Lecture 20: Linear Programming: A really very extremely big hammer

Note: Wed Nov 22 will introduce a new topic.
Midterm grades will be released at the end of this class.

- Breathe!
- These grades don’t count for that much.
- In past 421 I’ve had a student with a midterm grade in the mid-60’s end with a 4.0 in the course

<table>
<thead>
<tr>
<th>Histogram</th>
<th>Median</th>
<th>Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>90s</td>
<td>10</td>
<td>Median</td>
<td>74</td>
</tr>
<tr>
<td>80s</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70s</td>
<td>23</td>
<td>Average</td>
<td>67.71</td>
</tr>
<tr>
<td>60s</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50s</td>
<td>10</td>
<td></td>
<td>20.32</td>
</tr>
<tr>
<td>40s</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;40</td>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Given:** a polytope

**Find:** the *lowest* point in the polytope
**Given:** a polytope

**Find:** the *lowest* point in the polytope

We have fast algorithms for this!

Maximize $z_1 + 2z_3$

subject to:

$2z_1 - z_2 + 3z_3 \leq 1$

$-z_1 + z_2 - z_3 \leq 5$
Linear Algebra primer

For $\mathbf{a}, \mathbf{x} \in \mathbb{R}^n$ we think of $\mathbf{a}$ and $\mathbf{x}$ as column vectors

$$\mathbf{a}^\top \mathbf{x} = a_1 x_1 + \cdots + a_n x_n$$

The set of $\mathbf{x}$ satisfying $\mathbf{a}^\top \mathbf{x} = \mathbf{0}$ is hyperplane
\[ a^T x = 0 \]
$a^T x \leq 0$
**Given:** a polytope

**Find:** the *lowest* point in the polytope
Given: a polytope

Find: the \textit{lowest} point in the polytope
For $a, x \in \mathbb{R}^n$ we think of $a$ and $x$ as column vectors

$$a^\top x = a_1 x_1 + \cdots + a_n x_n$$

Write $m \times n$ matrix $A$, for $Ax = \begin{bmatrix} A_1 x \\ A_2 x \\ A_3 x \\ \vdots \\ A_m x \end{bmatrix}$ where $A_1, \ldots, A_n$ are rows of $A$. 
Given: a polytope

Find: the *lowest* point in the polytope

Maximize $c^T x$
subject to  
$Ax \leq b$.

$Ax \leq b$ means 
$(Ax)_i \leq b_i$
for all $i$
Given: a polytope

Find: the lowest point in the polytope

Maximize $c^T x$
subject to $Ax \leq b$.

Typically # constraints $m \geq n$.

Lowest point is a vertex defined by some $n$ rows, $A'x = b'$.
Maximize $c^T x$
subject to

$Ax \leq b$

$x \geq 0$

Maximize $z_1 + 2z_3$
subject to

$2z_1 - z_2 + 3z_3 \leq 1$

$-z_1 + z_2 - z_3 \leq 5$

replace each $z_i$ by $x_{i,a} - x_{i,b}$ for $x_{i,a}, x_{i,b} \geq 0$

Maximize $(x_{1,a} - x_{1,b}) + 2(x_{3,a} - x_{3,b})$
subject to

$2(x_{1,a} - x_{1,b}) - (x_{2,a} - x_{2,b}) + 3(x_{3,a} - x_{3,b}) \leq 1$

$-(x_{1,a} - x_{1,b}) + (x_{2,a} - x_{2,b}) - (x_{3,a} - x_{3,b}) \leq 5$

$x \geq 0$
Max Flow

**Given:** A Flow Network $G = (V, E)$ with source $s$, sink $t$, and $c: E \rightarrow \mathbb{R}^\geq 0$

**Maximize** flow out of $s$

subject to

- respecting capacities
- flow conservation at internal nodes

**LP Variables:**

$x_e$ for each $e \in E$ representing flow on edge $e$

Maximize

$$\sum_{e \text{ out of } s} x_e$$

subject to

$$0 \leq x_e \leq c(e) \text{ for every } e \in E$$

$$\sum_{e \text{ out of } v} x_e = \sum_{e \text{ into } v} x_e$$

for every node $v \in V - \{s, t\}$
Max Flow

Maximize \( \sum_{e \text{ out of } s} x_e \)
subject to

\( 0 \leq x_e \leq c(e) \) for every \( e \in E \)

