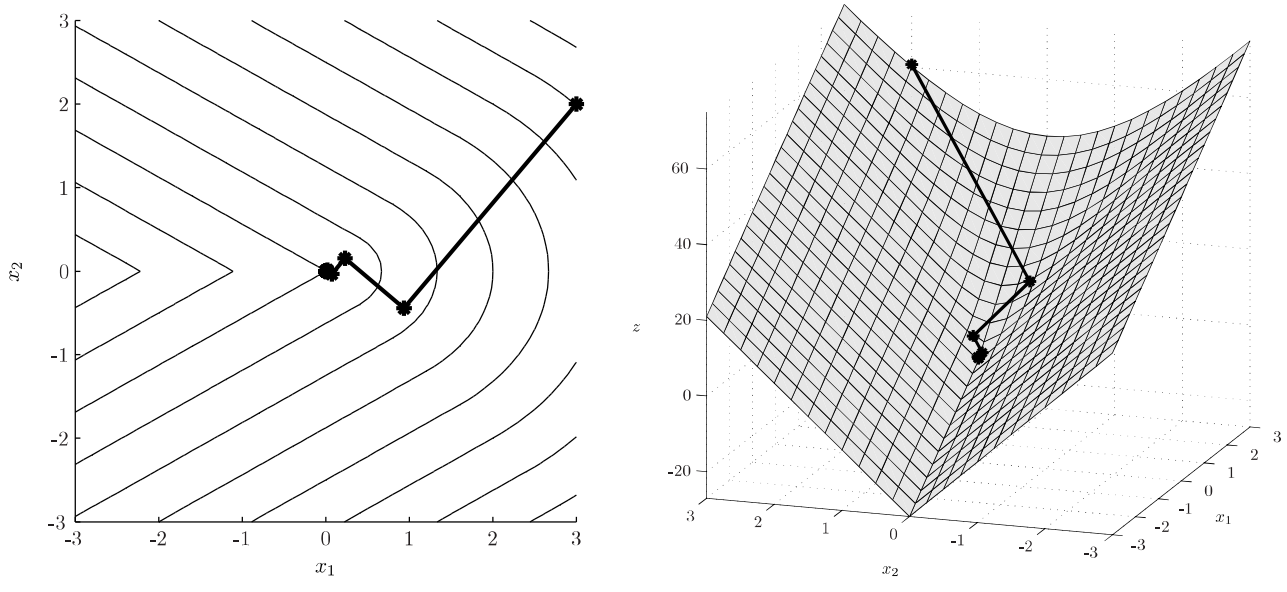
- Start with any $x_0 \in \Re^n$.
- Calculate d_k as the steepest descent direction at x_k

$$d_k = \arg \min_{\|d\|=1} f'(x_k; d)$$

and set

$$x_{k+1} = x_k + \alpha_k d_k$$

- Serious difficulties:
  - Computing d_k is nontrivial at points x_k where f is nondifferentiable.
  - Serious convergence issues due to discontinuity of steepest descent direction.
- Example with α_k determined by minimization along d_k: $\{x_k\}$ converges to nonoptimal point.



CONSTRAINED CASE: GRADIENT PROJECTION

- Problem: Minimization of differentiable $f : \Re^n \mapsto \Re$ over a closed convex set X.
- Cost function descent

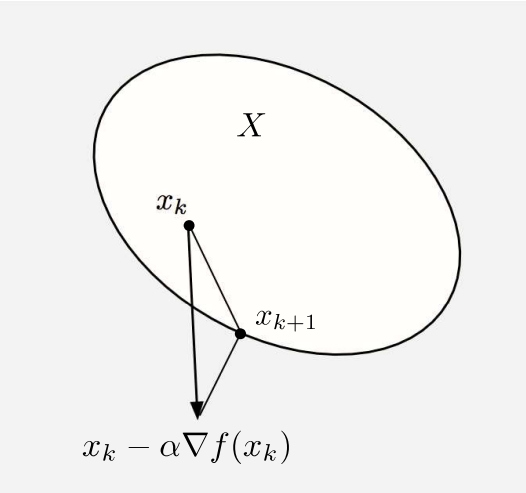
$$x_{k+1} = x_k + \alpha_k d_k$$

where d_k is a feasible descent direction at x_k: $x_k + \alpha d_k$ must belong to X for small enough $a > 0$.

- The gradient projection method:

$$x_{k+1} = P_X\big(x_k - \alpha_k \nabla f(x_k)\big)$$

where $\alpha_k > 0$ is a stepsize and $P_X(\cdot)$ denotes projection on X.



- Projection may be costly. Scaling is tricky.

SUBGRADIENT PROJECTION

- Problem: Minimization of nondifferentiable convex $f : \Re^n \mapsto \Re$ over a closed convex set X.

- Key notion: A subgradient of a convex function $f : \Re^n \mapsto \Re$ at a point x is a vector g such that

$$f(z) \geq f(x) + g'(z - x), \qquad \forall\ z \in \Re^n.$$

At points x where f is differentiable, $\nabla f(x)$ is the unique subgradient.

- Subgradient projection method:

$$x_{k+1} = P_X(x_k - \alpha_k g_k)$$

where g_k is an arbitrary subgradient at x_k.

- Does not attain cost function descent ... but has another descent property: at any nonoptimal point x_k, it satisfies for $a_k > 0$ small enough,

$$\text{dist}(x_{k+1}, X^*) < \text{dist}(x_k, X^*)$$

where X^* is the optimal solution set.

