L24 – A Quick Introduction to Game Theory

AIMA4e: Ch 5.1–2, 5.4–5; Ch 18.1–3

What you should know after this lecture

- Two-player zero-sum complete information alternating games can be handled using similarly to MDPs
- Fully cooperative games can be handled as POMDPs with execution-time constraints

Problem setting

- Centralization: Is the process of selecting actions performed jointly for all the agents or separately?
  - Online centralization: basically, then, we just have a single-agent decision problem with multiple outputs
  - Offline centralization: we can run a single policy-optimization process that produces a set of policies for de-centralized execution
  - No centralization: agents make completely independent decisions
- Payoffs:
  - In an zero-sum (usually two-player) game, player A's loss is always exactly equal to player B's game.
  - In a fully-cooperative game, all players have the same payoff for all states/actions.
  - Otherwise, it is a general-sum game.
- Turn-taking:
  - In an alternating game, players take actions sequentially in a fixed order.
  - Otherwise, they may decide and execute in parallel.
  - Very little work lets them execute completely asynchronously.
- Observations: may have complete, partial, or no observability of
  - World state
  - Other agents' actions
- Communication
  - May have extra "side channels" for communication
  - Can understand actions taken by one agent and observed by another as communication

Some simple cases

- Centralized decision-making (and observations): really a path-search, MDP, or POMDP with large action space
- Decentralized decision-making but with centralized "planning/optimization" time: do some kind of policy optimization to find individual policies for the agents that perform well under decentralized execution.

Game theorists use "cooperative games" to handle much more complex situations in which players don't really have the same payoffs, but can make strategic alliances.

Mechanism design

Mechanism design is a way of setting up decision-making processes that have the property that if agents collectively make their decisions in that way, certain fairness-type outcomes will be guaranteed. Various types of auctions are an example of this.

Example: a sealed-bid second-price auction: the highest bidder wins but pays the second highest price.

The optimal strategy is for each player to bid their actual utility.

Value to the seller is the second-highest utility (which is actually also true, in a certain limit, for first-price auctions).

Two-player, turn-taking, zero-sum, perfect information games

Chess, Go, Backgammon, etc.

- Generally discrete state and action spaces
- Zero-sum: $U(s, p_1) = -U(s, p_2)$
- Added state variable: $\textit{to\_move} \in \{p_1, p_2\}$

Big question: what does it mean for a plan or policy to be optimal in this case?

If you know your opponent's policy, then it becomes a path-search problem (or MDP if the game has stochastic transitions).

Otherwise, we often look for the minimax optimal strategy: each player assumes the other player is perfectly rationally trying to win, and acts rationally under that assumption.

Note that the minimax optimal strategy isn't necessarily the best thing to play against a stupid opponent!

Minimax optimal strategy

- Solving online: Can make a "game tree"
  - Similar to alternating layers of expectimax, but this time we alternate between min and max
  - Max is the player selecting their current move, assuming the other player will take actions to try to minimize their score
  - Will find the optimal strategy if the whole tree can be searched
  - alpha-beta pruning is a cool strategy for reducing search-space size while retaining optimality
- If the game has a stochastic transition function, then you need to add expectation layers, between the min and max layers.
- If the tree is too big to search, then we are generally left with MCTS-type strategies
- Learned heuristic guidance (in the form of policies for biasing which actions to expand during search and estimated Q values) are very helpful! (See alpha-Go-zero, etc!)

Offline minimax strategies

Just as for MDPs, it's theoretically possible to compute an optimal value function, and hence, policy, using dynamic programming! Here's the value function for player max, assuming deterministic transitions, and a terminal 0-1 reward:

$$V_{\max}(s) = \begin{cases} +1 & \text{if } s \text{ is a win for max} \\ -1 & \text{if } s \text{ is a win for min} \\ \max_a V_{\min}(T(s,a)) & \text{otherwise} \end{cases}$$

$$V_{\min}(s) = -V_{\max}(s)$$

- Depending on the game, the state might need to include steps left to go (if finite horizon).
- Just need to compute one value function!

