Games

CS 470 Introduction To Artificial Intelligence

Daqing Yi

Department of Computer Science
Brigham Young University

Outline

Outline

Multi-Agent in a Game

- $s \in$ State : states
    - S_0 : initial state
    - $S_terminate$: terminal state
- $p \in \textsc{Player}$: players
- $a \in \textsc{Actions}$: actions
- $c \in \textsc{Consequence}(s, a)$: consequences - transition model
- $u \in \textsc{Utility}(c, p)$: utility - utility function

Payoff matrix

- Two players
- Player 1 has m actions
- Player 2 has n actions
-

P2 / P1	a_1^{P2}	$\cdots$	a_n^{P2}
a_1^{P1}	$(u_{1,1}^{P1}, u_{1,1}^{P2})$	$\cdots$	$(u_{1,n}^{P1}, u_{1,n}^{P2})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_m^{P1}	$(u_{m,1}^{P1}, u_{m,1}^{P2})$	$\cdots$	$(u_{m,n}^{P1}, u_{m,n}^{P2})$

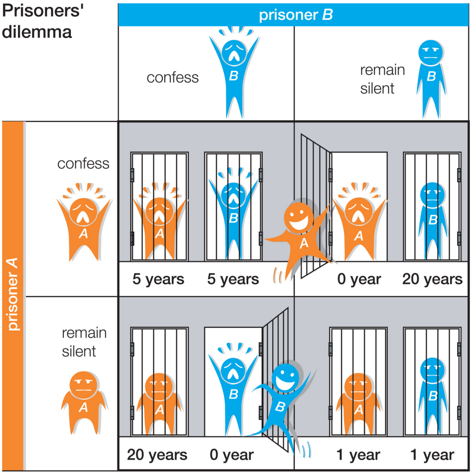
Zero-sum game

- **Zero-sum game**
  - The total payoff to all players is the same for every instance of the game
  - Equivalence : the sum is zero
  - One player's profit is the other player's loss
  - Show only the utility of Player 1 in the payoff matrix

$P1 \backslash P2$	a_1^{P2}	$\cdots$	a_n^{P2}
a_1^{P1}	$u_{1,1}^{P1}$	$\cdots$	$u_{1,n}^{P1}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_m^{P1}	$u_{m,1}^{P1}$	$\cdots$	$u_{m,n}^{P1}$

- **Non zero-sum game**

Prisoners' dilemma



Battle of sexes

		Woman	
		Baseball	Ballet
Man	Baseball	(3, 2)	(1, 1)
	Ballet	(0, 0)	(2, 3)

Outline

Strategic dominance

For player i, a strategy $s^* \in S_i$ and another strategy $s' \in S_i$

- s^* **weakly dominates** s'
  - for any strategy of the other player S_i
  - None worse $\forall s_{-i} \in S_{-i}, u_i(s^*, s_{-i}) \geq u_i(s', s_{-i})$
  - At least one better $\exists s_{-i} \in S_{-i}, u_i(s^*, s_{-i}) > u_i(s', s_{-i})$
- s^* **strictly dominates** s'
  - for any strategy of the other player S_i
  - All better $\forall s_{-i} \in S_{-i}, u_i(s^*, s_{-i}) > u_i(s', s_{-i})$

Pareto dominance

strategy profile

- a set of strategies for all the players
- one and only strategy for every player
- (s_i, s_{-i})

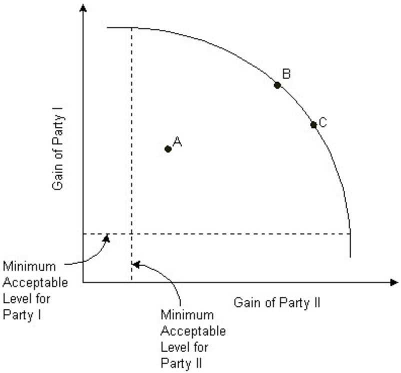
Strategy profile S **Pareto dominates** strategy profile S'

