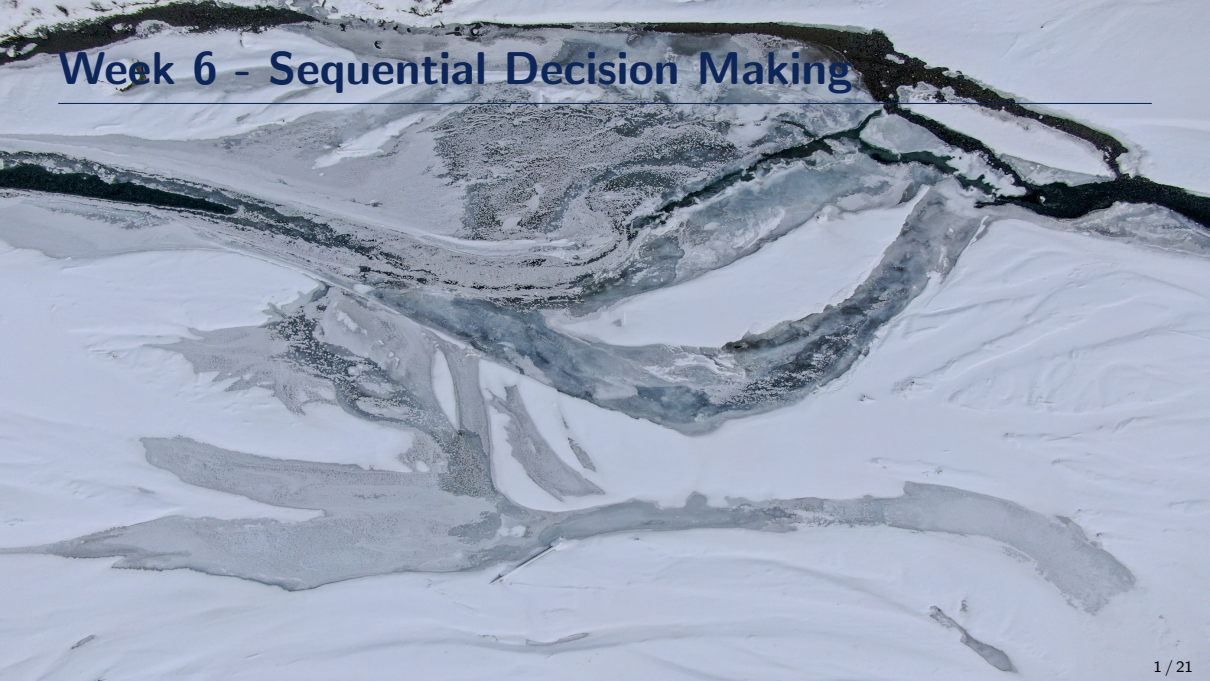
Week 6 - Sequential Decision Making



Extensive Form Games

Sequential moves

- Let's use a tree-like structure, similarly to Chess

Extensive Form Games

Imperfect information

- Player can't see opponent's cards
- Consider these two situations
  $(A\spadesuit 8\diamondsuit) - (K\heartsuit K\diamondsuit)$ and $(A\spadesuit 8\diamondsuit) - (2\heartsuit 7\heartsuit)$
- Even though that these situations are different, player 1 can't distinguish them
- Let's make some game states indistinguishable (from the player's) point of view, so that he must use the same strategy in all the nodes he can't tell apart

Extensive Form Games

Chance

- Let's add another player, the chance player (typically denoted as the player 0 or the player c)
- The chance does plays according to some fixed probability distribution

Extensive-Form Games Formalization

Extensive-form game G is a tuple consisting of:

- a finite set of players $\mathcal{N} = \{1, 2, \ldots, n\}$,
- a finite set of sequences $\mathcal{H}$. Each member $h \in \mathcal{H}$ is called a **history** and consists of a list of actions. The empty sequence $\varnothing$ is in H and every prefix of a history is also a history, i.e. $(h, a) \in \mathcal{H} \implies h \in \mathcal{H}$. $h'a \sqsubseteq h$ denotes that h' is a prefix of h. $\mathcal{Z} \subseteq \mathcal{H}$ is the set of terminal histories, i.e. histories that are not prefixes of any other history,
- a finite set of actions available in every non-terminal history $A(h) = \{a : (h, a) \in \mathcal{H}\}$,
- a function $p\colon \mathcal{H} \setminus \mathcal{Z} \to \mathcal{N} \cup \{c\}$ that assigns each non-terminal history $h \in \mathcal{H} \setminus \mathcal{Z}$ an **acting player** (either $i \in \mathcal{N}$ or the chance player c,
- a function f_c that associates every history h where $p(h) = c$ a probability measure over $A(h)$. Each such probability measure is independent of every other measure,
- a partition $\mathcal{S}_i$ of histories $h \in \mathcal{H}\colon p(h) = i$ where player i is to act. A set $s_i \in \mathcal{S}_i$ is an **information state** of player $i \in \mathcal{N}$,
- a **utility function** $u_i\colon \mathcal{Z} \to \mathbb{R}$ for every player $i \in \mathcal{N}$.

