Markov Decision Processes (MDPs)

Course Overview: Where are we?

- Introduction & Agents
- Search and Heuristics
- Adversarial Search
- Logical Knowledge Representation
- Markov Decision Processes (MDPs)
- Reinforcement Learning
- Uncertainty & Bayesian Networks
- Machine Learning
MDPs

Markov Decision Processes

- Planning Under Uncertainty
- Mathematical Framework
- Bellman Equation
- Value Iteration
- Policy Iteration
- Reinforcement Learning

Andrey Markov
(1856-1922)

Planning Agent

Environment

Static vs. Dynamic

Fully vs. Partially Observable

Deterministic vs. Stochastic

Percepts → What action next? → Actions
Review: Expectimax

- What if we don’t know what the result of an action will be? E.g.,
  - In Solitaire, next card is unknown
  - In Pacman, the ghosts act randomly

- Can do expectimax search
  - Max nodes as in minimax search
  - Chance nodes, like min nodes, except the outcome is uncertain - take average (expectation) of children
  - Calculate expected utilities

- Today, we formalize this as a Markov Decision Process
  - Handles intermediate rewards & infinite search trees
  - More efficient processing

Example: Grid World

- Walls block the agent’s path
- Agent’s actions are noisy:
  - 80% of the time, North action takes the agent North (assuming no wall)
  - 10% - actually go West
  - 10% - actually go East
  - If there is a wall in the chosen direction, the agent stays put
- Small “living” penalty (e.g., -0.04) each step
- Big reward/penalty (e.g., +1 or -1) comes at the end
- Goal: maximize sum of rewards
Markov Decision Processes

- An MDP is defined by:
  - A set of states $s \in S$
  - A set of actions $a \in A$
  - A transition function $T(s,a,s')$
    - Probability that action $a$ in $s$ leads to $s'$
    - i.e., $P(s' | s,a)$
    - Also called “the model”
  - A reward function $R(s, a, s')$
    - Sometimes just $R(s)$ or $R(s')$
  - A start state
  - Maybe a terminal state

What is Markov about MDPs?

- “Markov” generally means that
  - Given the present state, the future is independent of the past
  
- For Markov decision processes, “Markov” means:

\[
P(S_{t+1} = s'| S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1}, \ldots, S_0 = s_0) = P(S_{t+1} = s'| S_t = s_t, A_t = a_t)
\]

Next state only depends on current state and action

Andrey Markov (1856-1922)
Solving MDPs

- In deterministic search problems, want an optimal path or plan (sequence of actions) from start to a goal
- MDP: Stochastic actions, don’t know what next state will be
- Instead of path/plan, use an optimal policy $\pi^*$: $S \rightarrow A$
  - Policy $\pi$ prescribes an action for every state
  - Defines a reflex agent
  - An optimal policy maximizes expected reward if followed

Optimal policy when

$R(s, a, s') = -0.04$

for all non-terminals $s$
More Example Optimal Policies

Conservative

Aggressive

Suicidal

Example: High-Low Card Game
Example: High-Low

- Suppose three card types: 2, 3, 4
  - Infinite deck, twice as many 2’s
- Start with 3 showing
- After each card, say “high” or “low”
- New card is revealed
  - If you’re right, you win the points shown on the new card
  - Tie: no reward, choose again
  - If you’re wrong, game ends

- Differences from expectimax problems:
  - #1: get rewards as you go
  - #2: you might play forever!

High-Low as an MDP

- States:
  - 2, 3, 4, done
- Actions:
  - High, Low
- Model: \(T(s, a, s’):\)
  - \(P(s’=4 \mid 4, \text{Low}) = 1/4\)
  - \(P(s’=3 \mid 4, \text{Low}) = 1/4\)
  - \(P(s’=2 \mid 4, \text{Low}) = 1/2\)
  - \(P(s’=\text{done} \mid 4, \text{Low}) = 0\)
  - \(P(s’=4 \mid 4, \text{High}) = 1/4\)
  - \(P(s’=3 \mid 4, \text{High}) = 0\)
  - \(P(s’=2 \mid 4, \text{High}) = 0\)
  - \(P(s’=\text{done} \mid 4, \text{High}) = 3/4\)
  - ...

- Rewards: \(R(s, a, s’):\)
  - Number shown on \(s’\) if \(s’ > s\) \(\wedge a=\text{“High” etc.}\)
  - 0 otherwise
  - Start: 3
Next Time

- Value iteration
- Finding the optimal policy
- To Do
  - Read chapters 13 and 17