15451 Spring 2023

Zero-sum games

Elaine Shi

What is a game?

Chess, checkers, poker, tennis, football

Game theory is everywhere:

- Behavior of people on social networks
- Routing in large networks
- Behavior of miners and users in blockchains
- Complexity theory, cryptography

……

"**Game theory** is the study of mathematical models of strategic interactions among rational agents."

--- Wikipedia

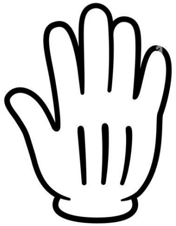
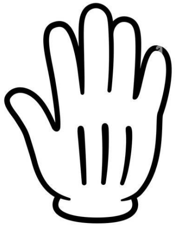
Today: 2-player zero-sum games

Example: shooter-goalie game

payoff
utility

		goalie	
		L	R
shooter	L	$(-1, 1)$	$(1, -1)$
	R	$(1, -1)$	$(-1, 1)$

Example: rock, paper, scissors

			
	0, 0	-1, 1	1, -1
	1, -1	0, 0	-1, 1
	-1, 1	1, -1	0, 0

More generally, view a 2-player game as matrix

$$M_{i,j} = (R_{i,j}, C_{i,j})_{i,j}$$

- Players
- Actions
- Payoffs

$\boxed{\text{Zero-sum:}}$ $R_{i,j} + C_{i,j} = 0$

Zero-sum game simplified using row player's matrix

		goalie L	goalie R
shooter	L	-1	1
	R	1	-1

Pure strategy: the player **deterministically** selects a single action

Pure strategy: the player deterministically selects a single action

		goalie	
		L	R
shooter	L	-1	1
	R	1	-1

L, L

L, R

shooter goalie

If shooter always shoots left, the goalie can always move left

Pure strategy: the player deterministically selects a single action

			
	0, 0	-1, 1	1, -1
	1, -1	0, 0	-1, 1
	-1, 1	1, -1	0, 0

Mixed strategy: a probability distribution of actions

p_i: probability of action i for row player

q_i: probability of action i for column player

$$\sum_i p_i = 1, \quad \sum_i q_i = 1$$

Mixed strategy: a probability distribution of actions

p_i: probability of action i for row player

q_i: probability of action i for column player

$$\mathbf{p} := (p_i)_i, \quad \mathbf{q} := (q_i)_i$$

Expected payoff to row player given mixed strategies **p** and **q**

$$V_R(\mathbf{p},\mathbf{q}) := \sum_{i,j} \Pr[\text{row player plays } i \text{ and column player plays } j] \cdot R_{ij} = \sum_{ij} p_i q_j R_{ij},$$

- Expected payoff to row player given mixed strategies **p** and **q**

$$V_R(\mathbf{p},\mathbf{q}) := \sum_{i,j} \Pr[\text{row player plays } i \text{ and column player plays } j] \cdot R_{ij} = \sum_{ij} p_i q_j R_{ij},$$

- Expected payoff to column player given mixed strategies **p** and **q**

$$V_C(\mathbf{p},\mathbf{q}) := \sum_{i,j} p_i q_j C_{ij}$$

- Expected payoff to row player given mixed strategies **p** and **q**

$$V_R(\mathbf{p},\mathbf{q}) := \sum_{i,j} \Pr[\text{row player plays } i \text{ and column player plays } j] \cdot R_{ij} = \sum_{ij} p_i q_j R_{ij},$$

- Expected payoff to column player given mixed strategies **p** and **q**

$$V_C(\mathbf{p},\mathbf{q}) := \sum_{i,j} p_i q_j C_{ij}$$

- **Zero sum:** $V_C(\mathbf{p},\mathbf{q}) = -V_R(\mathbf{p},\mathbf{q})$

Example of mixed strategy and payoff

$\frac{1}{3}$		0, 0	-1, 1	1, -1
$\frac{1}{3}$		1, -1	0, 0	-1, 1
$\frac{1}{3}$		-1, 1	1, -1	0, 0

Lower bound for the row player

maximin strategy

$$\text{lb} := \overbrace{\max_{\mathbf{p}} \underbrace{\min_{\mathbf{q}} V_R(\mathbf{p}, \mathbf{q})}_{\text{payoff when opponent plays the optimal response against our choice p}}}^{\text{mixed strategy that maximizes the minimum expected payoff}}$$

