Stanford University

AA228/CS238: Decision Making Under Uncertainty

Winter 2023

Liam Kruse • 320-105 (Braun Hall) • email: *lkruse@stanford.edu*

PROBLEM SESSION 4: EXACT SOLUTION METHODS **February 1, 2023 4:30pm PT**

Topic 1. MDP Overview

a) Markov Decision Process (MDP): defined by the tuple $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$

- $\mathcal{S}$ - State Space: the environment, the *minimum information set* required to make a decision

  - Grid World
  - $(x, y, \theta, \dot{x}, \ddot{x})$
  - Discrete or continuous (or mixed!)

- $\mathcal{A}$ - Action Space: what the agent can do

  - Grid World actions: $\leftarrow, \uparrow, \rightarrow, \downarrow$
  - Driving actions: $\dddot{x}, \dot{\theta}$
  - Discrete ($\dddot{x} \in \{-0.3, 0.0, 0.3\}$), continuous ($\dddot{x} \in [-0.3, 0.3]$), or mixed

- T - Transition model: system dynamics (how the system evolves)

  - Tables (only feasible for small discrete problems)
  - Generative model $s' \sim T(s, a)$; $\quad x^{t+1} = x^t + v^t \Delta t + \frac{1}{2}\dot{v}^t \Delta t^2$

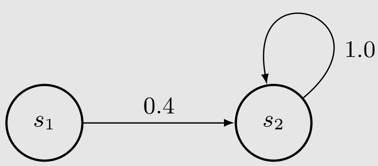


Figure 1: Simple MDP

	s_1	s_2
s_1	0.6	0.4
s_2	0.0	1.0

Figure 2: Transition Model

- R - Reward Function: expected reward from taking action a in state s and transitioning to state s'

  - Example: $R(s, a, s') = -\lambda_1 \times \text{hasCollided} + \lambda_2 \times |\dddot{x}|$
  - Reward shaping: crafting a reward function to achieve desired behavior

- γ - Discount factor: used to weight future rewards

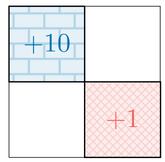
  - $\gamma \in [0, 1)$
  - Used to make an agent more or less *myopic*.

b) Utility: a discounted sequence of rewards

- Utility of a sequence of states **without** discounting: why is this problematic?

$$U\left([s_1, s_2, \ldots, s_n)]\right) = \sum_{t=1}^{n} r_t$$

- Thought exercise: would an agent want to collect rewards in the blue cell (bricks) or the red cell (crosshatch) forever?



Is there a preference for $10 + 10 + 10 + \ldots$ or $1 + 1 + 1 + \ldots$ as $n \to \infty$ ("infinite horizon")?

- Solution: discount with γ!

$$U\left([s_1, s_2, \ldots, s_n)]\right) = \sum_{t=1}^{n} \gamma^{t-1} r_t, \qquad \gamma \in [0, 1)$$

c) Policy π: a function of the state that tells you *what to do* in every state

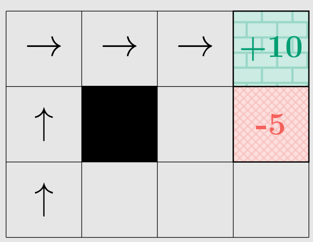


Figure 3: Optimal Path

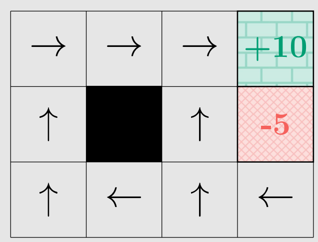


Figure 4: Optimal Policy

- Optimal Path
  - Solution to A* Search, Dijkstra's Algorithm
  - Falls apart if we end up in a new state due to outcome uncertainty
- Optimal Policy
  - Solution to (PO)MDPs
  - Tells us what to do in EVERY state

- $U^\pi(s) \to$ utility from executing policy π from state s (the *value function*)
- $\pi^*(s) = \arg\max_\pi U^\pi(s)$

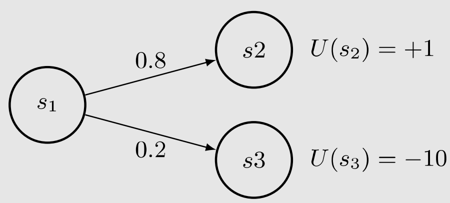
d) Bellman Equation: "The expected utility of a state is the reward at that state plus the discounted sum of expected future rewards."

$$U_{k+1}(s) = \max_a \left(\underbrace{R(s,a)}_{①} + \underbrace{\gamma}_{②} \underbrace{\sum_{s'} T(s' \mid s, a) U_k(s')}_{③} \right)$$

