CS 4180/5180: Reinforcement Learning and Sequential Decision Making (Fall 2024)

Exercise 2: Markov Decision Processes

By:

Srinivas Peri

1.a)

- State Space S depicts every state the agent could be in inside the "four-room domain." Every state has a corresponding location or configuration in the environment. A set of all feasible (x, y) coordinates inside the grid world could be defined as S.
- Area of Action A is a representation of every action the agent might do in any state. One can move throughout the "four-room domain" in a variety of directions, including up, down, left, and right.
- A = {left, right, up, down}

1.b)

- Interior states: 9 states/room × 4 rooms × 4 actions/state = 144 non-zero rows
- Wall states: 16 states/room × 4 rooms × 3 actions/state = 192 non-zero rows
- Corner states: 4 states/room × 4 rooms × 2 actions/state = 32 non-zero rows
- Goal state transition: The transition from (10, 10) to (1, 1) adds 1 non-zero row since the agent returns to the start state with probability 1 after reaching the goal.
- So the total number of non-zero rows without considering the door states would be:
- 144 +192+32+1=369 non-zero rows

1.c)

2.a)

2a) Episodic task with discounting:-

$G_t = R_{t+1} + \gamma G_{t+1}$

$G_t = \sum_{K=0}^{\infty} \gamma^K R_{t+K+1}$

$R_{t+k+1} = 0$ for all k except the one instance where $t+k+1 = T$, we only have one non-zero term in the sum

$\therefore t+K+1 = T$ at $R_T = -1$

Discount factor for this reward is

$\therefore (K = T-t-1) \therefore \gamma^K = \gamma^{T-t-1}$

we count $t+1$ rather than t hence, γ^{T-t}

$G_t = \gamma^{T-t} \cdot R_T$

At fail state $R_T = -1$

$\therefore G_t = -\gamma^{T-t}$

2.b) For every episode, the reward stays consistent regardless of the duration it takes to finish the task. This indicates that completing the maze more quickly does not yield additional rewards. Consequently, this reward structure does not effectively encourage the agent to seek quicker or more efficient solutions. Without the incentive to optimize its path, the agent has no reason to improve its performance in terms of speed. Hence its not effectively communicated with the agent want it needs to achieve.

3.a,b)

3.a) $\gamma = 0.5$

$R_1 = -1,\ R_2 = 2,\ R_3 = 6,\ R_4 = 3,\ R_5 = 2$

$T = 5,\quad G_t = R_{t+1} + \gamma G_{t+1}$ (Hint: from backward

$G_5 = 0,\quad G_4 = R_5 + \gamma G_5 = 2 + 0 = 2$

$G_3 = R_4 + \gamma G_4 = 3 + \frac{1}{2}(2) = 4$

$G_2 = R_3 + \gamma G_3 = 6 + \frac{1}{2}(4) = 8$

$G_1 = R_2 + \gamma G_2 = 2 + \frac{1}{2}(8) = 6$

$G_0 = R_1 + \gamma G_1 = -1 + \frac{1}{2}(6) = \underline{\underline{2}}$

3.b) $\gamma = 0.9$

$R_1 = 2$

$R_2, R_3, \ldots R_{\infty}$, all 7's

$\Rightarrow$ For Constant reward

$\Rightarrow G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$ (eq 3.10)

$\therefore G_1 = \sum_{k=0}^{\infty} (0.9)^k\, 7 = \frac{7}{1-0.9} = 70$

$G_0 = R_1 + \gamma G_1 = 2 + (0.9)(70) = \underline{\underline{65}}$

4)

4.) $Up\_G_t = 50 + \gamma(-1) + \gamma^2(-1) + \gamma^3(-1) \ldots \gamma^{100}(-1)$

$$= 50 - \sum_{i=1}^{100} \gamma^i$$

$$Down\_G_t = -50 + \sum_{i=1}^{100} \gamma^i$$

To figure out the threshold value of γ where the agent will be indifferent b/w going up & down can be obtained by equating up_int = down_int

$$\therefore \quad 50 - \sum_{i=1}^{100} \gamma^i = -50 + \sum_{i=1}^{100} \gamma^i$$

$$\left[\begin{array}{l} 100 = 2 \sum_{i=1}^{100} \gamma^i \\ 50 = \sum_{i=1}^{100} \gamma^i \end{array} \right]$$

