Games 2:
TD learning, Simultaneous games, Non-zerosum games

Hwanjo Yu

POSTECH

http://di.postech.ac.kr/hwanjoyu

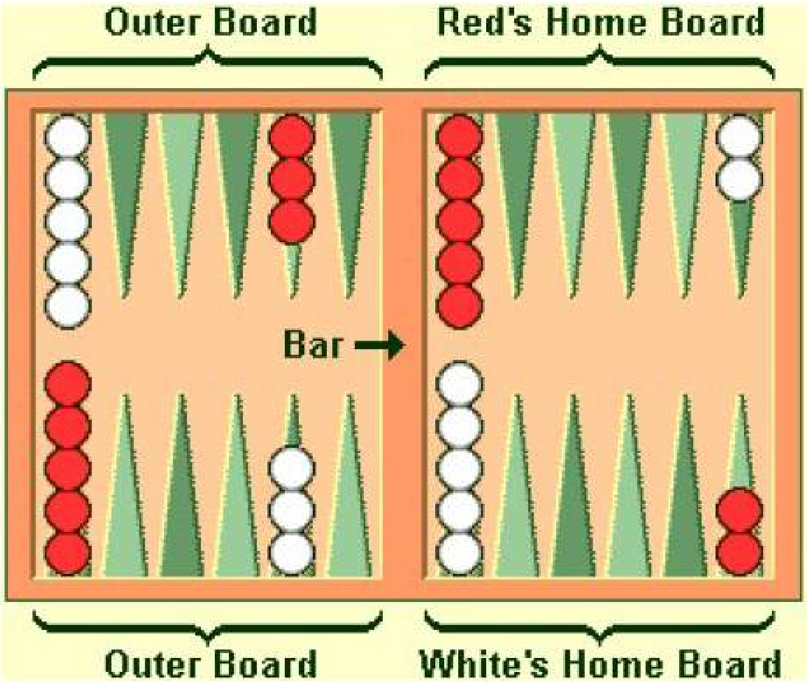
Roadmap

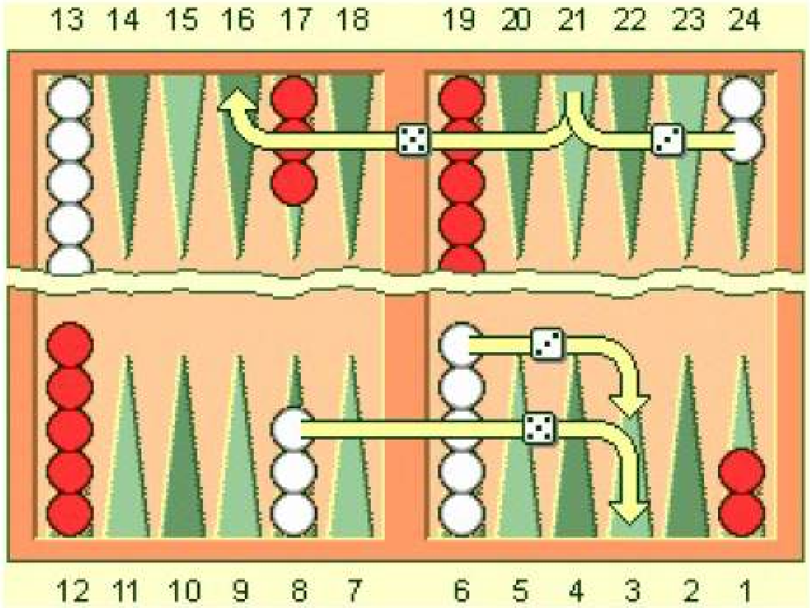
TD learning

Simultaneous games

Non-zero-sum games

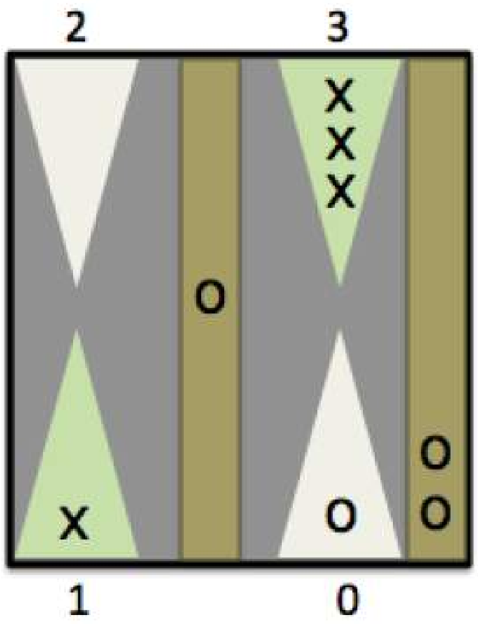
Example: Backgammon





Features for Backgammon

state s



Features $\phi(s)$:

- $[(\# \text{ o in column } 0) = 1]$: 1
- $[(\# \text{ o in bar})]$: 1
- $[(\text{fraction o removed})]$: 1/2
- $[(\# \text{ x in column } 1) = 1]$: 1
- $[(\# \text{ x in column } 3) = 3]$: 1
- $[(\text{is it o's turn})]$: 1

Generating data

Generate using policies based on current $V(s;\mathbf{w})$:

$$s_0; a_1, r_1, s_1; a_2, r_2, s_2; a_3, r_3, s_3; \ldots; a_n, r_n, s_n$$

- $\pi_{\text{agent}}(s;\mathbf{w}) = \arg\max_{a \in \text{Actions}(s)} V(\text{Succ}(s,a);\mathbf{w})$
- $\pi_{\text{opp}}(s;\mathbf{w}) = \arg\min_{a \in \text{Actions}(s)} V(\text{Succ}(s,a);\mathbf{w})$

Note: no need to randomize (ϵ-greedy) since the game is already stochastic (dice)!

Learning algorithm

