Extract from the slides of the course

CPSC 532L - Foundations of Multiagent Systems by Kevin Leyton-Brown, University of British Columbia.

https://www.cs.ubc.ca/~kevinlb/teaching/cs532l%20-%202011-12/

Other strategic game types

Extensive Form Games

Lecture 7

Lecture Overview

1. Perfect-Information Extensive-Form Games
2. Subgame Perfection
3. Backward Induction

Introduction

- The normal form game representation does not incorporate any notion of sequence, or time, of the actions of the players
- The extensive form is an alternative representation that makes the temporal structure explicit.
- Two variants:
  - perfect information extensive-form games
  - imperfect-information extensive-form games

Definition

A (finite) perfect-information game (in extensive form) is defined by the tuple $(N, A, H, Z, \chi, \rho, \sigma, u)$, where:

- Players: N is a set of n players

Definition

A (finite) perfect-information game (in extensive form) is defined by the tuple $(N, A, H, Z, \chi, \rho, \sigma, u)$, where:

- Players: N
- Actions: A is a (single) set of actions

Definition

A (finite) perfect-information game (in extensive form) is defined by the tuple $(N, A, H, Z, \chi, \rho, \sigma, u)$, where:

- Players: N
- Actions: A
- Choice nodes and labels for these nodes:
  - Choice nodes: H is a set of non-terminal choice nodes

Definition

A (finite) perfect-information game (in extensive form) is defined by the tuple $(N, A, H, Z, \chi, \rho, \sigma, u)$, where:

- Players: N
- Actions: A
- Choice nodes and labels for these nodes:
  - Choice nodes: H
  - Action function: $\chi : H \to 2^A$ assigns to each choice node a set of possible actions

Definition

A (finite) perfect-information game (in extensive form) is defined by the tuple $(N, A, H, Z, \chi, \rho, \sigma, u)$, where:

- Players: N
- Actions: A
- Choice nodes and labels for these nodes:
  - Choice nodes: H
  - Action function: $\chi : H \rightarrow 2^A$
  - Player function: $\rho : H \rightarrow N$ assigns to each non-terminal node h a player $i \in N$ who chooses an action at h

Definition

A (finite) perfect-information game (in extensive form) is defined by the tuple $(N, A, H, Z, \chi, \rho, \sigma, u)$, where:

- Players: N
- Actions: A
- Choice nodes and labels for these nodes:
  - Choice nodes: H
  - Action function: $\chi : H \to 2^A$
  - Player function: $\rho : H \to N$
- Terminal nodes: Z is a set of terminal nodes, disjoint from H

Definition

A (finite) perfect-information game (in extensive form) is defined by the tuple $(N, A, H, Z, \chi, \rho, \sigma, u)$, where:

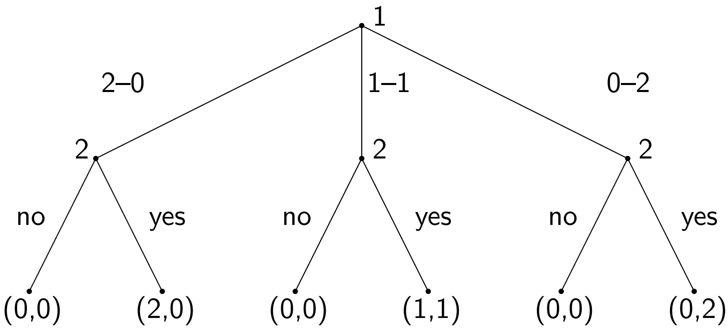
- Players: N
- Actions: A
- Choice nodes and labels for these nodes:
  - Choice nodes: H
  - Action function: $\chi : H \to 2^A$
  - Player function: $\rho : H \to N$
- Terminal nodes: Z
- Successor function: $\sigma : H \times A \to H \cup Z$ maps a choice node and an action to a new choice node or terminal node such that for all $h_1, h_2 \in H$ and $a_1, a_2 \in A$, if $\sigma(h_1, a_1) = \sigma(h_2, a_2)$ then $h_1 = h_2$ and $a_1 = a_2$
  - The choice nodes form a tree, so we can identify a node with its history.