$q^* = \text{argmin}$

Lower bound for the row player

$$\mathsf{lb} := \max_{\mathbf{p}} \min_{\mathbf{q}} V_R(\mathbf{p}, \mathbf{q})$$

p^*

Row player can guarantee this payoff for itself

Upper bound for the column player

mini-max strategy

$$\mathrm{ub} := \min_{\mathbf{q}} \underbrace{\max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q})}$$

Upper bound for the column player

$$\mathrm{ub} := \min_{\mathbf{q}} \max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q})$$

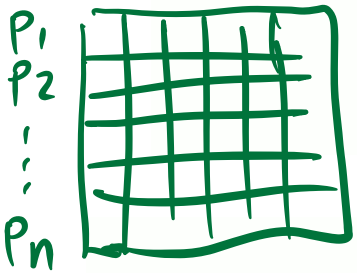
Column player can guarantee that the row player does not get more than this

Claims:

earlier defn: $\text{lb} = \max_{p} \min_{q} \sum_{i,j} p_i q_j R_{ij}$

mixed strategy

$$\text{lb} = \max_{\mathbf{p}} \min_{j} \sum_{i} p_i R_{ij}$$



pure strategy

jth column

$$\text{ub} = \min_{\mathbf{q}} \max_{i} \sum_{j} q_j R_{ij}$$

Example: Shooter-Goalie Game

Suppose $\mathbf{p} = (\overset{p}{p_1}, \overset{1-p}{p_2})$ for **shooter**

$p_1 + p_2 = 1$

- If goalie plays L, shooter's expected payoff:

$$(-1)\cdot p + 1\cdot(1-p) = 1-2p$$

- If goalie plays R, shooter's expected payoff:

$$1\cdot p + (-1)(1-p) = p-1+p = 2p-1$$

$$\max_p \min(1-2p, 2p-1)$$

$$p = \frac{1}{2}$$

Example: Shooter-Goalie Game

Suppose $\mathbf{p} = (p_1, p_2)$ for **shooter**

- If goalie plays L, shooter's expected payoff:
- If goalie plays R, shooter's expected payoff:

Example: Shooter-Goalie Game

Suppose **q** = (q_1, q_2) for **goalie**

minimax

$\left(\frac{1}{2}, \frac{1}{2}\right)$

- If shooter plays L, goalie's expected payoff:
- If shooter plays R, shooter's expected payoff:

$$\text{lb} = \max_{\mathbf{p}} \min_{\mathbf{q}} V_R(\mathbf{p}, \mathbf{q})$$

$\boxed{\text{lb} = \text{ub}}$

Maximin strategy **p***

$$\min_{\mathbf{q}} \max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q}) = \text{ub}.$$

Minimax strategy **q***

One direction is easy to prove: lb <= ub

What happens if the players play (**p***, **q***)?

- Row player gets at least **lb**
- Column player ensures row player gets at most **ub**

- **Thus, lb <= ub**

is it also true that $ub \le lb$?

Minimax Theorem (von Neumann, 1928)

For any finite 2-player 0-sum game:

$$\text{lb} = \max_{\mathbf{p}} \min_{\mathbf{q}} V_R(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{q}} \max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q}) = \text{ub}.$$

Minimax Theorem (von Neumann, 1928)

For any finite 2-player 0-sum game:

$$\mathsf{lb} = \max_{\mathbf{p}} \min_{\mathbf{q}} V_R(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{q}} \max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q}) = \mathsf{ub}.$$

- Called "**the value of the game**"

Minimax Theorem (von Neumann, 1928)

For any finite 2-player 0-sum game:

$V =$

$$\text{lb} = \max_{\mathbf{p}} \min_{\mathbf{q}} V_R(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{q}} \max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q}) = \text{ub}.$$

- Called "**the value of the game**"
- **($\mathbf{p}^*$, $\mathbf{q}^*$)** is one Nash equilibrium

Minimax Theorem (von Neumann, 1928)

For any finite 2-player 0-sum game:

$$\mathrm{lb} = \max_{\mathbf{p}} \min_{\mathbf{q}} V_R(\mathbf{p}, \mathbf{q}) = \min_{\mathbf{q}} \max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q}) = \mathrm{ub}.$$