Episode (generated according to π_{agent} and π_{opp}):

$$s_0; a_1, r_1, s_1; a_2, r_2, s_2; a_3, r_3, s_3; \ldots; a_n, r_n, s_n$$

A small piece of experience:

$$(s, a, r, s')$$

Prediction:

$$V_\pi(s; \mathbf{w})$$

Target:

$$r + \gamma V_\pi(s'; \mathbf{w})$$

General framework

Objective function:

$$\frac{1}{2}(\text{prediction}(\mathbf{w}) - \text{target})^2$$

Gradient:

$$(\text{prediction}(\mathbf{w}) - \text{target})\nabla_{\mathbf{w}}\text{prediction}(\mathbf{w})$$

Update:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta\underbrace{(\text{prediction}(\mathbf{w}) - \text{target})\nabla_{\mathbf{w}}\text{prediction}(\mathbf{w})}_{\text{gradient}}$$

Temporal difference (TD) learning

Algorithm: TD learning

- On each (s, a, r, s'):

$$\mathbf{w} \leftarrow \mathbf{w} - \eta[\underbrace{V_\pi(s;\mathbf{w})}_{\text{prediction}} - \underbrace{(r + \gamma V_\pi(s';\mathbf{w}))}_{\text{target}}]\nabla_{\mathbf{w}} V_\pi(s;\mathbf{w})$$

For linear functions:

- $V(s;\mathbf{w}) = \mathbf{w} \cdot \phi(s)$
- $\nabla_{\mathbf{w}} V(s;\mathbf{w}) = \phi(s)$