Extensive-Form Game Tree Example

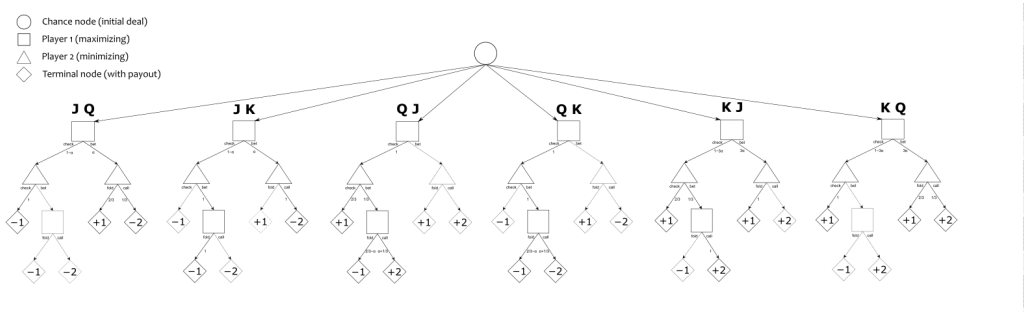


Figure: Game tree of a simple game of Kuhn Poker

Extensive-Form Game Tree Example

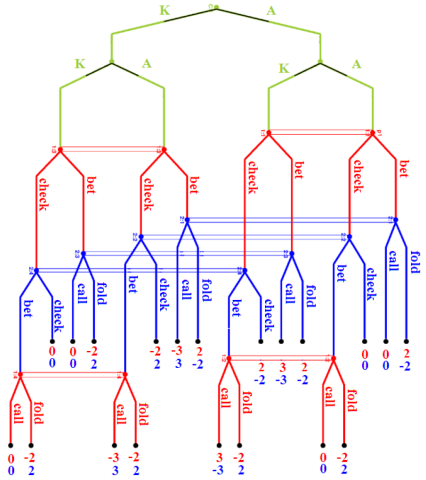


Figure: An example of a game tree a simple Poker-like game.

Extensive-Form Games Formalization - Example

- $\mathcal{N} = \{1, 2\}$,
- $\mathcal{H} = \{(\varnothing), (K), (K, A), (K, A, bet), (K, A, bet, call), \cdots\}$,
- $\mathcal{Z} \subseteq \mathcal{H}$, $\mathcal{Z} = \{(K, A, bet, call), (K, A, bet, fold), \cdots\}$,
- $A(K, A) = \{bet, fold\}$
  $A(K, A, bet) = \{call, fold\}$,
- $p(\varnothing) = c, p(K) = c, p(K, A) = 1, p(K, A, check) = 2$,
- $f_c(\varnothing) = 0.5A, 0.5K$
  $f_c(A) = 0.5A, 0.5K$,
- $\mathcal{S}_1 = \{\{(K, A), (K, K)\}, \{(K, A, check), (K, K, check)\}, \cdots\}$
  $\mathcal{S}_2 = \{\{(A, K), (A, K)\}, \{(A, K, check), (K, K, check)\}, \cdots\}$,
- $u_1(K, A, bet, call) = -3$
  $u_2(K, A, bet, fold) = -2$.

Strategies

- Now the player does not choose a row/column, instead an edge in the game tree
- Since we need that the player can't distinguish the states merged into information sets, we allow the player to choose an action in information sets in contrast to histories/nodes. This way, the player must play the same strategy in all histories grouped in that information set.
- **A behavioral policy** is a mapping from a (info)state to a distribution over the available actions $s \to \Delta(\mathcal{A}(s))$. We denote a policy of a player i as π_i and the policy in a state $s \in \mathcal{S}$ as $\pi_i(s)$. The set of all policies of player i is then Π_i. As a single player acts in a state, we often use simply $\pi(s)$.
- Furthermore, the probability mass for an action $a \in \mathcal{A}(s)$ is $\pi_i(s, a)$. A policy profile consists of strategies of both player $\pi = (\pi_1, \pi_2)$.

