Lecture #34: Zero Sum Games

Last lecture with new material, rest will be review

Today we will talk about strategic games and how to model them as LPs

Example: Matching Pennies

- Alice and Bob (simultaneously) choose either H or T
- If they match
  Alice gets \$1 from Bob

  Else
  Bob gets \$1 from Alice

def: The payoff matrix for a zero sum game with m strategies for Alice and n strategies for Bob is an $m \times n$ matrix A with

$A_{ij} \triangleq$ Alice's payoff (Bob's loss) when she plays strategy i against strategy j

e.g. for matching pennies:

$$p = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \qquad A = \begin{array}{c c} & \begin{array}{cc} H & T \end{array} \\ \begin{array}{c} H \\ T \end{array} & \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \end{array} \qquad p^T A = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

<u>Q</u>: what strategies should Alice and Bob play?

<u>def</u>: A <u>Nash equilibrium</u> is a pair of distributions p and q that satisfy

(1) If Alice chooses her strategy according to p, Bob cannot do better than choosing according to q

(2) and vice versa

<u>Key Point</u>: We'll be able to encode these conditions using an LP

Step #1: Write the constraint that p is a distribution:

$$\vec{0} \leq p \leq \vec{1} \quad \text{and} \quad \vec{1}^T p = 1$$

Similarly for q: $\vec{0} \leq q \leq \vec{1}$ and $\vec{1}^T q = 1$, but note the dimensions are different

Step #2: Enforce the constraint that there is no better response

$$\forall_i \qquad p^T A e_i \geq p^T A q$$

This says: For any alternative strategy for Bob, he loses at least as much

Problem: This is quadratic in the variables p and q, so it doesn't lead to an LP

Solution: We'll change the game so Alice goes first

$$(P) = \max \gamma$$
$$\text{s.t. } p^T A \geq \gamma \vec{1}^T$$
$$\vec{0} \leq p \leq \vec{1}$$
$$\vec{1}^T p = 1$$

Q2: What is this LP saying in words?

Maximize γ so that Alice can find some strategy p where no matter Bob does, Alice wins at least γ

Q3: This doesn't seem fair! Why doesn't Bob go first?

$$(D) = \min \lambda$$
$$\text{s.t. } \lambda \vec{1} \geq Aq$$
$$\vec{0} \leq q \leq \vec{1}$$
$$\vec{1}^T q = 1$$

Theorem [von Neumann] The optimal values of these two linear programs are equal

The optimal value is called the game value

Let's do another example

$$A = \begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix}$$

Q4: How do you expect the game value to change compared to matching pennies?

Let's guess the answer (can find with LP)

$$p = \begin{bmatrix} \frac{2}{5} \\ \frac{3}{5} \end{bmatrix} \qquad q = \begin{bmatrix} \frac{3}{5} \\ \frac{2}{5} \end{bmatrix}$$

$$p^T A = [-\tfrac{1}{5}, -\tfrac{1}{5}]$$

$$Aq = \begin{bmatrix} -\frac{1}{5} \\ -\frac{1}{5} \end{bmatrix}$$

<u>Q5</u>: So what is the game value?

$$-\frac{1}{5}$$

In particular, no matter who goes first they can guarantee $-\frac{1}{5}$ payoff

Let's do a more sophisticated example

<u>Colonel Blotto Games</u>:

- <u>Defender</u>: allocates m armies between two battlefields
- <u>Attacker</u>: same, but n armies

Defender loses if he has fewer on either

e.g. $m=3$, $n=2$

$$\begin{array}{c c} & \begin{array}{ccc} (2,0) & (1,1) & (0,2) \end{array} \\ \begin{array}{c} (3,0) \\ (2,1) \\ (1,2) \\ (0,3) \end{array} & \begin{bmatrix} 1 & -1 & -1 \\ 1 & 1 & -1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 \end{bmatrix} \end{array}$$

Optimal strategy for the attacker

$$q = \begin{bmatrix} \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix}$$

And hence game value $= 0$

Q6: Who has the advantage for $m=4, n=3$?

	(3,0)	(2,1)	(1,2)	(0,3)
(4,0)	1	−1	−1	−1
(3,1)	1	1	−1	−1
(2,2)	−1	1	1	−1
(1,3)	−1	−1	1	1
(0,4)	−1	−1	−1	1

Vote: Do you think the game value is

(a) > 0 defender wins (b) $= 0$ (c) < 0 attacker wins ?

These games are hard to predict exactly b/c they are complex (but tractable) optimization problems!

Last Application: How should you allocate advertising money to win an election?

We can model it as a giant zero sum game

battlefields: states

armies: campaign funds

outcome: first to 270

Can also give each state its own leaning

Parting Thoughts: Linear algebra and optimization are powerful and widely useful

E.g. in class we saw applications in

circuits

dynamical systems

graph theory

computer graphics

recommendation systems

population genetics

text analysis

web search

healthcare

finance

politics

And there's much, much more out there!