LECTURE 25

LECTURE OUTLINE

- Interior point methods
- Coordinate descent methods
- Distributed asynchronous fixed point computation

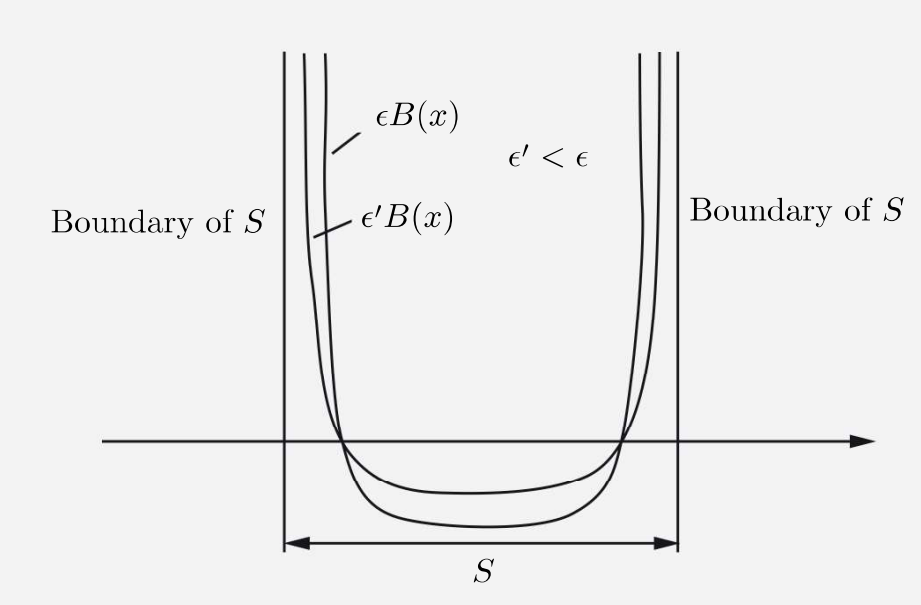
****

References:

- Boyd and Vanderbergue, 2004. Convex Optimization, Cambridge U. Press.
- Bertsekas, D. P., 1999. Nonlinear Programming, Athena Scientific, Belmont, MA.
- Bertsekas, D. P., and Tsitsiklis, J. N., 1989. Parallel and Distributed Algorithms: Numerical Methods, Prentice-Hall.

INTERIOR POINT METHODS

- Problem: $\min_{x \in X,\, g_j(x),\, j=1,\ldots,r} f(x)$
- Let $S = \{x \in X \mid g_j(x) < 0,\, j = 1, \ldots, r\}$ (assumed nonempty). A barrier function, is defined and is continuous on S, and goes to ∞ as any one of the constraints $g_j(x) \uparrow 0$.



- Examples:

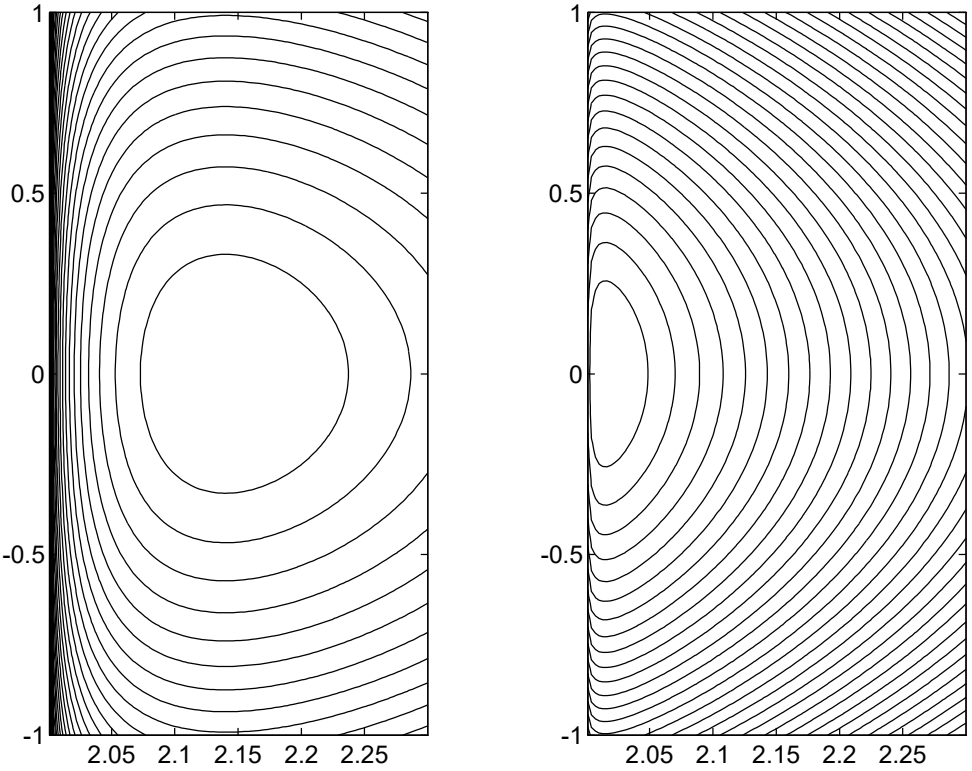
$$B(x) = -\sum_{j=1}^{r} \ln\bigl\{-g_j(x)\bigr\}, \qquad B(x) = -\sum_{j=1}^{r} \frac{1}{g_j(x)}.$$