- no player gets a worse payoff with S than with S'
  $\forall i, U_i(S) \geq U_i(S')$
- at least one player gets a better payoff with S than with S'
  $\exists i, U_i(S) > U_i(S')$

Pareto optimal

Strategy profile S^* is **Pareto optimal**

- No there strategy S' that Pareto dominates S^*



Nash equilibria

For player i, s_i is a **best response** to S_{-i}

- $\forall s_i' \in S_i, U_i(s_i, S_{-i}) \geq U_i(s_i', S_{-i})$

A strategy profile $S = (s_1, \cdots, s_n)$ is a **Nash equilibrium**

- no agent can do by better unilaterally changing his/her strategy
- $\forall i, s_i$ is a best response to S_{-i}.

Maximin strategy

Maximin strategy

- maximizes a player's worst possible outcome
- $s^* = \arg\max_{s \in S} \min_{t \in T} u(s, t)$

Minimax strategy

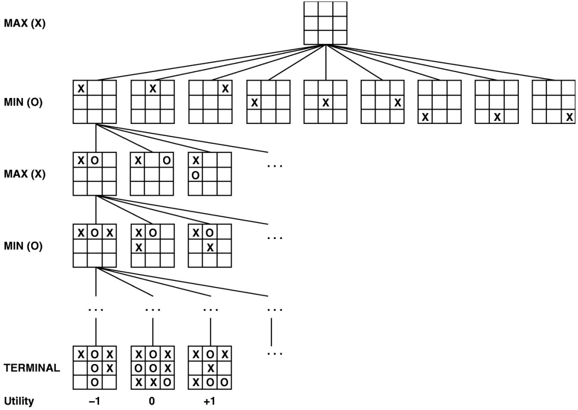
Minimax strategy

- minimizes opponent player's best possible outcome
- $t^* = \arg\min_{t \in T} \max_{s \in S} u(s, t)$
- Minimax theorem - John Von Neumann
  - zero-sum game
  - optimal strategy

Outline

Turn-taking games

Tic-Tac-Toe game



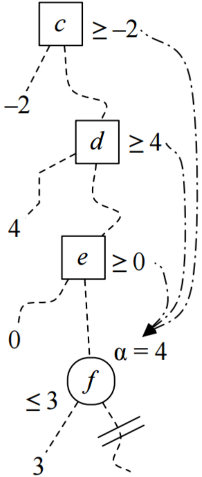
Turn-taking games

- **game tree**
  - initial state S_0 is the root
  - each state maps to a node
  - each action maps to an edge
  - minimax tree
    - optimal strategy - minimax
- **evaluation function** approximates the utility of a state without a complete search

Alpha-beta pruning

- exponential growth in minimax tree
- pruning : removes away branches that cannot possibly influence the final decision
- minimax search $\Rightarrow$ depth-first
  - Alpha-beta pruning

Alpha-beta pruning



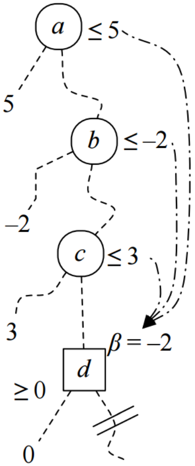
Alpha cutoff

- $\alpha =$ the value of the best choice at any node for the MAX algorithm
- biggest lower bound
- Example
  - At node f, MAX get utility ≤ 3
  - At node f, $\alpha = 4$
  - Node f will never be reached
  - Pruning Node f

Alpha-beta pruning

Beta cutoff

- $\beta =$ the value of the best choice at any node for the MIN algorithm
- smallest upper bound
- Example
  - At node *d*, MIN get utility ≥ 0
  - At node *d*, $\beta = -2$
  - Node *d* will never be reached
  - Pruning Node *d*



Alpha-beta pruning

Example