Lecture 28

An interior-point method algorithm

Consider the linear program

$$\begin{aligned} \max \quad & c^T x & \\ s.t. \quad & Ax & \le b \\ & x & \ge 0 \end{aligned}$$

An interior-point method algorithm

Example

$$\begin{aligned} \max \quad & 192x_1 + 97x_2 & \\ s.t. \quad & x_1 + x_2 & \le 100 \\ & x_1 & \ge 0 \\ & x_2 & \ge 0 \end{aligned}$$

An interior-point method algorithm

Consider the linear program

$$\begin{aligned} \max \quad & c^T x \\ s.t. \quad & Ax \le b \\ & x \ge 0 \end{aligned}$$

An interior-point method algorithm

Consider the linear program

$$
\begin{aligned}
\max \quad & c^T x & \\
s.t. \quad & b - Ax & \geq 0 \\
& x & \geq 0
\end{aligned}
$$

An interior-point method algorithm

Consider the linear program

$$
\begin{aligned}
\max \quad & c^T x & \\
s.t. \quad & b_1 - a_1^T x & \geq 0 \\
& b_2 - a_2^T x & \geq 0 \\
& \vdots & \vdots \\
& b_m - a_m^T x & \geq 0 \\
& x & \geq 0
\end{aligned}
$$

The barrier function

$$f_{\mu}(x) = c^T x + \mu \left[\sum_{j=1}^{m} \log(b_j - a_j{}^T x) + \sum_{i=1}^{n} \log(x_i) \right]$$

An interior-point method algorithm

Example

$$
\begin{array}{lll}
\max & 192x_1 + 97x_2 & \\
s.t. & x_1 + x_2 & \leq 100 \\
 & x_1 & \geq 0 \\
 & x_2 & \geq 0
\end{array}
$$

An interior-point method algorithm

Example

$$\begin{array}{lll}\max & 192x_1 + 97x_2 & \\ s.t. & 100 - x_1 - x_2 & \geq 0 \\ & x_1 & \geq 0 \\ & x_2 & \geq 0\end{array}$$

An interior-point method algorithm

Example

$$\begin{aligned} \max \quad & 192x_1 + 97x_2 & \\ s.t. \quad & x_1 + x_2 & \leq 100 \\ & x_1 & \geq 0 \\ & x_2 & \geq 0 \end{aligned}$$

$$f_\mu(x) = c^T x + \mu \left[\sum_{j=1}^{m} \log(b_j - a_j{}^T x) + \sum_{i=1}^{n} \log(x_i) \right]$$

$$f_\mu(x) = 192x_1 + 97x_2 + \mu\left[\log(100 - x_1 - x_2) + \log(x_1) + \log(x_2)\right]$$

Q: Write the barrier function for the following linear program

$$
\begin{array}{lcr}
\max & -5x_1 + 9x_2 + 3x_3 & \\
s.t. & 3x_1 - 2x_2 & \leq 20 \\
 & 4x_2 - 5x_3 & \leq 17 \\
 & x_1 & \geq 0 \\
 & x_2 & \geq 0 \\
 & x_3 & \geq 0
\end{array}
$$

i>clicker

Q: Which is the barrier function for the above linear program?

A. $f_\mu(x) = (-5x_1+9x_2+3x_3) + \mu[\log(3x_1 - 2x_2-20) + \log(4x_2-5x_3-17) + \log(x1) + \log(x2) + \log(x3)]$

B. $f_\mu(x) = (-5x_1+9x_2+3x_3) + \mu[\log(20 - 3x_1 + 2x_2) + \log(17-4x_2+5x_3) + \log(x1) + \log(x2) + \log(x3)]$

C. $f_\mu(x) = \log(-5x_1+9x_2+3x_3) + \mu[\log(20 - 3x_1 + 2x_2) + \log(17-4x_2+5x_3) + \log(x1) + \log(x2) + \log(x3)]$

Q: Which is the barrier function for the above linear program?

A. $f_\mu(x) = (-5x_1+9x_2+3x_3) + \mu[\log(3x_1 - 2x_2-20) + \log(4x_2-5x_3-17) + \log(x1) + \log(x2) + \log(x3)]$

B. $f_\mu(x) = (-5x_1+9x_2+3x_3) + \mu[\log(20 - 3x_1 + 2x_2) + \log(17-4x_2+5x_3) + \log(x1) + \log(x2) + \log(x3)]$

C. $f_\mu(x) = \log(-5x_1+9x_2+3x_3) + \mu[\log(20 - 3x_1 + 2x_2) + \log(17-4x_2+5x_3) + \log(x1) + \log(x2) + \log(x3)]$