- Barrier method: Generates

$$x_k = \arg\min_{x \in S} \bigl\{ f(x) + \epsilon_k B(x) \bigr\}, \qquad k = 0, 1, \ldots,$$

where $\epsilon_k \downarrow 0$.

BARRIER METHOD - EXAMPLE



$$\text{minimize} \quad f(x) = \tfrac{1}{2}\left((x^1)^2 + (x^2)^2\right)$$

$$\text{subject to} \quad 2 \leq x^1,$$

with optimal solution $x^* = (2, 0)$.

- Logarithmic barrier: $B(x) = -\ln(x^1 - 2)$
- We have $x_k = \left(1 + \sqrt{1 + \epsilon_k}\,, 0\right)$ from

$$x_k \in \arg\min_{x^1 > 2} \left\{ \tfrac{1}{2}\left((x^1)^2 + (x^2)^2\right) - \epsilon_k \ln(x^1 - 2) \right\}$$

- As ϵ_k is decreased, the unconstrained minimum x_k approaches the constrained minimum $x^* = (2, 0)$.

- As $\epsilon_k \to 0$, computing x_k becomes more difficult because of ill-conditioning (a Newton-like method is essential for solving the approximate problems).

CONVERGENCE

- Assume that X is closed convex, and f, and g_j are convex. Every limit point of a sequence $\{x_k\}$ generated by a barrier method is a minimum of the original constrained problem.

Proof: Let $\{\overline{x}\}$ be the limit of a subsequence $\{x_k\}_{k\in K}$. Since $x_k \in S$ and X is closed, $\overline{x}$ is feasible for the original problem.

If $\overline{x}$ is not a minimum, there exists a feasible x^* such that $f(x^*) < f(\overline{x})$ and therefore (by the Line Segment Principle) also an interior point $\tilde{x} \in S$ such that $f(\tilde{x}) < f(\overline{x})$. By the definition of x_k,

$$f(x_k) + \epsilon_k B(x_k) \leq f(\tilde{x}) + \epsilon_k B(\tilde{x}), \qquad \forall\ k,$$

so by taking limit

$$f(\overline{x}) + \liminf_{k\to\infty,\ k\in K} \epsilon_k B(x_k) \leq f(\tilde{x}) < f(\overline{x})$$

Hence $\liminf_{k\to\infty,\ k\in K} \epsilon_k B(x_k) < 0$.

If $\overline{x} \in S$, we have $\lim_{k\to\infty,\ k\in K} \epsilon_k B(x_k) = 0$, while if $\overline{x}$ lies on the boundary of S, we have by assumption $\lim_{k\to\infty,\ k\in K} B(x_k) = \infty$. Thus

$$\liminf_{k\to\infty} \epsilon_k B(x_k) \geq 0, \qquad \text{a contradiction.}$$

SECOND ORDER CONE PROGRAMMING

- Consider the SOCP

$$\begin{aligned} &\text{minimize} \quad c'x \\ &\text{subject to} \quad A_i x - b_i \in C_i, \ i = 1, \ldots, m, \end{aligned}$$

where $x \in \Re^n$, c is a vector in $\Re^n$, and for $i = 1, \ldots, m$, A_i is an $n_i \times n$ matrix, b_i is a vector in $\Re^{n_i}$, and C_i is the second order cone of $\Re^{n_i}$.

- We approximate this problem with

$$\begin{aligned} &\text{minimize} \quad c'x + \epsilon_k \sum_{i=1}^{m} B_i(A_i x - b_i) \\ &\text{subject to} \quad x \in \Re^n, \quad A_i x - b_i \in \text{int}(C_i), \ i = 1, \ldots, m, \end{aligned}$$

where B_i is the logarithmic barrier function:

$$B_i(y) = -\ln\left(y_{n_i}^2 - (y_1^2 + \cdots + y_{n_i-1}^2)\right), \quad y \in \text{int}(C_i),$$

and $\epsilon_k \downarrow 0$.