- Typically, a diminishing stepsize α_k is needed.

INCREMENTAL GRADIENT METHOD

- Problem: Minimization of $f(x) = \sum_{i=1}^{m} f_i(x)$ over a closed convex set X (f_i differentiable).

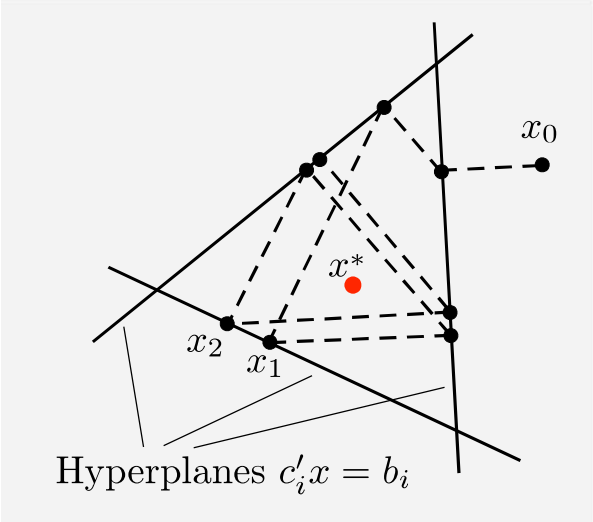
- Operates in cycles: If x_k is the vector obtained after k cycles, the vector x_{k+1} obtained after one more cycle is $x_{k+1} = \psi_{m,k}$, where $\psi_{0,k} = x_k$, and

$$\psi_{i,k} = P_X\big(\psi_{i-1,k} - \alpha_k \nabla f_{i,k}(\psi_{i-1,k})\big), \qquad i = 1, \ldots, m$$

- Example: The Kazcmarz method

$$\psi_{i,k} = \psi_{i-1,k} - \frac{1}{\|c_i\|^2}(c_i'\psi_{i-1,k} - b_i)c_i, \quad i = 1, \ldots, m,$$

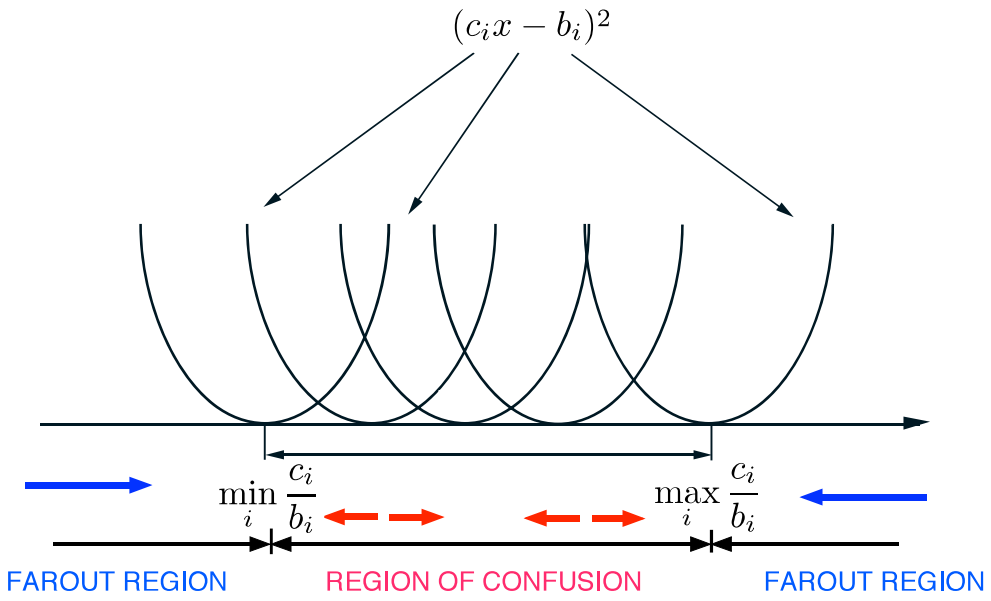
for the case $f_i(x) = \frac{1}{2\|c_i\|^2}(c_i'x - b_i)^2$



COMPARE W/ NONINCREMENTAL GRADIENT

- Two complementary performance issues:
  - Progress when far from convergence. Here the incremental method can be much faster.
  - Progress when close to convergence. Here the incremental method can be inferior.
- Example: Scalar case

$$f_i(x) = \tfrac{1}{2}(c_i x - b_i)^2, \qquad x \in \Re$$



- A diminishing stepsize is necessary for convergence (otherwise the method ends up oscillating within the region of confusion).
- Randomization of selection of component f_i is possible.

OTHER INCREMENTAL METHODS

- Aggregated gradient method:

$$x_{k+1} = P_X \left(x_k - \alpha_k \sum_{\ell=0}^{m-1} \nabla f_{i_{k-\ell}}(x_{k-\ell}) \right)$$

- Gradient method with momentum (heavy ball method):

$$x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k) + \beta_k (x_k - x_{k-1})$$

- Stochastic gradient method for $f(x) = E\{F(x,w)\}$ where w is a random variable, and $F(\cdot, w)$ is a convex function for each value of w:

$$x_{k+1} = P_X \big(x_k - \alpha_k \nabla F(x_k, w_k) \big)$$

where $\nabla F(x_k, w_k)$ is a "sampled" gradient.

- Incremental Newton method.

- Incremental Gauss-Newton method for least squares (extended Kalman filter).