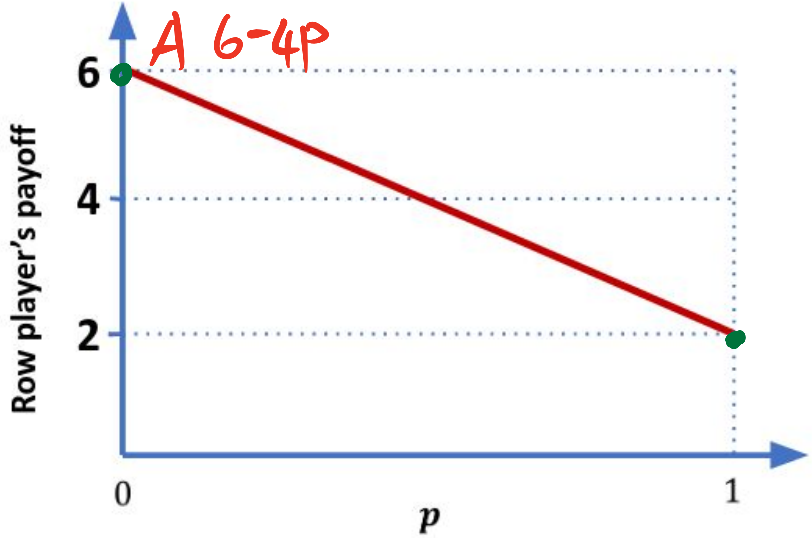
- Called "**the value of the game**"
- **($\mathbf{p}^*$, $\mathbf{q}^*$)** is one Nash equilibrium
- $\mathbf{p}^*$ is a best response to $\mathbf{q}^*$ and vice versa

General method for solving 2-row games

		column player		
		A	**B**	**C**
row	**1**	2	6	3
player	**2**	6	2	4

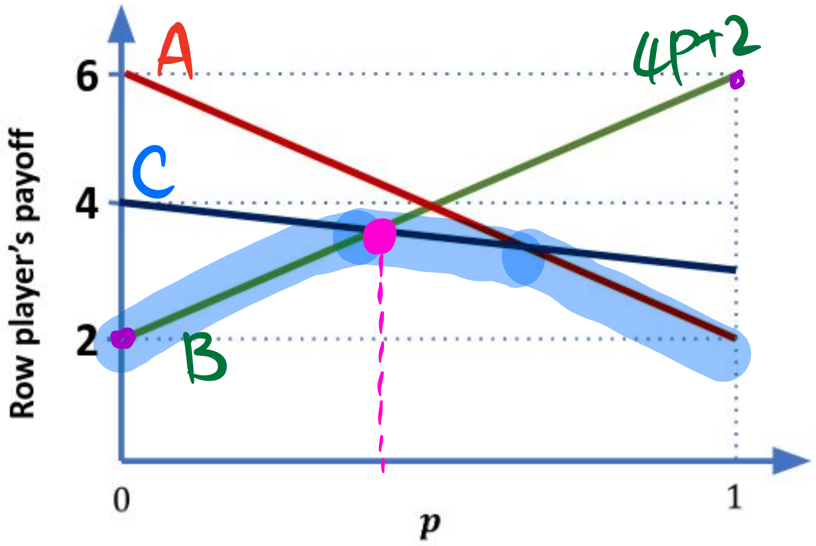
		column player		
		A	B	C
row p	1	2	6	3
player $1-p$	2	6	2	4

$2p+6\cdot(1-p)=6-4p$



			column player	
		A	B	C
row p	**1**	2	6	3
player $1-p$	**2**	6	2	4

$$6 \cdot p + 2(1-p) = 4p+2$$



		column player		
		A	B	C
row	**1**	2	6	3
player	**2**	6	2	4

maximin strat. for row player:

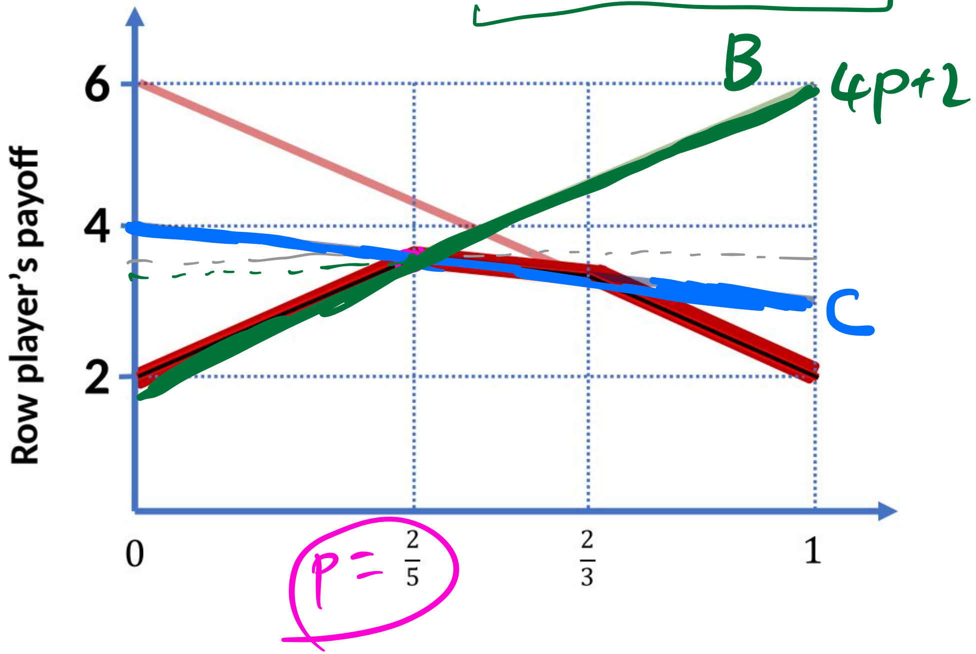
$p = \frac{2}{5}$

minimax strat for col player:

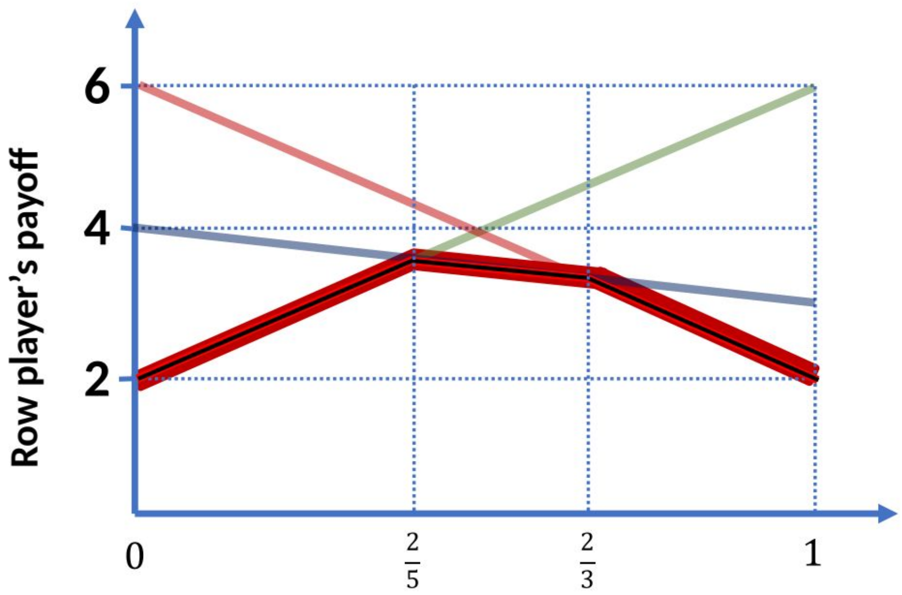
$$\underbrace{\frac{4}{5}C + \frac{1}{5}B}$$

$4 \cdot \frac{2}{5} + 2$

$= \frac{8}{5} + 2 = 3 + \frac{3}{5}$



What is a good strategy for the column player?

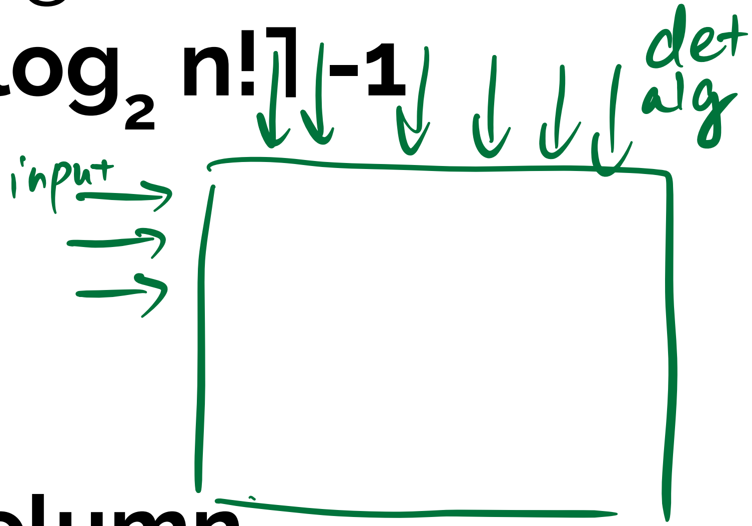


Application: lower bounds for randomized algorithms

Theorem: for any **randomized** comparison-based sorting alg A, there exists an input on which A performs **$\lceil \log_2 n! \rceil - 1$** comparisons **in expectation**