- Essential to use Newton's method to solve the approximating problems.

- Interesting complexity analysis

SEMIDEFINITE PROGRAMMING

- Consider the dual SDP

$$\begin{aligned} &\text{maximize} \quad b'\lambda \\ &\text{subject to} \quad D - (\lambda_1 A_1 + \cdots + \lambda_m A_m) \in C, \end{aligned}$$

where $b \in \Re^m$, $D, A_1, \ldots, A_m$ are symmetric matrices, and C is the cone of positive semidefinite matrices.

- The logarithmic barrier method uses approximating problems of the form

$$\text{maximize} \quad b'\lambda + \epsilon_k \ln\big(\det(D - \lambda_1 A_1 - \cdots - \lambda_m A_m)\big)$$

over all $\lambda \in \Re^m$ such that $D - (\lambda_1 A_1 + \cdots + \lambda_m A_m)$ is positive definite.

- Here $\epsilon_k \downarrow 0$.

- Furthermore, we should use a starting point such that $D - \lambda_1 A_1 - \cdots - \lambda_m A_m$ is positive definite, and Newton's method should ensure that the iterates keep $D - \lambda_1 A_1 - \cdots - \lambda_m A_m$ within the positive definite cone.

COORDINATE DESCENT

- Problem

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in X, \end{aligned}$$

where $f : \Re^n \mapsto \Re$ is a differentiable convex function, and

$$X = X_1 \times X_2 \times \cdots \times X_m,$$

where X_i is a closed convex subset $\Re^{n_i}$.

- Partition x into "block" components

$$x = (x^1, x^2, \ldots, x^m),$$

constrained by $x^i \in X_i$.

- (Block) Coordinate descent: At each iteration, minimized the cost w.r.t. each of the block components x_k^i, in cyclic order

$$x_{k+1}^i \in \arg\min_{\xi \in X_i} f(x_{k+1}^1, \ldots, x_{k+1}^{i-1}, \xi, x_k^{i+1}, \ldots, x_k^m)$$

COORDINATE DESCENT CONVERGENCE

- Proposition: Assume that f is convex and continuously differentiable. Assume also that for each $x = (x^1, \ldots, x^m) \in X$ and i,

$$f(x^1, \ldots, x^{i-1}, \xi, x^{i+1}, \ldots, x^m)$$

viewed as a function of ξ, attains a unique minimum over X_i. Let $\{x_k\}$ be the sequence generated by the block coordinate descent method. Then, every limit point of $\{x_k\}$ minimizes f over X.

- Variant to eliminate the uniqueness assumption:

$$x_{k+1}^i = \arg\min_{\xi \in X_i} f(x_{k+1}^1, \ldots, x_{k+1}^{i-1}, \xi, x_k^{i+1}, \ldots, x_k^m) + \frac{1}{2c}\|\xi - x_k^i\|^2,$$

where c is any fixed positive scalar.

- Justification: Apply the original method to minimization over $(x, y) \in X \times X$ of

$$f(x) + \frac{1}{2c}\|x - y\|^2$$

COORDINATE DESCENT EXTENSIONS

- When f is convex but nondifferentiable, the coordinate descent approach may fail in general (there may be nonoptimal points for which descent along all coordinate directions is impossible).
- Favorable special case, when the nondifferentiable portion of f is separable, i.e., f has the form

$$f(x) = F(x) + \sum_{i=1}^{n} G_i(x^i),$$

where F is convex and differentiable, and each $G_i : \Re \mapsto \Re$ is convex.

- A case of special interest is ℓ_1-regularization:

$$\sum_{i=1}^{n} G_i(x^i) = \gamma \|x\|_1$$

- It is possible to iterate the block components in an irregular even randomized order instead of a fixed cyclic order.
- Distributed asynchronous implementation.

ASYNCHRONOUS FIXED POINT ALGORITHMS

- Fixed point problem $x = F(x)$, where $x = (x^1, \ldots, x^m)$, to be solved with m processors.
- Asynchronous fixed point algorithm:

$$x_{t+1}^i = \begin{cases} F_i\left(x_{\tau_{i1}(t)}^1, \ldots, x_{\tau_{im}(t)}^m\right) & \text{if } t \in \mathcal{R}_i, \\ x_t^i & \text{if } t \notin \mathcal{R}_i. \end{cases} \quad (1)$$

$\mathcal{R}_i$ are the computation times of processor i and $t - \tau_{ij}(t)$ are the interprocessor communication delays.