Reach Probabilities

Given a strategy profile π, we define:

- the **reach probability of a history** $h \in \mathcal{H}$ as

$$P^{\pi}(h) = \prod_{h'a \sqsubseteq h} \pi(h', a)$$

- the **reach probability of an information state** $s \in \mathcal{S}$ as

$$P^{\pi}(s) = \sum_{h \in \mathcal{H}(s)} P^{\pi}(h)$$

Note that, for brevity, we overload the notation $P^{\pi}(\cdot)$ for both the histories and the information states. The same applies to $\pi(\cdot, a)$.

Strategy Evaluation

Expected utility $u_i(\pi)$ of player i given a strategy profile π is a weighted sum over the terminal histories $h \in \mathcal{Z}$ defined as

$$u_i(\pi) = \sum_{h \in \mathcal{Z}} P^{\pi}(h) u_i(h)$$

Derived Concepts

- The history value $v_i^\pi(h)$ is the expected future reward under policy π to player i given the history h.
- The history-action value $q_i^\pi(h, a_i)$ is defined analogously, except that the player i first takes the action a_i.
- To compute state value $v_i^\pi(s)$ and state-action value $q_i^\pi(s, a_i)$, we need to correctly weight the corresponding histories $h \in \mathcal{H}$ [1]

$$v_i^\pi(s) = \sum_{h \in \mathcal{H}(s)} P^\pi(h) v_i^\pi(h)$$
$$q_i^\pi(s, a) = \sum_{h \in \mathcal{H}(s)} P^\pi(h) q_i^\pi(h, a)$$

[1] You might notice that we do not normalize these values. While you are correct, it is actually easier to work with the non-normalized values in most cases.

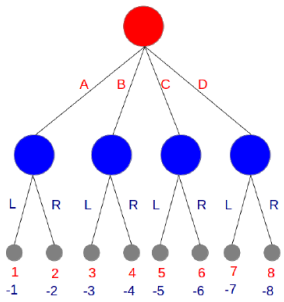
Perfect Recall

- Extensive form games is a powerful model, and it allows us to capture some non-realistic properties
- Real life players do not forget information that they already knew
- This property is called **perfect recall**. We say that an extensive form games satisfies perfect recall, if all players can recall their previous actions and the corresponding information sets
- This is a critical property as without this restriction, even basic operations (e.g. best response calculation) become hard

Extensive Form Games To Normal Form Games

- If the game satisfies perfect recall, we can convert an extensive form game to an equivalent normal form game
- A pure strategy in normal form game corresponds to all combinations of pure strategies in information sets of that player

Extensive Form Games To Normal Form Games



	A	B	C	D
(A-L, B-L, C-L, D-L)	1;-1	3;-3	5;-5	7;-7
(A-L, B-L, C-L, D-R)	1;-1	3;-3	5;-5	8;-8
(A-L, B-L, C-R, D-L)	1;-1	3;-3	6;-6	7;-7
(A-L, B-L, C-R, D-R)	1;-1	3;-3	6;-6	8;-8
(A-L, B-R, C-L, D-L)	1;-1	4;-4	5;-5	7;-7
(A-L, B-R, C-L, D-R)	1;-1	4;-4	5;-5	8;-8
(A-L, B-R, C-R, D-L)	1;-1	4;-4	6;-6	7;-7
(A-L, B-R, C-R, D-R)	1;-1	4;-4	6;-6	8;-8
(A-R, B-L, C-L, D-L)	2;-2	3;-3	5;-5	7;-7
(A-R, B-L, C-L, D-R)	2;-2	3;-3	5;-5	8;-8
(A-R, B-L, C-R, D-L)	2;-2	3;-3	6;-6	7;-7
(A-R, B-L, C-R, D-R)	2;-2	3;-3	6;-6	8;-8
(A-R, B-R, C-L, D-L)	2;-2	4;-4	5;-5	7;-7
(A-R, B-R, C-L, D-R)	2;-2	4;-4	5;-5	8;-8
(A-R, B-R, C-R, D-L)	2;-2	4;-4	6;-6	7;-7
(A-R, B-R, C-R, D-R)	2;-2	4;-4	6;-6	8;-8

