Algorithmic Game Theory and Applications

Lecture 10: Games in Extensive Form

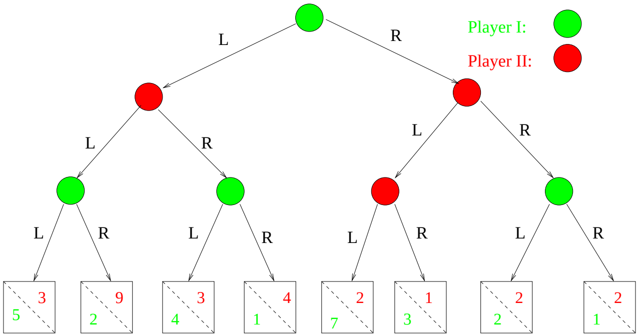
Kousha Etessami

the setting and motivation

▷ Most games in “real life” are not in “strategic form”: players don’t pick their entire strategies independently (“simultaneously”). Instead, the game transpires over time, with players making “moves” to which other players react with “moves”, etc. Examples: chess, poker, bargaining, dating, ...

the setting and motivation

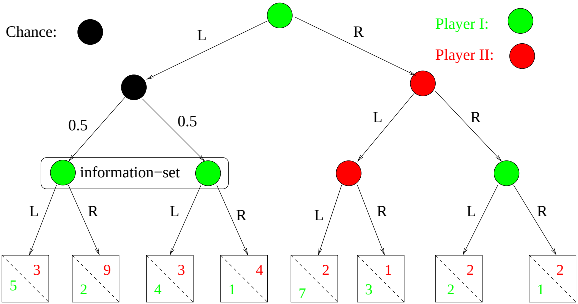
- Most games in "real life" are not in "strategic form": players don't pick their entire strategies independently ("simultaneously"). Instead, the game transpires over time, with players making "moves" to which other players react with "moves", etc. Examples: chess, poker, bargaining, dating, ...
- A "game tree" looks something like this:



- But we may also need some other "features".

chance, information, etc.

Some tree nodes may be *chance* (probabilistic) nodes, controlled by no player (by "nature"). (Poker, Backgammon.) Also, a player may not be able to distinguish between several of its "positions" or "nodes", because not all *information* is available to it. (Think Poker, with opponent's cards hidden.) Whatever move a player employs at a node must be employed at all nodes in the same "information set".



To define extensive form games, we have to formalize all these.

Trees: a formal definition

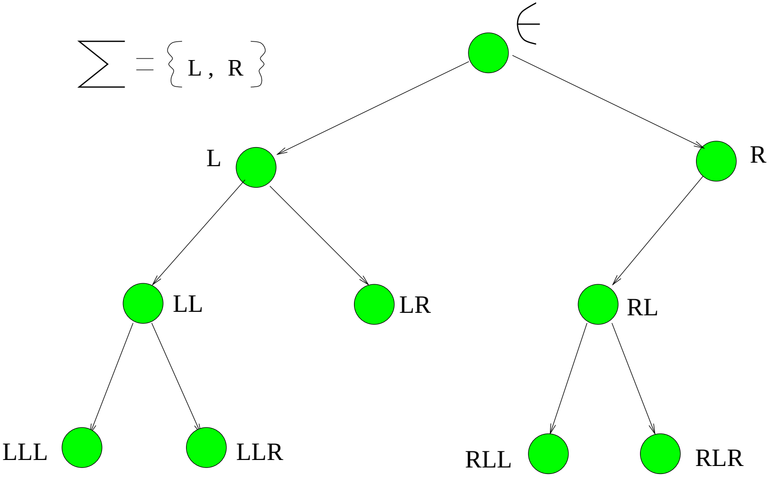
▷ Let $\Sigma = \{a_1, a_2, \ldots, a_k\}$ be an alphabet. A tree over Σ is a set $T \subseteq \Sigma^*$, of nodes $w \in \Sigma^*$ such that: if $w = w'a \in T$, then $w' \in T$. (I.e., it is a prefix-closed subset of Σ^*.)

