Multi-Agent Systems

Christos Dimitrakakis

April 29, 2024

Outline

Multi-agent decision making

- Two versus n-player games
- Co-operative games
- Zero-sum games
- General-sum games
- Stochastic games
- Partial information games

Rock/Paper/Scissors

- Number of players: 2
- Zero-sum
- Deterministic
- Simultaneous move

Chess/Go/Checkers/Othello

- Number of players: 2
- Zero-sum
- Deterministic,
- Alternating, Full information

Backgammon

- Number of players: 2
- Zero-sum
- Stochastic
- Alernating, Full information

Poker/Blackjack

- Number of players: n
- Zero-sum
- Stochastic
- Alternating, Partial information

Doom/Quake/CoD

- Number of players: n
- Zero-sum
- Stochastic
- Simultaneous, Partial information

Auctions

- Number of players: n
- General sum
- Deterministic
- Simultaneous move

Human preferences

- These are typically unknown
- They might not be expressible in mathematical form
- Nevertheless, we make the utility assumption

AI preferences

- These are typically known

Human-AI examples

Normal form

In the table below, we see how much reward each player obtains for every combination of actions

ρ^1, ρ^2	b = 0	b = 1
a = 0	2, 1	4, 0
a = 1	1, 0	3, 1

Simultaneous moves

We assume that each player is playing without seeing the move of the other player.

Commitment

However, we can also look at commitment or Stackleberg games, where one player either *commits* to playing a move, or plays before the other player.

Information structure

For other types of move sequencing, we have to encode the information structure of a game as a graph.

Basic concepts in normal form games

ρ^1, ρ^2	b = 0	b = 1
a = 0	2, 1	4, 0
a = 1	1, 0	3, 1

Domination and best response

- $b = 1$ is a best response to $a = 1$, i.e. $\rho^2(1,1) > \rho^2(1,0)$ \]
- $a = 0$ is a strictly dominant strategy. Given any b, it is strictly better to play $a = 0$, i.e. $\rho^1(0, b) > \rho^1(1, b)$.
- If a pair (a, b) is *not dominated*, then it is Pareto-efficient.

Questions

- How much reward can a obtain?
- Does b have a dominant strategy?
- Does this take into account what b likes?

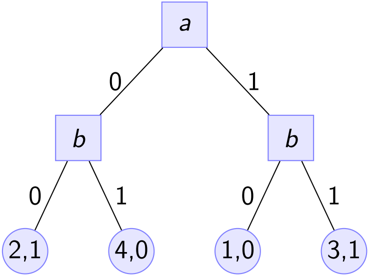
Commitment

Let us see what happens when one player commits to a move

ρ^1, ρ^2	b = 0	b = 1
a = 0	2, 1	4, 0
a = 1	1, 0	3, 1

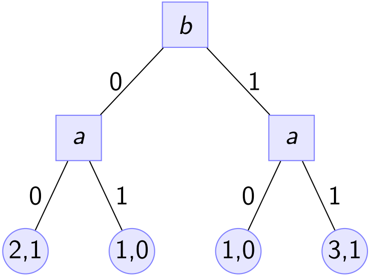
Player *a* is first

- What should *b* play?
- What is *a*'s best move?



Player *b* is first

- What should *a* play in each case?



Fully collaborative games

In team games, $\rho^i = \rho^j$ for all players i, j.

One-shot alternating move 2-player games

- Player 1 plays a
- Player 2 plays b
- Player 1 obtains $\rho^1(a, b)$
- Player 2 obtains $\rho^2(a, b)$

Extensive-form alternating-move zero sum games

- At time t:
- The state is s_t, players receive rewards $\rho(s_t), -\rho(s_t)$
- Player chooses action a_t, which is revealed.
- The state changes to s_{t+1}, and is revealed.
- Players receive reward $\rho(s_{t+1}), -\rho(s_{t+1})$
- Player chooses action b_{t+1}.
- The state changes to s_{t+2}.
- Player a receives $\rho(s_t)$ and b receives $-\rho(s_t)$.

The utility for player a is

$$U^1 = \sum_t \rho(s_t),$$

while for b it is

$$U^2 = -\sum_t \rho(s_t)$$

Backwards induction for Alternating Zero Sum Games

Let π_1 and π_2 be the policies of each player and π the joint policy.

The value function of a policy $\pi = (\pi_1, \pi_2)$

For the utility of player 1, we get:

$$\begin{aligned} V_t^{1,\pi}(s) &\triangleq \mathbb{E}_\pi[U_t^1 \mid s_t = s] = \rho(s) + \mathbb{E}[U_{t+1}^1 \mid s_t = s] && (1)\\ &= \rho(s) + \sum_a \pi(a \mid s) \sum_j V_{t+1}^{1,\pi}(j) P(j \mid s, a) && (2) \end{aligned}$$

$$V_{t+1}^{1,\pi}(j) = \rho(j) + \sum_b \pi(b \mid j) \sum_j V_{t+2}^{1,\pi}(j) P(k \mid j, b) \qquad (3)$$

We can define the optimal value function analogously to MDPs, but player 2 is minimising

$$\begin{aligned} V_t^{1,*}(s) &= \max_{\pi_1} \min_{\pi_2} \mathbb{E}_\pi[U_t^+ \mid s_t = s] && (4)\\ &= \rho(s) + \max_a \sum_j V_{t+1}^{1,*}(j) P(j \mid s, a) && (5) \end{aligned}$$

$$V_{t+1}^{1,*}(j) = \rho(j) + \min_b \sum_j V_{t+1}^{1,*}(j) P(k \mid j, b) \qquad (6)$$

Normal-form simultaneous-move zero-sum games

(Also called minimax games)

- Player a chooses action a in secret.
- Player b chooses action b in secret.
- Players observe both actions
- Player a receives $\rho(a, b)$, and b receives $-\rho(a, b)$.