Example of TD learning

Step size $\eta = 0.5$, discount $\gamma = 1$, reward is end utility

S1	**r:0**	**S4**	**r:0**	**S8**	**r:1**	**S9**
$\phi: \binom{0}{1}$		$\phi: \binom{1}{0}$		$\phi: \binom{1}{2}$		$\phi: \binom{1}{0}$
$\mathbf{w}: \binom{0}{0}$	p:0 t:0 p-t:0	$\mathbf{w}: \binom{0}{0}$	p:0 t:0 p-t:0	$\mathbf{w}: \binom{0}{0}$	p:0 t:1 p-t:-1	$\mathbf{w}: \binom{0.5}{1}$

S1	**r:0**	**S2**	**r:0**	**S6**	**r:0**	**S9**
$\phi: \binom{0}{1}$		$\phi: \binom{1}{0}$		$\phi: \binom{0}{0}$		$\phi: \binom{1}{0}$
$\mathbf{w}: \binom{0.5}{1}$	p:1 t:0.5 p-t:0.5	$\mathbf{w}: \binom{0.5}{0.75}$	p:0.5 t:0 p-t:0.5	$\mathbf{w}: \binom{0.25}{0.75}$	p:0 t:0.25 p-t:-0.25	$\mathbf{w}: \binom{0.25}{0.75}$

Comparison

Algorithm: TD learning

- On each (s, a, r, s'):

$$\mathbf{w} \leftarrow \mathbf{w} - \eta[\underbrace{\hat{V}_\pi(s;\mathbf{w})}_{\text{prediction}} - \underbrace{(r + \gamma \hat{V}_\pi(s';\mathbf{w}))}_{\text{target}}]\nabla_\mathbf{w}\hat{V}_\pi(s;\mathbf{w})$$

Algorithm: Q-learning (a kind of off-policy TD learning)

- On each (s, a, r, s'):

$$\mathbf{w} \leftarrow \mathbf{w} - \eta[\underbrace{\hat{Q}_{\text{opt}}(s,a;\mathbf{w})}_{\text{prediction}} - \underbrace{(r + \gamma \max_{a' \in \text{Actions}(s)} \hat{Q}_{\text{opt}}(s',a';\mathbf{w}))}_{\text{target}}]\nabla_\mathbf{w}\hat{Q}_{\text{opt}}(s,a;\mathbf{w})$$

Comparison

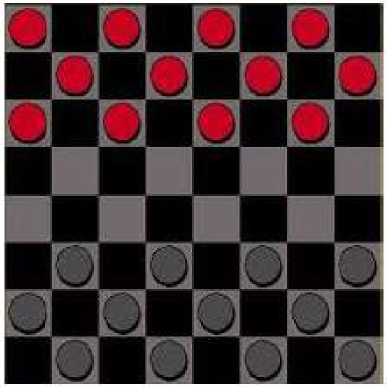
Q-learning:

- Operate on $\hat{Q}_{\text{opt}}(s, a; \mathbf{w})$
- Off-policy: value is based on estimate of optimal policy
- To use, don't need to know MDP transitions $T(s, a, s')$

TD learning:

- Operate on $\hat{V}_{\pi}(s; \mathbf{w})$
- On-policy: value is based on exploration policy (usually based on $\hat{V}_{\pi}$)
- To use, need to know rules of the game $\text{Succ}(s, a)$

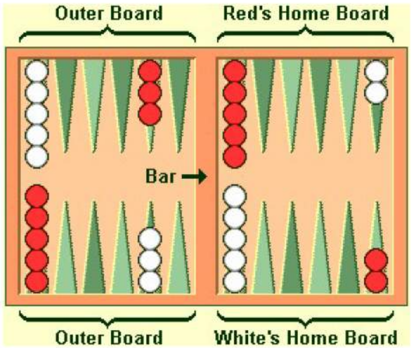
Learning to play checkers

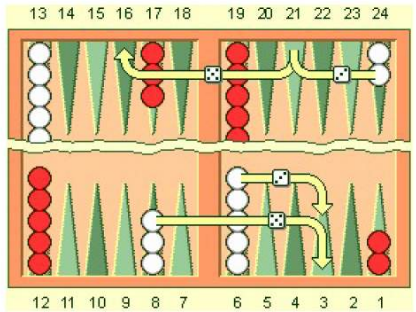


Arthur Samuel's checkers program [1959]:

- Learned by playing itself repeatedly (self-play)
- Smart features, linear evaluation function, use intermediate rewards
- Used alpha-beta pruning + search heuristics
- Reach human amateur level of play
- IBM 701: 9K of memory!