Want to lower bound: **expected cost of rand. alg on worst-case input**

Theorem: for any **randomized** comparison-based sorting alg A, there exists an input on which A performs $\lceil \log_2 n! \rceil - 1$ comparisons **in expectation**



Each input: **one row**

Each det. alg: **one column**

Randomized alg: **mixed strategy for column player**

q

For any rand. alg denoted **q'**

expected time of the best randomized alg over its worst-case input

$$\underbrace{\max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q}')}_{\text{expected time for worst-case input for } q'} \geq \underbrace{\min_{\mathbf{q}} \max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q})}$$

$$= \underbrace{\max_{\mathbf{p}} \min_{\mathbf{q}} V_R(\mathbf{p}, \mathbf{q})}$$

by minimax thm

$$\geq \underbrace{\min_{\mathbf{q}} V_R(\mathbf{p}^*, \mathbf{q})}$$

$\forall p^*$

For any rand. alg denoted **q'**

$$
\begin{aligned}
\max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q}') &\geq \min_{\mathbf{q}} \max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q}) \\
&= \max_{\mathbf{p}} \min_{\mathbf{q}} V_R(\mathbf{p}, \mathbf{q}) \\
&\geq \min_{\mathbf{q}} V_R(\mathbf{p}^*, \mathbf{q})
\end{aligned}
$$

hard distr

det

Construct **p*** s.t. for any **q**, V_R(**p***, **q**) is large

For any rand. alg denoted **q'**

$$\max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q}') \geq \min_{\mathbf{q}} \max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q})$$
$$= \max_{\mathbf{p}} \min_{\mathbf{q}} V_R(\mathbf{p}, \mathbf{q})$$
$$\geq \min_{\mathbf{q}} V_R(\mathbf{p}^*, \mathbf{q})$$

Construct a hard distr. **p*** over inputs, s.t. the best det. alg has large running time over a random input from **p***

For any rand. alg denoted **q'**

$$\max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q}') \geq \min_{\mathbf{q}} \max_{\mathbf{p}} V_R(\mathbf{p}, \mathbf{q})$$

$$= \max_{\mathbf{p}} \min_{\mathbf{q}} V_R(\mathbf{p}, \mathbf{q})$$

$$\geq \min_{\mathbf{q}} V_R(\mathbf{p}^*, \mathbf{q})$$

Construct **p*** s.t. for any **q**, V_R(**p***, **q**) is large

Pick **p*** to be the uniform distribution over all inputs

leaves = n!

Claim: for any det. alg **q**, think of the alg as a decision tree. The average depth of all leaf nodes is at least $\lceil \log_2 n! \rceil - 1$

Pick **p^*** to be the uniform distribution over all inputs

Claim: for any det. alg **q**, think of the alg as a decision tree. The average depth of all leaf nodes is at least **$\lceil \log_2 n! \rceil - 1$**

- Decision tree has n! Leaves.
- Mapping between leaves and inputs n!
- **Average depth to leaves is maximized when the tree is "somewhat balanced"**

Claim: T is a tree with $n!$ leaves,

suppose $\exists$ leaf nodes x, y s.t. $D(x) \geq D(y) + 2$

$\Rightarrow$ construct another tree T^*. also with $n!$ leaves s.t. avg depth to leaves of T^*

$<$ avg depth to leaves of T

T

$2^{k+1} \geq n!$

$k+1 \geq \log_2 n!$

Consider a tree with $n!$ leaves s.t. every leaf has depth k or $k+1$

Yao's Minimax Principle

expected cost of a randomized algorithm on its worst-case input

$$\geq$$

cost of the best deterministic algorithm on a random input from some distribution

Also used for proving comm. complexity of randomized protocols

Next lectures:

Linear Programming

(prove the minimax theorem)