\( \sum_{e \text{ out of } v} x_e = \sum_{e \text{ into } v} x_e \)
for every node \( v \in V - \{s, t\} \)

Replace equality constraints by a pair of inequalities

Maximize \( c^T x \)
subject to

\( Ax \leq b \)
\( x \geq 0 \)

1. \( c_e = \begin{cases} 1 & \text{if } e \text{ out of } s \\ 0 & \text{otherwise} \end{cases} \)
2. \( x_e \leq c(e) \)
3. \( \sum_{e \text{ out of } v} x_e - \sum_{e \text{ into } v} x_e \leq 0 \)
4. \( \sum_{e \text{ into } v} x_e - \sum_{e \text{ out of } v} x_e \leq 0 \)
5. \( x \geq 0 \)
Minimization or Maximization

Minimize $c^T x$
subject to
$Ax \geq b$
$x \geq 0$

Maximize $(-c)^T x$
subject to
$(-A)x \leq (-b)$
$x \geq 0$

Different objective function value (Sign flip)
Shortest Paths

**Given:** Directed graph $G = (V, E)$ vertices $s, t$ in $V$

**Find:** (length of) shortest path from $s$ to $t$

**Claim:** Length $\ell$ of the shortest path is the solution to this program.

**Proof sketch:** A shortest path yields a solution of cost $\ell$. Optimal solution must be a combination of flows on shortest paths also cost $\ell$; otherwise there is a part of the 1 unit of flow that gets counted on more than $\ell$ edges.

Minimize

$$\sum_{e} x_e$$

subject to

$$x \geq 0$$

$$\sum_{e \text{ out of } s} x_e = 1 \quad \text{Flow out of } s \text{ is 1}$$

$$\sum_{e \text{ into } t} x_e = 1 \quad \text{Flow into } t \text{ is 1}$$

$$\sum_{e \text{ out of } v} x_e = \sum_{e \text{ into } v} x_e$$

for every node $v \in V - \{s, t\}$

Flow conservation
Shortest Paths

**Given:** Directed graph $G = (V, E)$
vertices $s, t$ in $V$

**Find:** shortest path from $s$ to $t$

**Claim:** Length $\ell$ of the shortest path is the solution to this program.

**Proof sketch:** A shortest path yields a solution of cost $\ell$. Optimal solution must be a combination of flows on shortest paths, also cost $\ell$; otherwise parts of the 1 unit of flow that gets counted on more than $\ell$ edges.
**Vertex Cover**

**Given:** Undirected graph $G = (V, E)$

**Find:** smallest set of vertices touching all edges of $G$.

**Doesn’t work:** To define a set we need $x_v = 0$ or $x_v = 1$

Natural Variables for LP:

- $x_v$ for each $v \in V$

Minimize $\sum_{v} x_v$

subject to

- $0 \leq x_v \leq 1$ for each node $v \in V$
- $x_u + x_v \geq 1$ for each edge $\{u, v\} \in E$

This LP optimizes for a different problem: **“fractional vertex cover”**.

$x_v$ indicates the fraction of vertex $v$ that is chosen in the cover.

LP minimum = $3/2$

Vertex Cover minimum = $2$
What makes Max Flow different?

For Vertex Cover we only got a fractional optimum but for Max Flow can get integers.

- Why?
  - Ford-Fulkerson analysis tells us this for Max Flow.
  - Is there a reason we can tell just from the LP view?

Recall: Optimum is at some vertex \( x \) satisfying \( A' x = b' \) for some subset of exactly \( n \) constraints.

This means that \( x = (A')^{-1} b' \).

Entries of the matrix inverse are quotients of determinants of sub-matrices of \( A' \) so, for integer inputs, optimum is always rational.

Fact: Every full rank submatrix of MaxFlow matrix \( A \) has determinant \( \pm 1 \)

\[ \Rightarrow \] all denominators are \( \pm 1 \) \( \Rightarrow \) integers.

\( A \) is “totally unimodular”

Next: How MaxFlow=MinCut is an example of a general “duality” property of LPs