Learning to play Backgammon

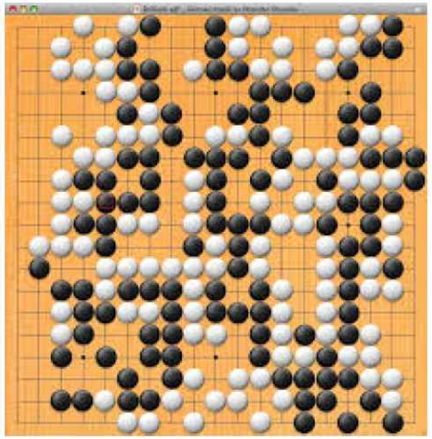




Gerald Tesauro's TD-Gammon [1992]:

- Learned weights by playing itself repeatedly (1 million times)
- Dumb features, neural network, no intermediate rewards
- Reached human expert level of play, provided new insights into opening

Learning to play Go



AlphaGo Zero (2017)

- Learned by self play (4.9 million games)
- Dumb features (stone positions), neural network, no intermediate rewards, Monte Carlo Tree Search
- Beat AlphaGo, which beat Le Sedol in 2016
- Provided new insights into the game

Summary so far

- Parametrize evaluation functions using features

- TD learning: learn an evaluation function

$$(\text{prediction}(\mathbf{w}) - \text{target})^2$$

Up next:

Turn-based => Simultaneous

Zero-sum => Non-zero-sum

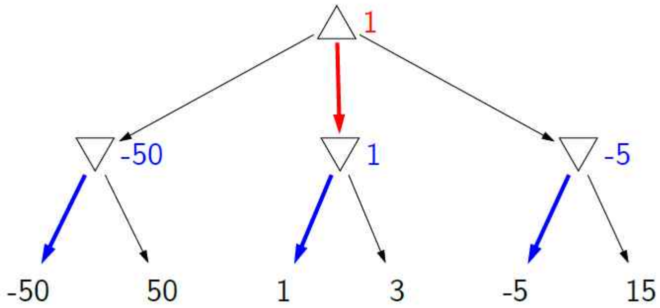
Roadmap

TD learning

Simultaneous games

Non-zero-sum games

- Turn-based games



- Simultaneous games:

Two-finger Morra

Example: two-finger Morra

- Players A and B each show 1 or 2 fingers.
- If both show 1, B gives A 2 dollars.
- If both show 2, B gives A 4 dollars.
- Otherwise, A gives B 3 dollars.

Payoff matrix

Definition: single-move simultaneous game

- Players = {A, B}
- Actions: possible actions
- $V(a, b)$: **A's utility** if A chooses action a, B chooses b (let V be **payoff matrix**)

Example: two-finger Morra payoff matrix

A \ B	1 finger	2 fingers
1 finger	2	-3
2 fingers	-3	4

Strategies (policies)

Definition: pure strategy (= deterministic policy)

- A pure strategy is a single action: $a \in \text{Actions}$

Definition: mixed strategy (= stochastic policy)

- A mixed strategy is a probability distribution: $0 \leq \pi(a) \leq 1$ for $a \in \text{Actions}$

Example: two-finger Morra strategies

- Always 1: π = [1, 0]
- Always 2: π = [0, 1]
- Uniformly random: π = [1/2, 1/2]

Game evaluation

Definition: game evaluation

- The **value** of the game if player A follows π_{A} and player B follows π_{B} is

$$V(\pi_{\mathrm{A}}, \pi_{\mathrm{B}}) = \sum_{a,b} \pi_{\mathrm{A}}(a)\pi_{\mathrm{B}}(b)V(a,b)$$

Example: two-finger Morra

- Player A always chooses 1: π_{A} = [1, 0]
- Player B picks randomly: π_{B} = [1/2, 1/2]
- Value: $-\frac{1}{2}$

How to optimize?