Adding stochasticity

$$V_{\max}(s) = \begin{cases} +1 & \text{if } s \text{ is a win for max} \\ -1 & \text{if } s \text{ is a win for min} \\ \max_a \sum_{s'} P(s' \mid s, a) V_{\min}(s') & \text{otherwise} \end{cases}$$

$$V_{\min}(s) = -V_{\max}(s)$$

- Still just need to compute one value function!
- Can use systematic value iteration to compute this
- In large spaces, do "self-play":
  - Basically RL in simulation, but alternating player perspectives
  - Still need to worry about function approximation strategy and exploration strategy

Normal-form games

One-step, simultaneous actions, no constraint on payoffs. Much harder! What does it mean to act rationally?
Prisoner's dilemma: two players, each can **Cooperate** or **Defect**:

Player A \ Player B	**Cooperate**	**Defect**
Cooperate	(3, 3)	(0, 5)
Defect	(5, 0)	(1, 1)

- Dominant strategy for player A: **Defect**, maximizes utility independent of other player's choice (same for B)
- If we allow binding agreements or repeated play, we get a higher payoff for everyone.

Nash equilibrium

What if there's no dominant strategy? Can look for a pair of strategies (π_1, π_2) such that:

- If player 1 knows that player 2 is going to do π_2 then π_1 maximizes player 1's expected payoff, and
- If player 2 knows that player 1 is going to do π_1 then π_2 maximizes player 2's expected payoff.

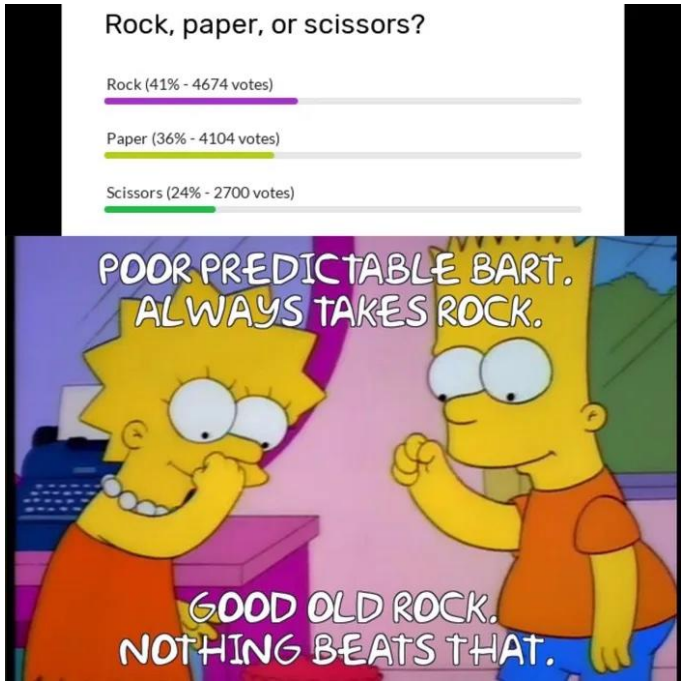
Player A \ Player B	**Left**	**Right**
Left	(0, 0)	(1, 1)
Right	(1, 1)	(0, 0)

- Two Nash equilibria: (L, R) and (R, L)
- Requires coordination to get a good payoff

Randomized policies

In MDPs and games like Backgammon, there is an optimal deterministic policy. But not necessarily true in matrix games. Here the Nash equilibrium is a mixed (randomized) strategy.

Bart \ Lisa	Rock	Paper	Scissors
Rock	(0, 0)	(-1, 1)	(1, -1)
Paper	(1, -1)	(0, 0)	(-1, 1)
Scissors	(-1, 1)	(1, -1)	(0, 0)



Straightforward linear algebra proof that there's a single Nash equilibrium in which each player chooses uniformly at random.

So much more!

- Ways of solving coordination problems
- What if we iterate a matrix game?
- What if two players are interacting in an MDP-like situation? Can seek a Nash equilibrium in the space of simple policies, for example.
- What if we have sequential games with partial information? (e.g. Poker)

Take Gabriele Farina's multi-agent learning class!

Next time: Last lecture

- Connections to machine learning
- Please fill out the subject evaluation