▷ For a node $w \in T$, the children of w are $ch(w) = \{w' \in T \mid w' = wa\ ,\ \text{for some } a \in \Sigma\}$. For $w \in T$, let $Act(w) = \{a \in \Sigma \mid wa \in T\}$ be "actions" available at w.

▷ A leaf (or terminal) node $w \in T$ is one where $ch(w) = \emptyset$. Let $L_T = \{w \in T \mid w \text{ a leaf}\}$.

▷ A (finite or infinite) path π in T is a sequence $\pi = w_0, w_1, w_2, \ldots$ of nodes $w_i \in T$, where if $w_{i+1} \in T$ then $w_{i+1} = w_i a$, for some $a \in \Sigma$. It is a complete path (or play) if $w_0 = \epsilon$ and every non-leaf node in π has a child in π. Let Ψ_T denote the set of plays of T.

Tree example



games in extensive form

A **Game in Extensive Form**, $\mathcal{G}$, consists of:

1. A set $N = \{1, \ldots, n\}$ of players.

games in extensive form

A **Game in Extensive Form**, $\mathcal{G}$, consists of:

1. A set $N = \{1, \ldots, n\}$ of players.
2. A tree T, called the game tree, over some Σ.

games in extensive form

A **Game in Extensive Form**, $\mathcal{G}$, consists of:

1. A set $N = \{1, \ldots, n\}$ of players.
2. A tree T, called the game tree, over some Σ.
3. A partition of the tree nodes T into disjoint sets $Pl_0, Pl_1, \ldots, Pl_n$, such that $T = \bigcup_{i=0}^{n} Pl_i$. Where Pl_i, $i \geq 1$, denotes the nodes of player i, where it is player i's turn to move. (And Pl_0 denotes the "chance"/"nature" nodes.)

games in extensive form

A **Game in Extensive Form**, $\mathcal{G}$, consists of:

1. A set $N = \{1, \ldots, n\}$ of players.
2. A tree T, called the game tree, over some Σ.
3. A partition of the tree nodes T into disjoint sets $Pl_0, Pl_1, \ldots, Pl_n$, such that $T = \bigcup_{i=0}^{n} Pl_i$. Where Pl_i, $i \geq 1$, denotes the nodes of player i, where it is player i's turn to move. (And Pl_0 denotes the "chance"/"nature" nodes.)
4. For each "nature" node, $w \in Pl_0$, a probability distribution $q_w : Act(w) \mapsto \mathbb{R}$ over its actions: $q_w(a) \geq 0$, & $\sum_{a \in Act(w)} q_w(a) = 1$.

games in extensive form

A **Game in Extensive Form**, $\mathcal{G}$, consists of:

1. A set $N = \{1, \ldots, n\}$ of players.
2. A tree T, called the game tree, over some Σ.
3. A partition of the tree nodes T into disjoint sets $Pl_0, Pl_1, \ldots, Pl_n$, such that $T = \bigcup_{i=0}^{n} Pl_i$. Where Pl_i, $i \geq 1$, denotes the nodes of player i, where it is player i's turn to move. (And Pl_0 denotes the "chance"/"nature" nodes.)
4. For each "nature" node, $w \in Pl_0$, a probability distribution $q_w : Act(w) \mapsto \mathbb{R}$ over its actions: $q_w(a) \geq 0$, & $\sum_{a \in Act(w)} q_w(a) = 1$.
5. For each player $i \geq 1$, a partition of Pl_i into disjoint non-empty information sets $Info_{i,1}, \ldots, Info_{i,r_i}$, such that $Pl_i = \bigcup_{j=0}^{r_i} Info_{i,j}$. Moreover, for any i, j, & all nodes $w, w' \in Info_{i,j}$, $Act(w) = Act(w')$. (I.e., the set of possible "actions" from all nodes in one information set is the same.)

games in extensive form

A **Game in Extensive Form**, $\mathcal{G}$, consists of:

1. A set $N = \{1, \ldots, n\}$ of players.
2. A tree T, called the game tree, over some Σ.
3. A partition of the tree nodes T into disjoint sets $Pl_0, Pl_1, \ldots, Pl_n$, such that $T = \bigcup_{i=0}^{n} Pl_i$. Where Pl_i, $i \geq 1$, denotes the nodes of player i, where it is player i's turn to move. (And Pl_0 denotes the "chance"/"nature" nodes.)
4. For each "nature" node, $w \in Pl_0$, a probability distribution $q_w : Act(w) \mapsto \mathbb{R}$ over its actions: $q_w(a) \geq 0$, & $\sum_{a \in Act(w)} q_w(a) = 1$.
5. For each player $i \geq 1$, a partition of Pl_i into disjoint non-empty information sets $Info_{i,1}, \ldots, Info_{i,r_i}$, such that $Pl_i = \bigcup_{j=0}^{r_i} Info_{i,j}$. Moreover, for any i, j, & all nodes $w, w' \in Info_{i,j}$, $Act(w) = Act(w')$. (I.e., the set of possible "actions" from all nodes in one information set is the same.)
6. For each player i, a function $u_i : \Psi_T \mapsto \mathbb{R}$, from (complete) plays to the payoff for player i.

explanation and comments

▷ Question: Why associate payoffs to "plays" rather than to leaves at the "end" of play?

explanation and comments

▷ Question: Why associate payoffs to "plays" rather than to leaves at the "end" of play?
Answer: We in general allow infinite trees. We will later consider "infinite horizon" games in which play can go on for ever. Payoffs are then determined by the entire history of play. For "finite horizon" games, where tree T is finite, it suffices to associate payoffs to the leaves, i.e., $u_i : L_T \mapsto \mathbb{R}$.

explanation and comments

▷ Question: Why associate payoffs to "plays" rather than to leaves at the "end" of play?
Answer: We in general allow infinite trees. We will later consider "infinite horizon" games in which play can go on for ever. Payoffs are then determined by the entire history of play. For "finite horizon" games, where tree T is finite, it suffices to associate payoffs to the leaves, i.e., $u_i : L_T \mapsto \mathbb{R}$.

▷ We defined our alphabet of possible actions Σ to be finite, which is generally sufficient for our purposes. In other words, the tree is finitely branching. In more general settings, even the set of possible actions at a given node can be infinite.

explanation and comments

▷ Question: Why associate payoffs to "plays" rather than to leaves at the "end" of play?
Answer: We in general allow infinite trees. We will later consider "infinite horizon" games in which play can go on for ever. Payoffs are then determined by the entire history of play. For "finite horizon" games, where tree T is finite, it suffices to associate payoffs to the leaves, i.e., $u_i : L_T \mapsto \mathbb{R}$.

▷ We defined our alphabet of possible actions Σ to be finite, which is generally sufficient for our purposes. In other words, the tree is finitely branching. In more general settings, even the set of possible actions at a given node can be infinite.

▷ Later, we will focus on the following class of games:
Definition An extensive form game $\mathcal{G}$ is called a game of **perfect information**, if every information set $Info_{i,j}$ contains only 1 node.

pure strategies

$\triangleright$ A pure strategy s_i for player i in an extensive game $\mathcal{G}$ is a function $s_i : Pl_i \mapsto \Sigma$ that assigns actions to each of player i's nodes, such that $s_i(w) \in Act(w)$, & such that if $w, w' \in Info_{i,j}$, then $s_i(w) = s_i(w')$.
Let S_i be the set of pure strategies for player i.
$\triangleright$ Given pure profile $s = (s_1, \ldots, s_n) \in S_1 \times \ldots \times S_n$, if there are no chance nodes (i.e., $Pl_0 = \emptyset$) then s uniquely determines a play π_s of the game: players move according their strategies:

- Initialize $j := 0$, and $w_0 := \epsilon$;
- While (w_j is not at a terminal node)
  If $w_j \in Pl_i$, then $w_{j+1} := w_j\ s_i(w_j)$;
  $j := j + 1$;
- $\pi_s = w_0, w_1, \ldots$

$\triangleright$ What if there are chance nodes?

pure strategies and chance

If there are chance nodes, then $s \in S$ determines a probability distribution over plays π of the game.

For finite extensive games, where T is finite, we can calculate the probability $p_s(\pi)$ of play π, using probabilities $q_w(a)$: Suppose $\pi = w_0, \ldots, w_m$, is a play of T. Suppose further that for each $j < m$, if $w_j \in Pl_i$, then $w_{j+1} = w_j \; s_i(w_j)$. Otherwise, let $p_s(\pi) = 0$.