Definition

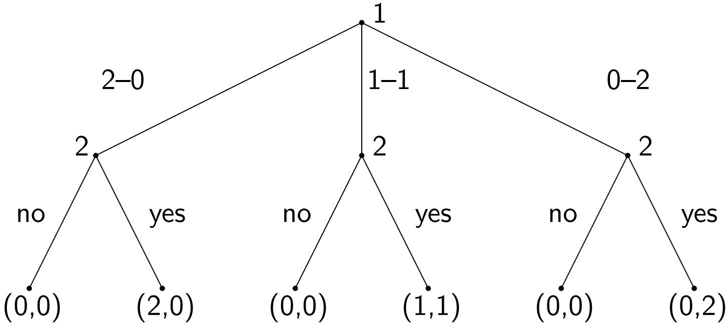
A (finite) perfect-information game (in extensive form) is defined by the tuple $(N, A, H, Z, \chi, \rho, \sigma, u)$, where:

- Players: N
- Actions: A
- Choice nodes and labels for these nodes:
  - Choice nodes: H
  - Action function: $\chi : H \to 2^A$
  - Player function: $\rho : H \to N$
- Terminal nodes: Z
- Successor function: $\sigma : H \times A \to H \cup Z$
- Utility function: $u = (u_1, \ldots, u_n)$; $u_i : Z \to \mathbb{R}$ is a utility function for player i on the terminal nodes Z

Example: the sharing game



Example: the sharing game



Play as a fun game, dividing 100 dollar coins. (Play each partner only once.)

Pure Strategies

- In the sharing game (splitting 2 coins) how many pure strategies does each player have?

Pure Strategies

- In the sharing game (splitting 2 coins) how many pure strategies does each player have?
  - player 1: 3; player 2: 8

Pure Strategies

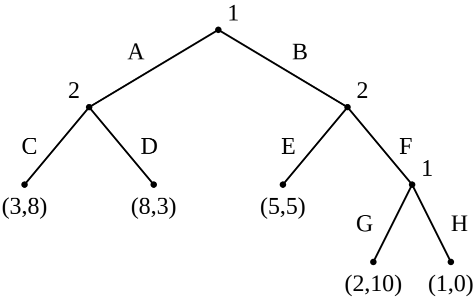
- In the sharing game (splitting 2 coins) how many pure strategies does each player have?
  - player 1: 3; player 2: 8
- Overall, a pure strategy for a player in a perfect-information game is a complete specification of which deterministic action to take at every node belonging to that player.

Definition (pure strategies)

Let $G = (N, A, H, Z, \chi, \rho, \sigma, u)$ be a perfect-information extensive-form game. Then the pure strategies of player i consist of the cross product

$$\underset{h \in H, \rho(h)=i}{\times} \chi(h)$$

Pure Strategies Example



In order to define a complete strategy for this game, each of the players must choose an action at each of his two choice nodes. Thus we can enumerate the pure strategies of the players as follows.

$$S_1 = \{(A, G), (A, H), (B, G), (B, H)\}$$

$$S_2 = \{(C, E), (C, F), (D, E), (D, F)\}$$

Nash Equilibria

Given our new definition of pure strategy, we are able to reuse our old definitions of:

- mixed strategies
- best response
- Nash equilibrium

Theorem

Every perfect information game in extensive form has a PSNE

This is easy to see, since the players move sequentially.

Lecture Overview

1. Perfect-Information Extensive-Form Games
2. Subgame Perfection
3. Backward Induction

Formal Definition

Definition (subgame of G rooted at h)

The subgame of G rooted at h is the restriction of G to the descendents of H.

Definition (subgames of G)

The set of subgames of G is defined by the subgames of G rooted at each of the nodes in G.

- s is a subgame perfect equilibrium of G iff for any subgame G' of G, the restriction of s to G' is a Nash equilibrium of G'
- Notes:
  - since G is its own subgame, every SPE is a NE.
  - this definition rules out "non-credible threats"

Lecture Overview

1. Perfect-Information Extensive-Form Games
2. Subgame Perfection
3. Backward Induction

Computing Subgame Perfect Equilibria

Idea: Identify the equilibria in the bottom-most trees, and adopt these as one moves up the tree

function BACKWARDINDUCTION (node h) **returns** $u(h)$
if $h \in Z$ **then**
  return $u(h)$
$best\_util \leftarrow -\infty$
forall $a \in \chi(h)$ **do**
  $util\_at\_child \leftarrow$ BACKWARDINDUCTION$(\sigma(h, a))$
  if $util\_at\_child_{\rho(h)} > best\_util_{\rho(h)}$ **then**
    $best\_util \leftarrow util\_at\_child$
return $best\_util$

- $util\_at\_child$ is a vector denoting the utility for each player
- the procedure doesn't return an equilibrium strategy, but rather labels each node with a vector of real numbers.
  - This labeling can be seen as an extension of the game's utility function to the non-terminal nodes
  - The equilibrium strategies: take the best action at each node.