From Constant rewards

$$\sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$$

$$50 - \frac{1}{1-\gamma} = -50 + \frac{1}{1-\gamma}$$

$$100 = \frac{2}{1-\gamma} \Rightarrow 50 = \frac{1}{1-\gamma}$$

$$\gamma = \frac{49}{50} = 0.98$$

5.a)

5a) $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ $\rightarrow$ eq (1) (3.8 Equation)

$\Rightarrow \bar{G}_t = \sum_{k=0}^{\infty} \gamma^k (R_{t+k+1} + c)$ $\rightarrow$ eq (2) (Adding constant c)

$\Rightarrow \bar{G}_t = \sum_{k=0}^{\infty} \gamma^k (R_{t+k+1}) + \sum_{k=0}^{\infty} \gamma^k c$ — eq (3)

$v_\pi(s) = \mathbb{E}_\pi [G_t | S_t = s]$ eq (3.12)

$\bar{v}_\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} + \sum_{k=0}^{\infty} \gamma^k c \middle| S_t = s \right]$

$\bar{v}_\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \right] + \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k c \middle| S_t = s \right]$

from eq (1) eq (3) becomes

$\bar{G}_t = G_t + \sum_{k=0}^{\infty} \gamma^k c$

$= G_t + \frac{c}{1-\gamma}$

$\therefore \Rightarrow \bar{v}_\pi(s) = \mathbb{E} \left[G_t + \frac{c}{1-\gamma} \middle| S_t = s \right]$

5.b) $G_t = \sum_{k=0}^{T} R_{t+k+1}$

$\bar{G}_t = \sum_{k=0}^{T} (R_{t+k+1} + c) = G_t + cT$

$\bar{v}_\pi(s) = \mathbb{E}_\pi [G_t | S_t = s]$

$\bar{v}_\pi(s) = \mathbb{E}_\pi [\bar{G}_t + S_t = s] =$

$= v_\pi(s) + c\,\mathbb{E}_\pi [T | S_t = s]$ //

5.b)

For an episodic task, such as maze running, adding a constant c to all rewards changes the task because the sum of the rewards is bounded by the length of the episode. In a continuing task with an infinite horizon, the effect of adding a constant to all rewards is offset over an infinite number of time steps, which results in a constant shift in the state-value functions. However, in an episodic task, the rewards accumulate over a finite number of steps, which means that adding a constant to the rewards can change the total return for an episode and potentially affect the optimal policy.

6 a,b)

6a) $V_\pi(s) = \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)\left[r + \gamma E_\pi[G_{t+1}|S_{t+1}=s']\right]$

$$= \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a)\left[r + \gamma V_\pi(s')\right]$$

$$V_\pi(s) = 0.25(0 + 0.9 \times 2.3) + 0.25(0 + 0.9 \times 0.4)$$

$$+ 0.25(0 + 0.9 \times -0.4) + 0.25(0 + 0.9 \times 0.7)$$

$$= 0.25 \times 0.9(2.3 + 0.4 - 0.4 + 0.7)$$

$$= 0.2 \times 0.9(3.0)$$

$$\approx 0.68$$

reward = 0
optimal probability of each action $\Rightarrow$ 0.25 i.e (1/4)

6.b) $V_\pi(s) = 0.5(0 + 0.9 \times 19.8) + 0.5(0 + 0.9 \times 19.8)$

$$= 0.5 \times 0.9 \times (19.8 + 19.8)$$

$$\approx 17.82$$

0.5 $\rightarrow$ (considering the agent choose each of the optimal actions with equal probability)

7)