Let $w_{j_1}, \ldots, w_{j_r}$ be the chance nodes in π, and suppose, for each $k = 1, \ldots, r$, $w_{j_k+1} = w_{j_k} a_{j_k}$, i.e., the required action to get from node w_{j_k} to node w_{j_k+1} is a_{j_k}. Then

$$p_s(\pi) := \prod_{k=1}^{r} q_{w_{j_k}}(a_{j_k})$$

For infinite extensive games, defining these distributions in general requires much more elaborate definitions (proper "measure theoretic" probability). We will avoid the heavy stuff.

chance and expected payoffs

For a finite extensive game, given pure profile $s = (s_1, \ldots, s_n) \in S_1 \times \ldots \times S_n$, we can, define the "expected payoff" for player i under s as:

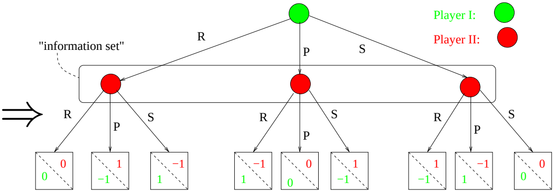
$$h_i(s) := \sum_{\pi \in \Psi_t} p_s(\pi) * u_i(\pi)$$

Again, for infinite games, much more elaborate definitions of "expected payoffs" would be required.
Note: This "expected payoff" does not arise because any player is mixing its strategies. It arises because the game itself contains randomness.
We can also combine both: players may also randomize amongst their strategies, and we could then define the overall expected payoff.

from strategic to extensive games

Every finite strategic game Γ can be encoded easily and concisely as an extensive game $\mathcal{G}_\Gamma$. We illustrate this via the Rock-Paper-Scissor 2-player game (the n-player case is an easy generalization):

		Player II	
	Rock	Paper	Scissors
Player I Rock	0, 0	−1, 1	1, −1
Player I Paper	1, −1	0, 0	−1, 1
Player I Scissors	−1, 1	1, −1	0, 0



from extensive to strategic games

Every extensive game $\mathcal{G}$ can be viewed as a strategic game $\Gamma_{\mathcal{G}}$:
▷ In $\Gamma_{\mathcal{G}}$, the strategies for player i are the mappings $s_i \in S_i$.
▷ In $\Gamma_{\mathcal{G}}$, we define payoff $u_i(s) := h_i(s)$, for all pure profiles s.
If the extensive game $\mathcal{G}$ is finite, i.e., tree T is finite, then the strategic game $\Gamma_{\mathcal{G}}$ is also finite.
Thus, all the theory we developed for finite strategic games also applies to finite extensive games.
Unfortunately, the strategic game $\Gamma_{\mathcal{G}}$ is generally exponentially bigger than $\mathcal{G}$. Note that the number of pure strategies for a player i with $|PI_i| = m$ nodes in the tree, is in the worst case $|\Sigma|^m$.
So it is often unwise to naively translate a game from extensive to strategic form in order to "solve" it.
If we can find a way to avoid this blow-up, we should.

imperfect information & "perfect recall"

▷ An extensive form game (EFG) is a game of **imperfect information** if it has non-trivial (size > 1) information sets. Players don't have full knowledge of the current "state" (current node of the game tree).

▷ Informally, an imperfect information EFG has **perfect recall** if each player i never "forgets" its own sequence of prior actions and information sets. I.e., a EFG has perfect recall if whenever $w, w' \in Info_{i,j}$ belong to the same information set, then the "*visible history*" for player i (sequence of information sets and actions of player i during the play) prior to hitting node w and w' must be exactly the same.

▷ [Kuhn'53]: with perfect recall it suffices to restrict players' strategies to **behavior strategies**: strategies that only randomize (independently) on actions at each information set.