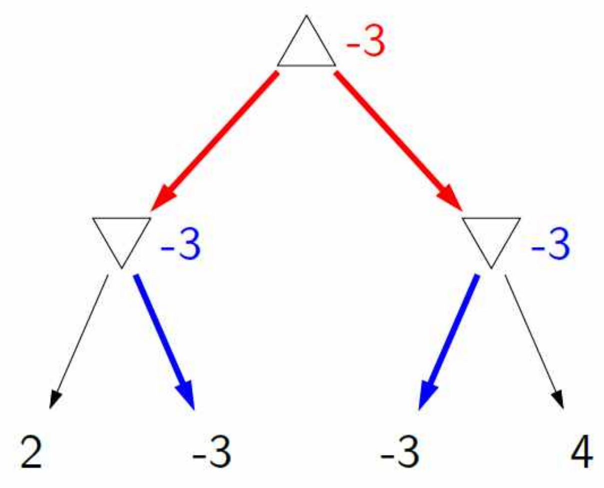
- Game value:

$$V(\pi_{\mathrm{A}}, \pi_{\mathrm{B}})$$

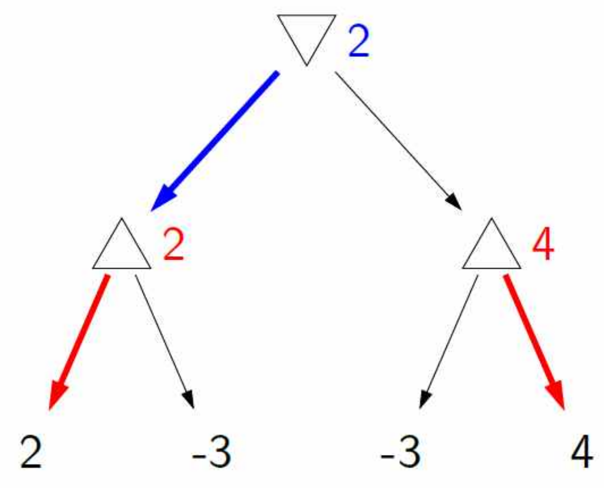
- Challenge: player A wants to maximize, player B wants to minimize simultaneously

Pure strategies: who goes first?

Player A goes first:



Player B goes first:



Proposition: going second is no worse

$$\max_a \min_b V(a,b) \leq \min_b \max_a V(a,b)$$

Mixed strategies

Example: two-finger Morra

- Player A reveals: $\pi_{\mathrm{A}} = \left[\frac{1}{2}, \frac{1}{2}\right]$
- Value $V(\pi_{\mathrm{A}}, \pi_{\mathrm{B}}) = \pi_{\mathrm{B}}(1)\left(-\frac{1}{2}\right) + \pi_{\mathrm{B}}(2)(+\frac{1}{2})$ (convex combination)
- Optimal strategy for player B is $\pi_{\mathrm{B}} = [1,0]$ (**pure!**)