- Some processors may execute more iterations than others, while the communication delays between processors may be unpredictable.
- Continuous Updating and Information Renewal Assumption:
  - The set of times $\mathcal{R}_i$ at which processor i updates x^i is infinite, for each $i = 1, \ldots, m$.
  - $\lim_{t\to\infty} \tau_{ij}(t) = \infty$ for all $i, j = 1, \ldots, m$.
- This is totally asynchronous operation.
- Can show that the algorithm works when F is a contraction with respect to a weighted sup-norm (special case of a more general theorem).

ASYNCHRONOUS CONVERGENCE THEOREM

- Let F have a unique fixed point x^*, and assume that there is a sequence of nonempty subsets $\{S(k)\} \subset \Re^n$ with

$$S(k+1) \subset S(k), \quad k = 0, 1, \ldots,$$

and is such that if $\{y_k\}$ is any sequence with $y_k \in S(k)$, for all $k \geq 0$, then $\{y_k\}$ converges to x^*. Assume further the following:

(1) Synchronous Convergence Condition: We have

$$F(x) \in S(k+1), \qquad \forall\ x \in S(k),\ k = 0, 1, \ldots.$$

(2) Box Condition: For all k, $S(k)$ is a Cartesian product of the form

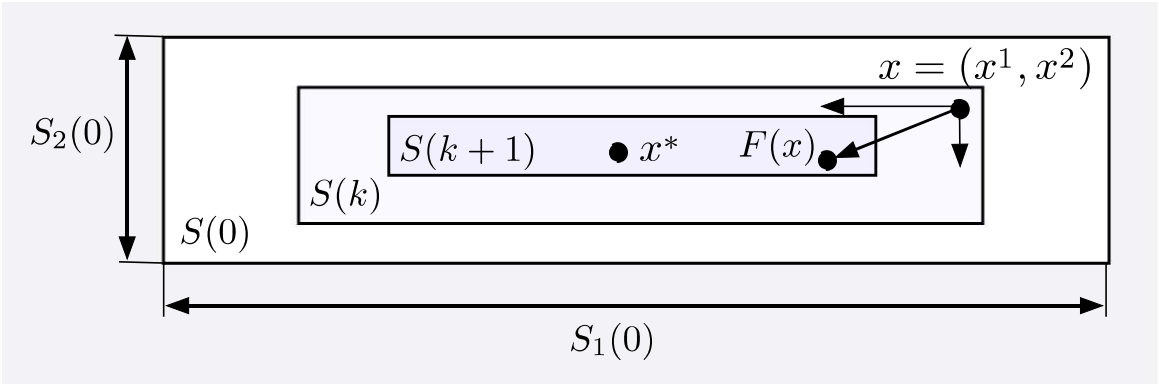
$$S(k) = S_1(k) \times \cdots \times S_m(k),$$

where $S_i(k)$ is a set of real-valued functions on X_i, $i = 1, \ldots, m$.

Then for every $x_0 \in S(0)$, the sequence $\{x_t\}$ generated by the asynchronous algorithm (1) converges to x^*.

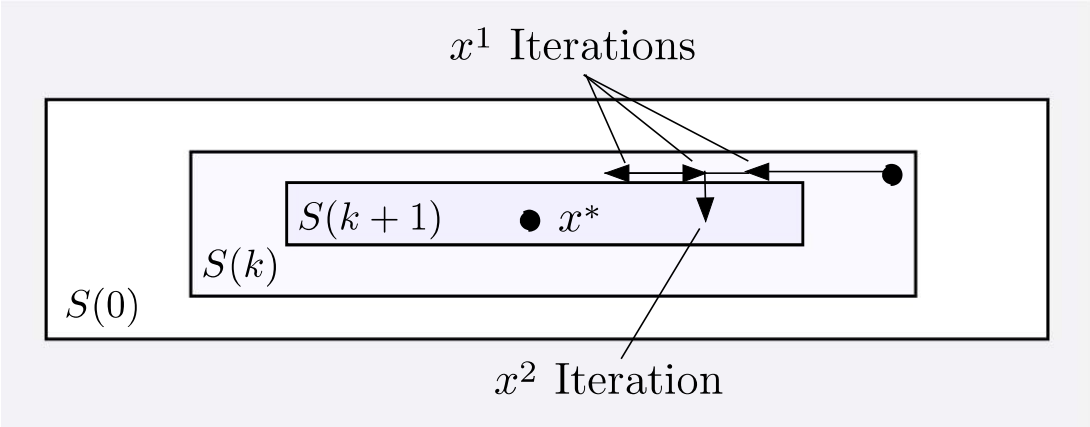
CONVERGENCE ANALYSIS

- Interpretation of assumptions:



A synchronous iteration from any x in $S(k)$ moves into $S(k+1)$ (component-by-component)

- Convergence mechanism:



Key: "Independent" component-wise improvement. An asynchronous component iteration from any x in $S(k)$ moves into the corresponding component portion of $S(k+1)$