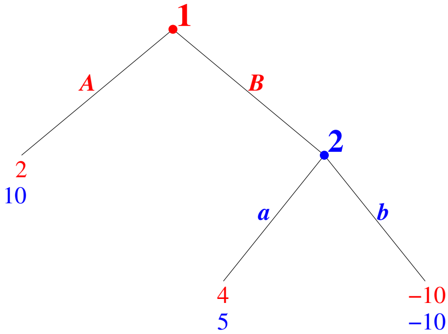
▷ Perfect recall is often assumed as a "sanity condition" for EFGs (most games we encounter do have perfect recall).

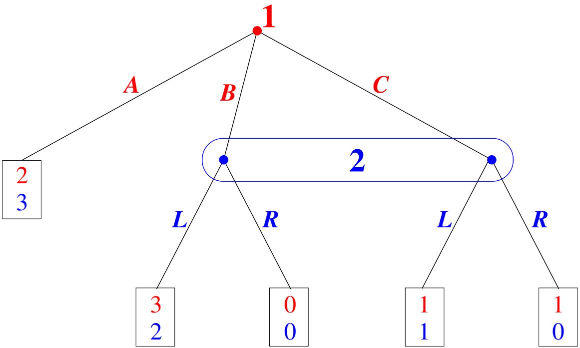
subgames and (subgame) perfection

▷ A **subgame** of an extensive form game is an induced subtree of the game tree, induced by a node (acting as its "root") and *all* of that node's descendants, such that moreover the subtree has only "*self-contained information sets*". In other words, every node in that subtree must be contained in an information set that is itself entirely contained within that subtree.

▷ For an extensive form game G, a profile of behavior strategies $b = (b_1, \ldots, b_n)$ for the players is called a **subgame perfect Nash equilibrium** (SPNE) if it defines a Nash equilibrium for *every* subgame of G.

▷ [Selten'75]: Nash equilibrium (NE) (and even SPNE) is inadequately refined as a solution concept for extensive form games. In particular, such equilibria can involve **"Non-credible threats"**:


1
A
B
2
2
10
a
b
4
5
−10
−10



Addressing this general inadequecy of NE and SPNE requires a more refined notion of equilibrium called **trembling-hand perfect equilibrium** [Selten'73].

solving games of imperfect info.

For EFGs *with perfect recall* there are ways to avoid the exponential blow-up of converting to normal form. We only briefly mention algorithms for imp-inf games.
(See, e.g., [Koller-Megiddo-von Stengel'94].)

▷ In strategic form 2-player zero-sum games we can find minimax solutions efficiently (P-time) via LP. For 2-player zero-sum extensive imp-info games (without perfect recall), finding a minimax solution is **NP-hard**. NE's of 2-player EFGs can be found in exponential time.

▷ The situation is better with perfect recall: 2-player zero-sum imp-info games of perfect recall can be solved in P-time, via LP, and 2-player NE's for arbitrary perfect recall games can be found in exponential time using a Lemke-type algorithm.

▷ [Etessami'2014,2021]: For EFGs with ≥ 3 players with perfect recall, computing refinements of Nash equilibrium (including computing a "trembling-hand perfect" equilibrium and a "quasi-perfect" equilibrium) can be reduced to computing a NE for a 3-player normal form game.

For more on this see:

K. Etessami, "The complexity of computing a (quasi-)perfect equilibrium for an n-player extensive form game."
Games and Economic Behavior, vol. 125, pp. 107–140, 2021.

Our main focus next will be games of perfect information. There the situation is much easier.

games of perfect information

A game of perfect information has only 1 node per information set. So, for these we can forget about information sets.

Examples: Chess, Backgammon, ...

counter-Examples: Poker, Bridge, ...

Theorem([Kuhn'53]) *Every finite extensive game of perfect information, $\mathcal{G}$, has a NE (in fact a SPNE) in pure strategies.*

In other words, there is a pure profile $(s_1, \ldots, s_n) \in S$ that is a Nash Equilibrium (and a subgame perfect equilibrium).

Our proof provides an efficient algorithm to compute such a pure profile, given $\mathcal{G}$, using "backward induction".

A special case of this theorem says the following:

Proposition([Zermelo'1912]) *In Chess, either:*

1. *White has a "winning strategy", or*
2. *Black has a "winning strategy", or*
3. *Both players have strategies to force a draw.*

Next time, perfect information games.