Proposition: second player can play pure strategy

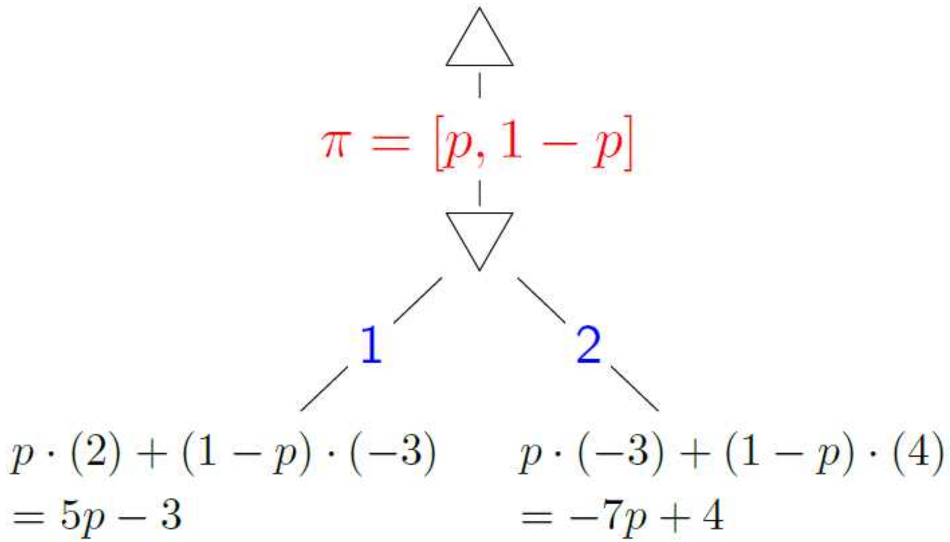
- For any fixed mixed strategy π_{A}:

$$\min_{\pi_{\mathrm{B}}} V(\pi_{\mathrm{A}}, \pi_{\mathrm{B}})$$

  can be attained by a **pure strategy**

Mixed strategies

Player A first reveals his/her mixed strategy

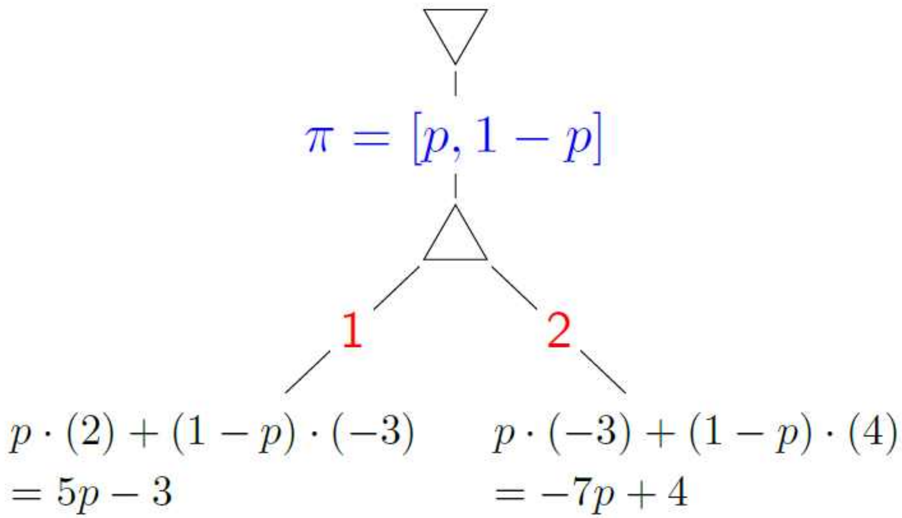


Best strategy for A?

$$\max_{0 \le p \le 1} \min\{5p - 3, -7p + 4\} = -\frac{1}{12} (\text{with } p = \frac{7}{12})$$

Mixed strategies

Player B first reveals his/her mixed strategy



Best strategy for B?

$$\min_{p \in [0,1]} \max\{5p - 3, -7p + 4\} = -\frac{1}{12} \text{ (with } p = \frac{7}{12}\text{)}$$

General theorem

Theorem: minimax theorem [von Neumann, 1928]

- For every simultaneous two-player zero-sum game with a finite number actions:

$$\max_{\pi_{\mathrm{A}}} \min_{\pi_{\mathrm{B}}} V(\pi_{\mathrm{A}}, \pi_{\mathrm{B}}) = \min_{\pi_{\mathrm{B}}} \max_{\pi_{\mathrm{A}}} V(\pi_{\mathrm{A}}, \pi_{\mathrm{B}})$$

  where $\pi_{\mathrm{A}}, \pi_{\mathrm{B}}$ range over **mixed strategies**.