An interior-point algorithm

Step 1: Choose an initial $\mu^{(0)} > 0$, $\beta > 1$

Step 2: At the i^{th} iteration: $\mu^{(i)}$

Find $x(\mu^{(i)})$, the optimal solution of

$$\text{Max } f_{\mu(i)}(x),$$

Step 3: Let $\mu^{(i+1)} = \mu^{(i)} / \beta$

Repeat steps 2 and 3, for i = 0, 1, … N

An interior-point algorithm

Step 1: Choose an initial $\mu^{(0)} > 0$, $\beta > 1$

Step 2: At the i^{th} iteration: $\mu^{(i)}$

Find $x(\mu^{(i)})$, the optimal solution of

Max $f_{\mu(i)}(x)$,

Step 3: Let $\mu^{(i+1)} = \mu^{(i)} / \beta$

Repeat steps 2 and 3, for i = 0, 1, … N

An interior-point algorithm

Step 1: Choose an initial $\mu^{(0)} > 0$, $x^{(0)}$, $\beta > 1$

Step 2: At the i^{th} iteration: $\mu^{(i)}$, $x^{(i-1)}$

Using $x^{(i-1)}$ as an initial point, carry out k_i iterations of Newton's Method to solve

$$\nabla f_{\mu(i)}(x) = 0$$

Let $x^{(i)}$ be the last Newton's Method iterate.

Step 3: Let $\mu^{(i+1)} = \mu^{(i)}/ \beta$

Repeat steps 2 and 3, for i = 0, 1, … N

Loose ends

- How do we choose $\mu^{(0)} > 0$, $x^{(0)}$, β ?
- How many iterations (N)?
- How many Newton's Method iterations (k_i)?
- We claim that $x(\mu)$ approaches x^* as μ approaches zero.
  Why don't we just compute $x(0)$ directly?

Loose ends

- How do we choose $\mu^{(0)} > 0$, $x^{(0)}$, β ?
- How many iterations (N)?
- How many Newton's Method iterations (k_i)?
- We claim that $x(\mu)$ approaches x^* as μ approaches zero.
  Why don't we just compute $x(0)$ directly?

Loose ends

- How do we choose $\mu^{(0)} > 0$, $x^{(0)}$, β ?
- How many iterations (N)?
- How many Newton's Method iterations (k_i)?
- We claim that $x(\mu)$ approaches x^* as μ approaches zero.
  Why don't we just compute $x(0)$ directly?

Loose ends

- How do we choose $\mu^{(0)} > 0$, $x^{(0)}$, β ?
- How many iterations (N)?
- How many Newton's Method iterations (k_i)?
- We claim that $x(\mu)$ approaches x^* as μ approaches zero.
  Why don't we just compute $x(0)$ directly?

Loose ends

- How do we choose $\mu^{(0)} > 0$, $x^{(0)}$, β ?
- How many iterations (N)?
- How many Newton's Method iterations (k_i)?
- We claim that $x(\mu)$ approaches x^* as μ approaches zero.
  Why don't we just compute $x(0)$ directly?

Loose ends

- How do we choose $\mu^{(0)} > 0$, $x^{(0)}$, β ?
- How many iterations (N)?
- How many Newton's Method iterations (k_i)?
- We claim that $x(\mu)$ approaches x^* as μ approaches zero.
  Why don't we just compute $x(0)$ directly?

Optimization 2: A sightseeing tour through Optimizationland

- Linear programming
  - Formulation
  - Tool for solving network flow problems
  - Tool for solving integer programming problems
    - Branch and bound
    - Gomory cutting planes method

Optimization 2: A sightseeing tour through Optimizationland

- Integer programming
  - Formulation
    - Formulating nonlinear constraints and objective using binary decision variables
  - Methods for solving
    - Branch and bound
    - Gomory cutting planes method

Optimization 2: A sightseeing tour through Optimizationland

- Integer programming
  - Algorithmic ideas:
    - Bounds and approximations
    - Constraint generation
    - Column generation (e.g. the cutting stock problem)

Optimization 2: A sightseeing tour through Optimizationland

- Dynamic programming (deterministic)
  - Example
    - Shortest-path problem
    - Inventory problem
    - Knapsack problems
  - Recurrence relation
  - “Tracing back” to obtain optimal solutions

Optimization 2: A sightseeing tour through Optimizationland

- Dynamic programming (stochastic)
  - Example
    - “Betting” games; games involving uncertainties
    - Inventory problem with stochastic demand
  - Maximizing/Minimizing expected values
  - Recurrence relation
  - “Tracing back” to obtain optimal solutions

Optimization 2: A sightseeing tour through Optimizationland

- Nonlinear optimization
  - Necessary conditions for optimality
    - Unconstrained
    - Constrained
  - Convex functions (are nice!)
  - Newton’s Method
    - As a tool for finding minimizers of convex functions
  - Interior-point Method (for linear program)
    - Barrier function

Optimization 2: A sightseeing tour through Optimizationland

- Network optimization
  - Building block problems:
    - Mincost flow
    - Shortest-path
    - The Assignment Problem
    - Maxflow/Mincut
  - Algorithms:
    - The Hungarian Algorithm
    - Ford-Fulkerson

Optimization 2: A sightseeing tour through Optimizationland

- Network optimization
  - Formulating problems as a network flow problem
    - Project assignment (as mincut)
    - Baseball elimination (as maxflow)
  - Important modeling ideas:
    - “Special case” of a problem
    - How to model a “real-world problem” to fit a specific model

Thank you for a great semester!