Computing Subgame Perfect Equilibria

Idea: Identify the equilibria in the bottom-most trees, and adopt these as one moves up the tree

function $\textsc{BackwardInduction}$ (node h) **returns** $u(h)$
if $h \in Z$ **then**
  return $u(h)$
$best\_util \leftarrow -\infty$
forall $a \in \chi(h)$ **do**
  $util\_at\_child \leftarrow \textsc{BackwardInduction}(\sigma(h, a))$
  if $util\_at\_child_{\rho(h)} > best\_util_{\rho(h)}$ **then**
    $best\_util \leftarrow util\_at\_child$
return $best\_util$

- For zero-sum games, $\textsc{BackwardInduction}$ has another name: the minimax algorithm.
  - Here it's enough to store one number per node.
  - It's possible to speed things up by pruning nodes that will never be reached in play: "alpha-beta pruning".

Lecture Overview

Intro

- Up to this point, in our discussion of extensive-form games we have allowed players to specify the action that they would take at every choice node of the game.
- This implies that players know the node they are in and all the prior choices, including those of other agents.
- We may want to model agents needing to act with partial or no knowledge of the actions taken by others, or even themselves.
- This is possible using imperfect information extensive-form games.
  - each player's choice nodes are partitioned into information sets
  - if two choice nodes are in the same information set then the agent cannot distinguish between them.

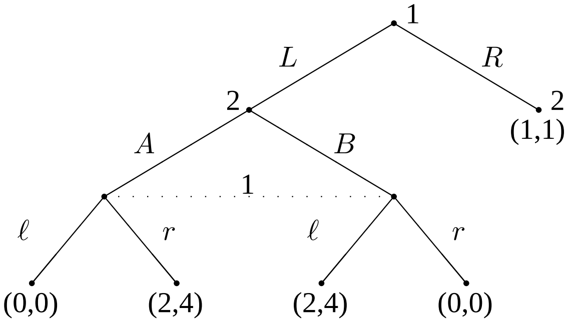
Formal definition

Definition

An imperfect-information game (in extensive form) is a tuple $(N, A, H, Z, \chi, \rho, \sigma, u, I)$, where

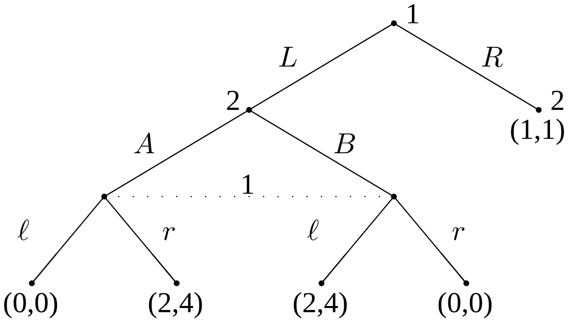
- $(N, A, H, Z, \chi, \rho, \sigma, u)$ is a perfect-information extensive-form game, and
- $I = (I_1, \ldots, I_n)$, where $I_i = (I_{i,1}, \ldots, I_{i,k_i})$ is an equivalence relation on (that is, a partition of) $\{h \in H : \rho(h) = i\}$ with the property that $\chi(h) = \chi(h')$ and $\rho(h) = \rho(h')$ whenever there exists a j for which $h \in I_{i,j}$ and $h' \in I_{i,j}$.

Example



- What are the equivalence classes for each player?
- What are the pure strategies for each player?

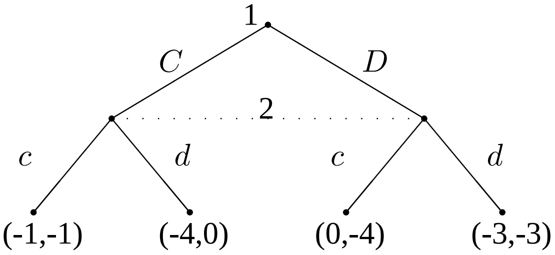
Example



- What are the equivalence classes for each player?
- What are the pure strategies for each player?
  - choice of an action in each equivalence class.
- Formally, the pure strategies of player i consist of the cross product $\times_{I_{i,j} \in I_i} \chi(I_{i,j})$.

