Game Playing AI

Background

Heuristics

In game playing, **heuristics** means a practical method that not guaranteed to be perfect or optimal, but could find a satisfactory solution with **limited information** at the **current stage without digging any further**.

Bayes' Theorem

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

$P(A|B)$: the probability of event A happen if B is true.

Monty Hall Problem

Suppose **Door B (Goat)** is open after you chose **Door A**. Now the probabilty of Car in **Door C** is:

$$\begin{aligned} P(Car\ in\ C|Open\ B) &= \frac{P(Open\ B|Car\ in\ C) \times P(Car\ in\ C)}{P(Open\ B)} \\ &= \frac{1 \times \frac{1}{3}}{P(Open\ B)} \end{aligned}$$

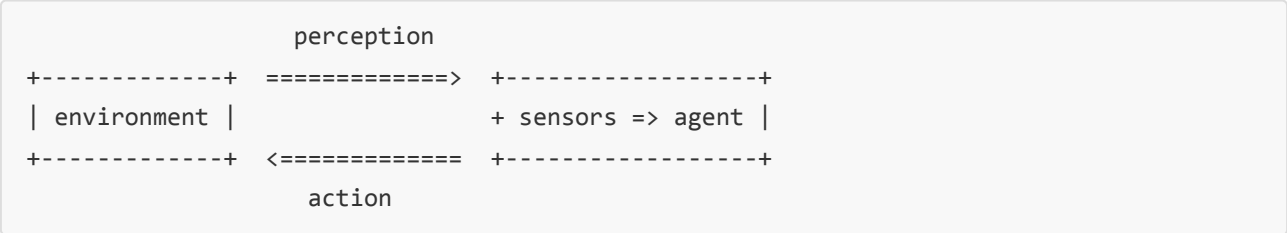
$$\begin{aligned} P(Open\ B) &= P(Car\ in\ A) \times P(Open\ B|Car\ in\ A) \\ &\quad + P(Car\ in\ B) \times P(Open\ B|Car\ in\ B) \\ &\quad + P(Car\ in\ C) \times P(Open\ B|Car\ in\ C) \\ &= \frac{1}{3} \times \frac{1}{2} + \frac{1}{3} \times 0 + \frac{1}{3} \times 1 \\ &= \frac{1}{2} \end{aligned}$$

$$P(Car\ in\ C|open\ B) = \frac{1}{3}$$

Introduction to AI

Intelligence

Cognition



- **States**: a snapshot of useful information from environment
- **Goal State**: the state AI try to achieve

Types of AI Problems

- **Fully vs Partially Observable** (Tic Tac Toe vs Battleship)
- **Stochastic vs Deterministic** (Poker vs Chess)
- **Continious vs Discrete** (Recognizing hand writing vs Poker)
- **Adversarial vs Benign** (Chess vs Self driving car)

Game Tree

The total number of nodes in the game tree is $\mathbf{b^d}$ where **b** is the **Braching Factor** and **d** is the depth of the game tree.

Average Braching Factor could be computer by playing the game randomly and record the total branches and total move played.

When the number of total nodes are too big, **Depth-Limited Search** is used to find result in given time limit.

$$d_{max} = log_{avg\ b}\ f_{CPU} \times t_{allowed}$$

Testing Evaluation Function

Test the evaluation function with different level of depth. By digging further into the depth, if **State of Quiescence** is achieved, that means the evaluation function is good.

Algorithms

Iterative Deepening

Evaluate each level before going any deeper, similar to **Breadth First Search**, more efficient with **big branching factor**, due to the **exponential** growth in time which is dominated by the deepest level searched.

$$number\ of\ nodes = \frac{b^{d+1} - 1}{b - 1}$$

with the **least branching factor of 2**, iterative deepening visits less than $2n$ nodes.

Resources:

- University of British Columbia's slides introducing the topic.
- 3.4.5 of Russel, Norvig (AIMA)
- A set of videos showing visually how Iterative Deepening is different from DFS in practice.

Minimax

Aggregate the leafs from bottom up, with alternating max and min level. At max level, return the max value of all branches, vice versa.

Alpha-Beta Pruning

Prune the branch that will not affect the upper level's bounds.

- Alpha: lower bound
- Beta: upper bound

Branch can be pruned when $\beta_{max} > \alpha_{min}$ between two consecutive levels.