Lecture 19: Linear Programming #2

Last week: basic definitions, examples

Today: some theory

Simplex Algorithm

Dantzig

Theorem: Suppose LP has at least one feasible point
and suppose the feasible region is bounded
then any optimal point is a corner of the feasible region.

Objective function:

\[ \text{max } 5x_1 + x_2 \]

Subject to:

1. \[ x_1 + x_2 \leq 7 \]
2. \[ -2x_1 + 3x_2 \leq 4 \]
3. \[ x_1 \geq 0 \]
4. \[ x_2 \geq 0 \]

Feasible region:
Finite algo — enumerate over all corners.
   (I finitely many corners)

Simplices explore corners "efficiently"
   — practically pretty good algorithm

Polyhedron = Intersection of finite # of half spaces
Polytope = Polyhedron that is bounded
\[ \text{max} \quad 5x \]
\[ \text{st} \quad x \geq 3 \]
\[ x \leq 22 \]

\[ \text{max} = 5x \]
\[ \text{st} \quad x \geq 3 \]
\[ x \leq 22 \]

Intuition for Theorem:
\[ H = \frac{2}{3} y_1 + y_2 = 0 \]
\[ \exists y : \langle (5,1), y \rangle = 0 \]

\( \text{Max value} \)
Example 1

\[ \text{max } \mathbf{x} = -x_1 \]
\[ \text{st } x_1 \geq 2 \]

Value = 7
but no corner.

Example 2

\[ \text{max } x_1 + x_2 \]
\[ x_1 + x_2 \leq 7 \]

Assume:

region is bounded.
Theorem: Suppose feasible region is non-empty and it is bounded \( \Rightarrow \) there is optimum

Corner (vertex, extreme point)

\( x \) is not a corner of polytope \( X \) if

\( \exists \) two other points \( x_1, x_2 \neq x \) such that

\[
 x = \frac{x_1 + x_2}{2}
\]

\( x \) is a corner otherwise
Idea: if \( \exists \theta > 0 \) s.t. \( x + \theta c \) is feasible, so for it
\[
\langle c, x \rangle < \langle c, y \rangle
\]

\[
\langle c, y \rangle = \langle c, x + \theta c \rangle
= \langle c, x \rangle + \theta \langle c, c \rangle
\]

\[\|c\|^2 \geq 20\]

Idea #2: if \( \exists \) any vector \( d \)
\[
\text{st.} \quad \langle c, d \rangle \geq 20
\]
then \( x + \theta d \) is in no worse

Case 1: get stuck at a corner

Case 2: get stuck at a face
Two things:

1. $x$ is optimal
2. If not at a corner $\Rightarrow$ can move to a corner without reducing value

Proof: Suppose $x$ is not optimal, i.e., $y \in X$ such that $\langle c, y \rangle > \langle c, x \rangle$

Then $\langle c, y-x \rangle > 0$ which is a contradiction.
\[
\langle c, z \rangle = \langle c, \frac{1}{2} x + \frac{1}{2} y \rangle = \frac{1}{2} \langle c, x \rangle + \frac{1}{2} \langle c, y \rangle
\]

\[
\max_x z^T x \quad \text{subject to} \quad z \in \{0, 1\}
\]
Simplicex Algorithm

1. Start at any corner of feasible region $x$ is corner
2. Look at the neighbors of $x$
   
   if $\exists$ a neighbor $x_i$ of $x$ that is better
   
   $\langle c, x_i \rangle > \langle c, x \rangle$
   
   then $x \leftarrow x_i$
   repeat step 2

else
   // all neighbors of $x$ are no better
   declare "$x$ is optimum"
   stop
Fact: if all neighbors of $x$ are no better than $x$, then $x$ is optimum.

1. **Finding the optimum?**
   - Where to start?
   - Which neighbor to pick?
   - If $\exists$ a choice

$$
\langle c, x_1 \rangle \leq \langle c, x \rangle \\
\langle c, x_2 \rangle \leq \langle c, x \rangle \\
\langle c, z \rangle \leq \langle c, x \rangle \quad \text{by linearly}
$$

So $y$ was better.

$\Rightarrow z_1$ is better than $x$. 
0. If $m$ constraints in the LP $\Rightarrow \leq m$ corners to 2-D polygon

0. Each time improve $\Rightarrow$ new corner

$\Rightarrow \leq m$ steps to finish

Thm: Simplex alg in 2D takes at most $m$ steps (at most $m$ constraints)

Thm: Simplex in high dimensions can take exponential time.

(However, runs very well in practice often)

Perhaps next lecture: Interior-point algos are fast (in theory and in practice).