Normal-form games

- We can represent any normal form game.



  - Note that it would also be the same if we put player 2 at the root node.

Induced Normal Form

- Same as before: enumerate pure strategies for all agents
- Mixed strategies are just mixtures over the pure strategies as before.
- Nash equilibria are also preserved.
- Note that we've now defined both mapping from NF games to IIEF and a mapping from IIEF to NF.
    - what happens if we apply each mapping in turn?
    - we might not end up with the same game, but we do get one with the same strategy spaces and equilibria.

Randomized Strategies

- It turns out there are two meaningfully different kinds of randomized strategies in imperfect information extensive form games
  - mixed strategies
  - behavioral strategies
- Mixed strategy: randomize over pure strategies
- Behavioral strategy: independent coin toss every time an information set is encountered

Lecture Overview

1. Recap
2. Repeated Games
3. Infinitely Repeated Games
4. Folk Theorem

Introduction

- Play the same normal-form game over and over
  - each round is called a "stage game"
- Questions we'll need to answer:
  - what will agents be able to observe about others' play?
  - how much will agents be able to remember about what has happened?
  - what is an agent's utility for the whole game?
- Some of these questions will have different answers for finitely- and infinitely-repeated games.

Finitely Repeated Games

- Everything is straightforward if we repeat a game a finite number of times
- we can write the whole thing as an extensive-form game with imperfect information
  - at each round players don't know what the others have done; afterwards they do
  - overall payoff function is additive: sum of payoffs in stage games

Notes

- Observe that the strategy space is much richer than it was in the NF setting
- Repeating a Nash strategy in each stage game will be an equilibrium in behavioral strategies (called a stationary strategy)
- In general strategies adopted can depend on actions played so far
- We can apply backward induction in these games when the normal form game has a dominant strategy.

Lecture Overview

Infinitely Repeated Games

- Consider an infinitely repeated game in extensive form:
  - an infinite tree!
- Thus, payoffs cannot be attached to terminal nodes, nor can they be defined as the sum of the payoffs in the stage games (which in general will be infinite).

Definition

Given an infinite sequence of payoffs $r_1, r_2, \ldots$ for player i, the average reward of i is

$$\lim_{k \to \infty} \sum_{j=1}^{k} \frac{r_j}{k}.$$

Discounted reward

Definition

Given an infinite sequence of payoffs $r_1, r_2, \ldots$ for player i and discount factor β with $0 \leq \beta \leq 1$, i's future discounted reward is

$$\sum_{j=1}^{\infty} \beta^j r_j.$$

- Interpreting the discount factor:
  1. the agent cares more about his well-being in the near term than in the long term
  2. the agent cares about the future just as much as the present, but with probability $1 - \beta$ the game will end in any given round.
- The analysis of the game is the same under both perspectives.

Strategy Space

- What is a pure strategy in an infinitely-repeated game?

Strategy Space

- What is a pure strategy in an infinitely-repeated game?
  - a choice of action at every decision point
  - here, that means an action at every stage game
  - ...which is an infinite number of actions!
- Some famous strategies (repeated PD):
  - Tit-for-tat: Start out cooperating. If the opponent defected, defect in the next round. Then go back to cooperation.
  - Trigger: Start out cooperating. If the opponent ever defects, defect forever.

Nash Equilibria

- With an infinite number of equilibria, what can we say about Nash equilibria?
  - we won't be able to construct an induced normal form and then appeal to Nash's theorem to say that an equilibrium exists
  - Nash's theorem only applies to finite games
- Furthermore, with an infinite number of strategies, there could be an infinite number of pure-strategy equilibria!
- It turns out we can characterize a set of payoffs that are achievable under equilibrium, without having to enumerate the equilibria.

Folk Theorem

Theorem (Folk Theorem)

Consider any n-player game G and any payoff vector $(r_1, r_2, \ldots, r_n)$.