- Revealing your mixed optimal strategy doesn't hurt you!
- Both ordering of the players yields the same answer.

Roadmap

TD learning

Simultaneous games

Non-zero-sum games

Utility functions

- Competitive games: minimax (linear programming)

- Collaborative games: pure maximization (plain search)

- Real life: ?

Prisoner’s dilemma

Example: Prisoner's dilemma

- Prosecutor asks A and B individually if each will testify against the other.
- If both testify, then both are sentenced to 5 years in jail.
- If both refuse, then the sentence is only 1 year.
- If only one testifies, then he/she gets out for free; the other gets a 10-year sentence.

Prisoner's dilemma

Example: payoff matrix

A\B	testify	refuse
testify	$A = -5, B = -5$	$A = 0, B = -10$
refuse	$A = -10, B = 0$	$A = -1, B = -1$

Definition: payoff matrix

- Let $V_p(\pi_{\mathrm{A}}, \pi_{\mathrm{B}})$ be the utility for player p
- Best strategy for A?
- $V_A(\pi_{\mathrm{A}}, \pi_{\mathrm{B}}) = \pi_{\mathrm{A}}(1)\pi_{\mathrm{B}}(1)(-5) + \pi_{\mathrm{A}}(1)\pi_{\mathrm{B}}(2)(-) + \pi_{\mathrm{A}}(2)\pi_{\mathrm{B}}(1)(-10) + \pi_{\mathrm{A}}(2)\pi_{\mathrm{B}}(2)(-1)$
  $= \pi_{\mathrm{B}}(1)[-5\pi_{\mathrm{A}}(1) - 10\pi_{\mathrm{A}}(2)] + \pi_{\mathrm{B}}(2)[0\pi_{\mathrm{A}}(1) - 1\pi_{\mathrm{A}}(2)]$

Nash equilibrium

Can't apply von Neumann's minimax theorem (not zero-sum), but get something weaker:

Definition: Nash equilibrium (a stable point)

- A **Nash equilibrium** is $(\pi_{\mathrm{A}}^*, \pi_{\mathrm{B}}^*)$ such that no player has an incentive to change his/her strategy:

$$V_{\mathrm{A}}(\pi_{\mathrm{A}}^*, \pi_{\mathrm{B}}^*) \geq V_{\mathrm{A}}(\pi_{\mathrm{A}}, \pi_{\mathrm{B}}^*) \text{ for all } \pi_{\mathrm{A}}$$

$$V_{\mathrm{B}}(\pi_{\mathrm{A}}^*, \pi_{\mathrm{B}}^*) \geq V_{\mathrm{B}}(\pi_{\mathrm{A}}^*, \pi_{\mathrm{B}}) \text{ for all } \pi_{\mathrm{B}}$$

Theorem: Nash's existence theorem [1950]

- In any finite-player game with finite number of actions, there exists **at least one** Nash equilibrium.

Examples of Nash equilibria

Example: Two-finger Morra

- Nash equilibrium: A and B both play $\pi = \left[\frac{7}{12}, \frac{5}{12}\right]$.

Example: Collaborative two-finger Morra

- Two Nash equilibria:
  - A and B both play 1 (value is 2).
  - A and B both play 2 (value is 4).
- For purely collaborative games, the equilibria are simply the entries of the payoff matrix for which no other entry in the row or column are larger.

Example: Prisoner's dilemma

- Nash equilibrium: A and B both testify.

Summary

Simultaneous zero-sum games:

- von Neumann's minimax theorem
- Multiple minimax strategies, single game value

Simultaneous non-zero-sum games:

- Nash's existence theorem
- Multiple Nash equilibria, multiple game values

Huge literature in game theory / economics