7a) $$v_\pi(A) = \overbrace{0.5(0+1)}^{\text{right}} + \overbrace{0.5(0+0)}^{\text{left}}$$

$$= 0.5$$

Value func'n of state A would be 0.5, because it has 50% chance of going either left or right to get the reword. As $\gamma = 1$ there is no discount and the value will be a full reword of 0.5

b) $$\Rightarrow v(e) = 0.5(0+v(d)) + 0.5(0+1) = 0.5(v(d)+1)$$

$$v(d) = 0.5(0+v(c)) + 0.5(0+v(e))$$
$$= 0.5(v(c)+v(e))$$

$$v(c) = 0.5(0+v(b)) + 0.5(0+v(d))$$
$$= 0.5(v(b)+v(d))$$

$$v(b) = 0.5(v(a)+v(c))$$

$$v(a) = 0.5(0) + 0.5(v(b)) = 0.5(v(b))$$

with the progressively increasing nature of the system and having no discounting the value function at state A = 0.5 due to its nature of choosing b/w two paths

$v(a) = 0.03125$

$v(b) = 0.0625$

$v(c) = 0.125$

$v(d) = 0.25$

$v(e) = 0.5$

c)

Mathematically, the value of a state i steps from the rightmost state can be expressed as the probability of reaching the rightmost state. Since each move is equiprobable, the agent has a 1/2 chance of moving right at each step. To reach the rightmost state from state i, the agent needs to make i right moves. The probability of making i right moves in a row is $\left(\frac{1}{2}\right)^i$

Therefore, the value function V for state i in an MDP with n states can be expressed as:

$$v(i) = (1\,/\,2)^i$$

8)

8a) $v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right]$

Given the parameters from domain

α is the probability of finding a can when searching

β is the " of robot staying in 'high' after ~~wait~~

r_{search} is the reward for searching

r_{wait} is the reward for waiting

For high state, the robot can 'search' or 'wait':

$v_\pi(\text{high}) = \pi(\text{search}|\text{high})\left[\alpha(r_{search} + \gamma v_\pi(\text{high})) + (1-\alpha)(-3 + \gamma v_\pi(\text{low}))\right] + \pi(\text{wait}|\text{high})\left[r_{wait} + \gamma\beta v_\pi(\text{high}) + \gamma(1-\beta) v_\pi(\text{low})\right]$

$v_\pi(\text{low}) = \pi(\text{wait}|\text{low})\left[r_{wait} + \gamma v_\pi(\text{low})\right] + \pi(\text{recharge}|\text{low})\left[\gamma v_\pi(\text{high})\right]$

8b) $\alpha = 0.8$, $\beta = 0.6$, $\gamma = 0.9$, $r_{search} = 10$, $r_{wait} = 3$

$\pi(search \mid high) = 1$ $\quad \therefore \pi(wait \mid high) = 0$

$\pi(wait \mid low) = 0.5$ $\quad \pi$ (lo

$\pi(recharge \mid low) = 0.5$

$\Rightarrow v_\pi(high) = 1\,[0.8(10 + 0.9\, v_\pi(high) + (1-0.8)(-3 + 0.9\, v_\pi(low))] + \bullet\,[- \; - \; -]$

$= 1\,[8 + 0.72\, v_\pi(high) + (0.2)(-3) + (-0.18\, v_\pi(h)$

on solving the equation's

$v_\pi(high) = 59.45$

$\Rightarrow v_\pi(low) = 0.5\,[3 + 0.9\, v_\pi(low)] + 0.5\,[0.9\, v_\pi(high)]$

on. solving the equation's

$v_\pi(low) = 61.37$

8c) $v_\pi(high) = \alpha\,(r_{search} + \gamma\, v_\pi(high)) + (1-\alpha)\,(-3 + \gamma\, v_\pi(low))$

$v_\pi(low) = \theta\,(r_{wait} + \gamma\, v_\pi(low)) + (1-\theta)\,(\gamma\, v_\pi(high))$

$$v_{high} = \frac{7.4 + 0.18\, v_\pi\, low}{0.28}$$

$$v_{low} : \frac{3\theta + 0.9\, v_\pi(high)(1-\theta)}{1 - 0.9\theta}$$

To find the optimal ~~(path)~~ θ, we differentiate $V_\pi(low)$ with respect to θ and set it to zero & solve for θ

on solving

$\therefore \theta = 0.616$ high & low $= 4.66, 7.77$