1. *If r is the payoff in any Nash equilibrium of the infinitely repeated G with average rewards, then for each player i, r_i is enforceable.*
2. *If r is both feasible and enforceable, then r is the payoff in some Nash equilibrium of the infinitely repeated G with average rewards.*

Evolutionary games

Strategies and populations

Strategies and populations

- Non-cooperative game theory:

Strategies and populations

- Non-cooperative game theory:
  Selfish players are involved in a game and each of them has to decide between different strategies in order to maximize an individual payoff, which depends on the strategies of all players

Strategies and populations

- Non-cooperative game theory:
  Selfish players are involved in a game and each of them has to decide between different strategies in order to maximize an individual payoff, which depends on the strategies of all players
- Evolutionary game theory:
  - Entire groups of players are involved in a game.
  - Each group is programmed to use some strategy.
  - Individuals meet (at pairs) and modify payoff $\rightarrow$ fitness
  - Overall fitness depends on the frequency of each strategy within the overall population.
  - This yields a feedback loop whose dynamics is the object of evolutionary game theory.

Strategies and populations

- Non-cooperative game theory:
  Selfish players are involved in a game and each of them has to decide between different strategies in order to maximize an individual payoff, which depends on the strategies of all players
- Evolutionary game theory:
  - Entire groups of players are involved in a game.
  - Each group is programmed to use some strategy.
  - Individuals meet (at pairs) and modify payoff $\rightarrow$ fitness
  - Overall fitness depends on the frequency of each strategy within the overall population.
  - This yields a feedback loop whose dynamics is the object of evolutionary game theory.
- Interaction is modeled by a 2-player strategic game.

Linear algebra notation

Consider a 2-player game $\Gamma = (A_1, A_2, u_1, u_2)$

Linear algebra notation

Consider a 2-player game $\Gamma = (A_1, A_2, u_1, u_2)$

We assume that the game is symmetric

Linear algebra notation

Consider a 2-player game $\Gamma = (A_1, A_2, u_1, u_2)$

We assume that the game is symmetric

- Players have the same set of strategies: $A_1 = A_2$ and $n = |A_1|$.
- $R = C^T = A$ (a $n \times n$ matrix)

Symmetric Nash equilibrium

Symmetric Nash equilibrium

- $x \in S = \Delta(A_1)$ is a probability distribution:
$$x = (x_1, \ldots, x_n),\ x_i \geq 0 \text{ and } x_1 + \cdots + x_n = 1$$

Symmetric Nash equilibrium

- $x \in S = \Delta(A_1)$ is a probability distribution:
  $$x = (x_1, \ldots, x_n),\ x_i \geq 0 \text{ and } x_1 + \cdots + x_n = 1$$
- Utilities can be described, for $x, y \in \Delta$, as

Symmetric Nash equilibrium

- $x \in S = \Delta(A_1)$ is a probability distribution:
$$x = (x_1, \dots, x_n),\ x_i \geq 0 \text{ and } x_1 + \cdots + x_n = 1$$
- Utilities can be described, for $x, y \in \Delta$, as
$$U_1(x, y) = x^T A\, y \text{ and } U_2(x, y) = x^T A^T\, y$$

Symmetric Nash equilibrium

- $x \in S = \Delta(A_1)$ is a probability distribution:
  $$x = (x_1, \ldots, x_n),\ x_i \geq 0 \text{ and } x_1 + \cdots + x_n = 1$$
- Utilities can be described, for $x, y \in \Delta$, as
  $$U_1(x,y) = x^T A\, y \text{ and } U_2(x,y) = x^T A^T\, y$$
- Nash conditions:
  $z^T A\, y \leq x^T A\, y$, for $z \in S$, and
  $x^T A^T\, y \leq x^T A^T\, z$, for $z \in S$,

Symmetric Nash equilibrium

- $x \in S = \Delta(A_1)$ is a probability distribution:
  $$x = (x_1, \ldots, x_n),\ x_i \geq 0 \text{ and } x_1 + \cdots + x_n = 1$$
- Utilities can be described, for $x, y \in \Delta$, as
  $$U_1(x, y) = x^T A\, y \text{ and } U_2(x, y) = x^T A^T\, y$$
- Nash conditions:
  $z^T A\, y \leq x^T A\, y$, for $z \in S$, and
  $x^T A^T\, y \leq x^T A^T\, z$, for $z \in S$,
- Symmetric Nash (x, x) conditions:
  $z^T A\, y \leq x^T A\, x$, for $y, z \in S$.

Replicator dynamics

- Population of n types.
- State of the population $x = (x_1, \dots, x_n) \in \Delta$.
- Assume that x_i are differentiable functions of time t.
- Individuals encounter randomly and engage in a symmetric game with payoff matrix A.
- Expected payoff for an individual of type i:

$$(Ax)_i = a_{i1}x_1 + \cdots + a_{in}x_n$$

- Average payoff in the population state x:

$$x^T A x$$