Mixed strategies

Each player chooses an action randomly, independently of one another:

$$\pi(a, b) = \pi_1(a)\pi_2(b)$$

π_i is called a mixed strategy.

Optimal strategies for zero-sum games

The value of a game

The expected value of the game for the first player is

$$V(\pi_1, \pi_2) \triangleq \sum_{a,b} \pi_1(a)\rho(a, b)\pi_2(b) = \boldsymbol{\pi}_1^\top \boldsymbol{R}\boldsymbol{\pi}_2,$$

where $\boldsymbol{\pi}_i$ is the vector form representation of i's strategy.

The value of the game

Any zero-sum game has at least one solution π^* over mixed strategies so that

$$U(\pi_1^*, \pi_2^*) = \max_{\pi_1} \min_{\pi_2} U(\pi_1, \pi_2) = \min_{\pi_2} \max_{\pi_1} U(\pi_1, \pi_2)$$

The problem can be solved through linear programming
The idea is to set find a policy corresponding to the greatest lower bound (or lowest upper bound) on the value.

Linear programming solution for ZSG

linear programming problem

This is an optimisation problem with linear objective and constraints. In canonical form it is written as:

$$\min_x \ \theta^\top x, \qquad \text{s.t. } c^\top x \geq 0.$$

Primal formulation

Find the higest lower bound for player 1

$$\max_v \ v, \qquad \text{s.t. } (\boldsymbol{R}\boldsymbol{\pi}_2)_j \geq v \ \forall j, \ \sum_j \pi_2(j) = 1, \pi_2(j) \geq 0$$

Dual formulation

Find the lowest upper bound for player 2

$$\min_v \ v, \qquad \text{s.t. } (\boldsymbol{\pi}_1^\top \boldsymbol{R})_j \leq v \ \forall j, \ \sum_j \pi_1(j) = 1, \pi_1(j) \geq 0$$

Normal-form general sum games

Each player moves at the same time

Example: Chicken

ρ^1, ρ^2	turn	dare
turn	0, 0	-5, -1
dare	1, -5	-10, -10

Example: Prisoner's dilemma

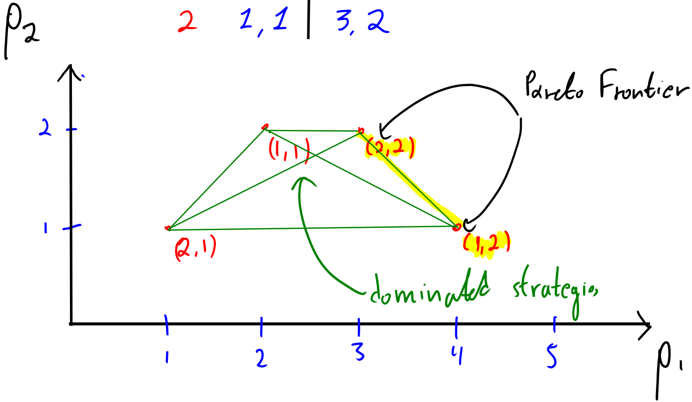
ρ^1, ρ^2	cooperate	defect
cooperate	0, 0	-5, -1
defect	1, -5	-10, -10

Example: penalty shot

ρ^1, ρ^2	kick left	kick right
dive left	1, -1	-1, 1
dive right	-1 1	1, -1

Pareto-Optimality

p_1, p_2		Player 2: 1	2
play.	1	2, 2	4, 1
	2	1, 1	3, 2



Nash equilibria

Computing Nash equlibria

Extensive-form general sum games

- At time t:
- The state is s_t, players receive rewards $\rho^i(s_t)$.
- Player $i = I(s_t)$ chooses an action.
- The state changes to s_{t+1}, and is revealed.

The utility for each player is

$$U^i = \sum_t \rho^i(s_t)$$

Backwards induction for Alternating General Sum Games

Let π_i be the policy of the i-th player and π the joint policy.

The value function of a policy $\pi = (\pi_i)_{i=1}^n$

For any player i, we can define their value at time t as:

$$V_t^{i,\pi}(s) \triangleq \mathbb{E}_\pi[U_t^i \mid s_t = s] \tag{7}$$

$$= \rho^i(s) + \sum_{a \in A} \pi_{I(s)}(a \mid s) \sum_j V_{t+1}^{1,\pi}(j) P(j \mid s, a) \tag{8}$$

Optimal policies

For perfect information games, we can use this recursion:

$$a_t^*(s) = \arg\max_{a \in A} \sum_j V_{t+1}^{I(s),*}(j) P(j \mid s, a) \tag{9}$$

$$V_t^{i,*} = \rho^i(s) + \sum_j V_{t+1}^{i,\pi}(j) P(j \mid s, a_t^*(s)) \qquad \forall i \tag{10}$$

This ensures that we update the values of all players at each step.