Extensive Form Games To Normal Form Games

Lemma: Extensive Form Games to Normal Form Games

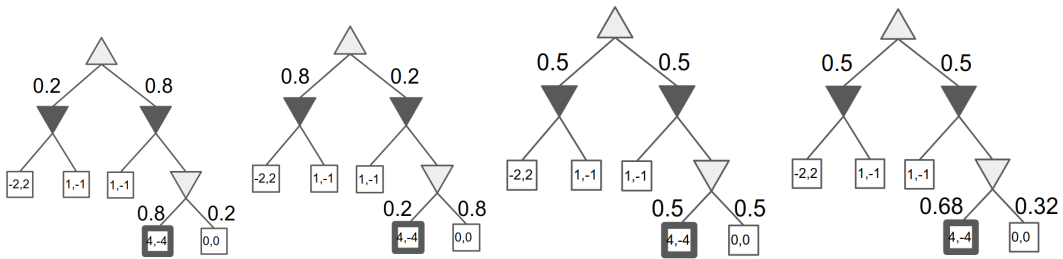
Given any two-player extensive form game with perfect recall, it's possible to create an equivalent normal form game

- Therefore, all properties that we showed for the normal-from games, do also hold for the extensive games.
- Existence of the equilibrium.
- There is always some pure best response.
- Nice properties of equilibrium for two players zero sum games.
- Not-so-nice properties of other games ...
- It is also easy to represent any normal form game as an extensive game.

Policy Averaging

- One must exercise caution during any averaging in the space of behavioral strategies.
- A common mistake is to simply average the strategy locally, in each individual state.
- This is incorrect as for $\pi_1^c = 0.5\pi_1^a + 0.5\pi_1^b$, we expect $R_1(\pi_1^c, \pi_2) = 0.5R_1(\pi_1^a, \pi_2) + 0.5R_1(\pi_1^b, \pi_2)$.
- Figure 3 shows a counterexample to this expectation for naive per-state averaging of the policy.
- The issues is because the dynamics are inherently sequential — the distribution over the states $P^\pi(z)$ depends on the full sequence.Proper averaging in behavioral state must thus take into account the reach probability of the state (Figure 3d).

Policy Averaging Example



(a) π_1^a (b) π_1^b (c) π_1^c - naive average (d) π_1^c - proper average

Figure: Consider the highlighted state s against an opponent who plays to reach that state, and its reach probability $P^{\pi}(s)$. a) $P^{\pi_1^a}(s) = 0.64$ b) $P^{\pi_1^b}(s) = 0.04$ c) Naive per-state averaging of the strategy simply averages the state strategy in isolation: $P^{\pi_1^c}(s) = 0.25 \neq 0.5 \cdot 0.64 + 0.5 \cdot 0.04$ d) Proper averaging of the strategy takes reach account into consideration:

Counterfactual Reach Probability

Given a strategy profile π, we define:

- the **counterfactual reach probability of a history** $h \in \mathcal{H}$ as

$$P^{\pi}_{-i}(h) = \prod_{h'a \sqsubseteq h : p(h') \neq i} \pi(h', a).$$

The counterfactual reach probability $P^{\pi}_{-i}(h)$ is the probability of reaching history h when player i *attempts* to reach that particular history and all other players stick to their strategies π_j. In other words, at every history h' that is a prefix of history h, where player i has to act, it places all of its probability mass on the particular action a that is on the path to h, i.e. $\pi_i(h', a) = 1$, where $h'a \sqsubseteq h$ and $p(h') = i$.

Best Response

- Just like for perfect information games, we traverse the tree bottom-up and greedily select the best action-value, propagating the values up the tree.
- In imperfect information games, we need to weight the action values for all the histories in an (info)state to compute the state's action value. But to compute the weights, we also need a strategy up the tree.
- Furthermore, to compute $P^{\pi}(h)$ we need strategy for both players. How can we know our (best response) strategy up the tree if we are computing it bottom up?
- Perfect recall guarantees that our contribution to $P^{\pi}(h)$ for all $h \in \mathcal{H}(s)$ is the same! $P_i^{\pi}(h) = P_i^{\pi}(h') \forall h \in \mathcal{H}(s)$
- Thus, in best response calculation, we can weight the history-action values only using opponent's policy $\sum_{h \in \mathcal{H}(s)} P_{-i}^{\pi}(h) q_i^{\pi}(h, a)$

Week 6 Homework

This week, your task is to implement:

- a suitable data structure to represent extensive-form games
- two-player version of Kuhn Poker using the data structure
- a function that computes the utility of both players for a given strategy profile
- a function that properly averages a pair of strategies
- a function that computes a best response to a given strategy of the opponent
- the Fictitious Play algorithm by appropriately combining the functions above

Finally, plot the exploitability of the sequence of average strategy profiles produced by the algorithm in Kuhn Poker