① reward at current state
② discount factor
③ expected utility at next state

e) A Note On Expectation

An expected value is a *weighted average*



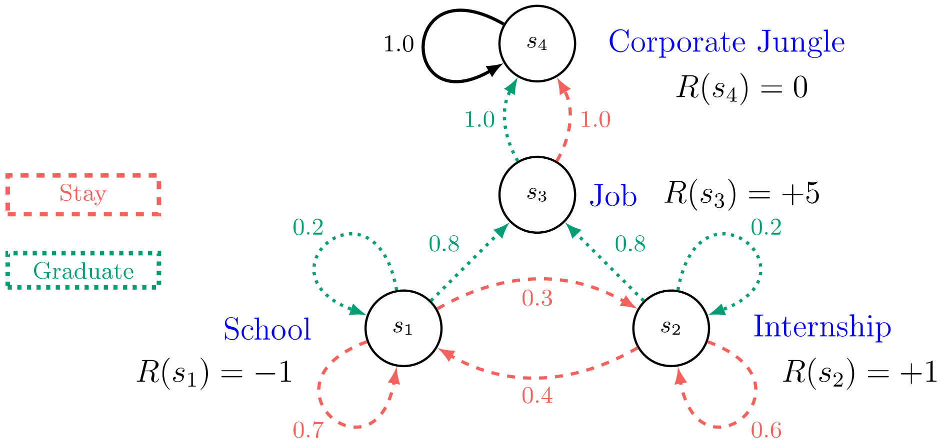
What is the expected utility when transitioning out of s_1?

$$\mathbb{E}[U] = (0.8)(1) + (0.2)(-10) = -1.2$$

Topic 2. Value Iteration Example

Algorithm 1 The Value Iteration Algorithm

procedure VALUE ITERATION($\mathcal{P}$ :: MDP, $k_{\max}$)
    $U(s) \leftarrow 0$ for all $s \in \mathcal{S}$
    for $k \leftarrow 1, k_{\max}$ **do**
        for all $s \in \mathcal{P}.\mathcal{S}$ **do**
            $U_{k+1}(s) = \max_a \left(R(s,a) + \gamma \sum_{s'} T(s' \mid s, a) U_k(s')\right)$
        end for
    end for
end procedure



a) Define the tuple for this MDP

- $\mathcal{S}$: School, Job, Internship (Corporate Jungle - *Absorbing State*)
- $\mathcal{A}$: **Stay**, **Graduate**
- T:

	s_1	s_2	s_3	s_4
s_1	0.7	0.3	0.0	0.0
s_2	0.4	0.6	0.0	0.0
s_3	0.0	0.0	0.0	1.0
s_4	0.0	0.0	0.0	1.0

Figure 5: **Stay**

	s_1	s_2	s_3	s_4
s_1	0.2	0.0	0.8	0.0
s_2	0.0	0.2	0.8	0.0
s_3	0.0	0.0	0.0	1.0
s_4	0.0	0.0	0.0	1.0

Figure 6: **Graduate**

  - Note: rows must sum to 1!
  - For a discrete problem: Need $|\mathcal{A}|$ tables of size $|\mathcal{S}|^2$

- R: $R(s_1) = -1, R(s_2) = +1, R(s_3) = +5, R(s_4) = 0$

b) Perform two iterations of value iteration:

Iteration 1:

$$U_1(s_1) = -1 + 0.9 \max_a \{\underbrace{0.7 \times 0 + 0.3 \times 0}_{\text{Stay: } 0.0}, \underbrace{0.2 \times 0 + 0.8 \times 0}_{\text{Grad: } 0.0}\} = -1$$

$$U_1(s_2) = +1 + 0.9 \max_a \{\underbrace{0.6 \times 0 + 0.4 \times 0}_{\text{Stay: } 0.0}, \underbrace{0.2 \times 0 + 0.8 \times 0}_{\text{Grad: } 0.0}\} = +1$$

$$U_1(s_3) = +5 + 0.9 \max_a \{\underbrace{0}_{\text{Stay: } 0.0}, \underbrace{0}_{\text{Grad: } 0.0}\} = +5$$

Iteration 2:

$$U_1(s_1) = -1 + 0.9 \max_a \{\underbrace{0.7 \times -1 + 0.3 \times 1}_{\text{Stay: -0.4}}, \underbrace{0.8 \times 5 + 0.2 \times -1}_{\text{Grad: 3.8}}\} = 2.42$$

$$U_1(s_2) = +1 + 0.9 \max_a \{\underbrace{0.6 \times 1 + 0.4 \times -1}_{\text{Stay: 0.2}}, \underbrace{0.8 \times 4 + 0.2 \times 1}_{\text{Grad: 4.2}}\} = 4.78$$

$$U_1(s_3) = +5 + 0.9 \max_a \{\underbrace{0}_{\text{Stay: 0.0}}, \underbrace{0}_{\text{Grad: 0.0}}\} = +5$$

c) What is our policy after two rounds of value iteration?

$$\pi = \{(s_1, a_2), (s_2, a_2), (s_3, N/A)\}$$

d) What is the time complexity of value iteration?

$$\mathcal{O}\left(|\mathcal{S}|^2|\mathcal{A}|\right)$$