

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

**Exercise 2: Markov Decision Processes** 

By:

**Srinivas Peri** 

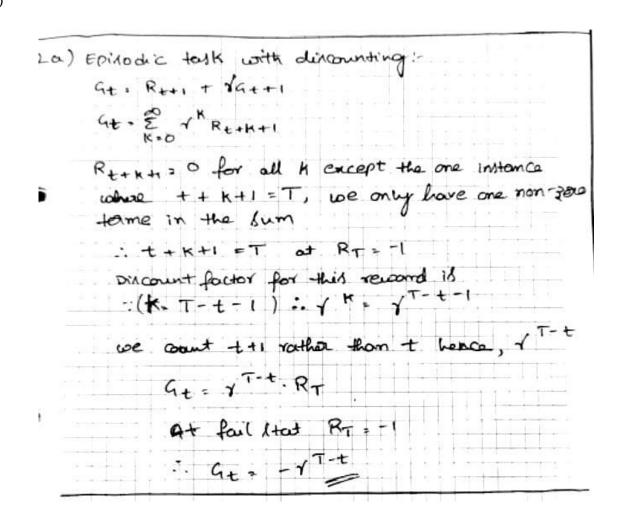
- State Space S depicts every state the agent could be in inside the "four-room domain." Every state has a corresponding location or configuration in the environment. A set of all feasible (x, y) coordinates inside the grid world could be defined as S.
- Area of Action A is a representation of every 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)



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)

3.0) 
$$\sqrt{1.0.5}$$
 $R_{1.7}-1$ ,  $R_{2.7}$ ,  $R_{3.7}$ 6,  $R_{4.7}$ 3,  $R_{5.7}$ 2

 $T=5$ ,  $G+1$ ,  $R_{4+1}+\sqrt{1.6}+1$  (From backward

 $G=0$ ,  $G_{4}$ ,  $R_{5}+\sqrt{1.6}+1$  2 + 0 = 2

 $G_{3.7}$   $R_{4}+\sqrt{1.6}+1$  3 +  $V_{2}(2)=\frac{4}{4}$ 
 $G_{2.7}$   $R_{3}+\sqrt{1.6}$  6 +  $V_{2}(4)=8$ 
 $G_{1.7}$   $R_{2}+\sqrt{1.6}$  2 +  $V_{2}(6)=6$ 
 $G_{0.7}$   $R_{1}+\sqrt{1.6}$  3 - 1 +  $V_{2}(6)=2$ 
 $R_{1.7}$   $R_{2.7}$   $R_{41}$   $R_{1.7}$   $R_{1.7}$   $R_{2.7}$   $R_{41}$   $R_{1.7}$   $R_{2.7}$   $R_$ 

4.) Up 
$$-4i - 50 + \sqrt{(-1)} + \sqrt{(-1)} + \sqrt{3}(-1) - \sqrt{100}(-1)$$

$$= 50 - \frac{600}{119} \sqrt{1}$$

$$00000, 4i = -50 + \frac{100}{5} + \frac{1}{11}$$

To figure out the threshold value of I whose the agent will be indifferent the going up & down a down to be obtained by command up but down!

Farom Grotant rewords

0

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)

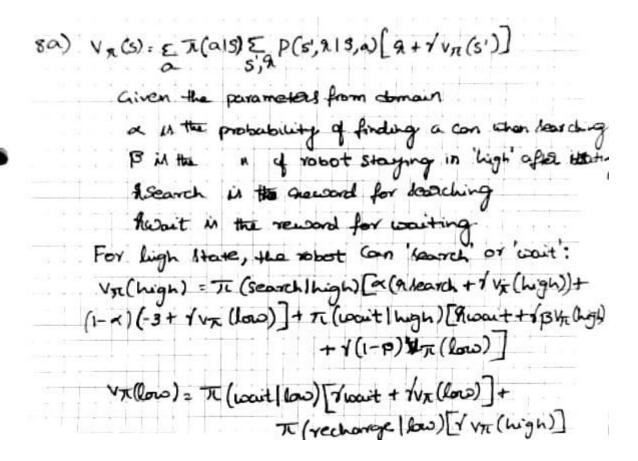
	gright gest !	
70)	) Vx(A) = 0.5 (0+1) + 0.5 (0+0)	
	= 0.5	15
	value funcin of state A wold be 0.5, becau	ge !
	it has 50% chance of going either left on 9 mg.	at .
	to get the accord. As $\sqrt{-1}$ there is no discount and the value will be a feul howard of	0.2
4		(d) H)
	V(d)205(0+V(c))+05(0+V(e)) = 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(5) 2 0.5 (v(a) + v(c))	
	v(a)2 0.5(0) + 0.5(v(b)) = 0.5(v(b))	
	with the progressively increasing native of the	TO.
	calue function at state A = 0.5 55 due to its natea of charming blue two paths	
	V(a) 2 0.03125	
	V(b) = 0.0625 V(c) 2 0.125	
	V(c) 2 0 25 V(e) 2 0 5	
	V(e) 2 0 5/	

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)



```
8b) α, 0.8, β, 0.6, ν = 0.9, 9, people = 10, 9, people = 3

π(λεστά | high) = 1 ... π (poolt | high) = 0

π (poolt | low) = 0.5 π (po

π (poolt | low) = 0.5

> νπ(high) = 1 [0.8 (10 + 0.9 νπ(high) + (1-0.8)(-3+0.9 νπ(high) + (0.2)(-3) + (-0.18 νπ(high) + (0.2)(-3) + (-0.18 νπ(high) = 59.45

> νπ (high) = 59.45

> νπ (low) = 0.5 [3 + 0.9 νπ(low)] + 0.5 [0.9 νπ (high)]

on lowing 13 the equation's

νπ (low) = 51.37
```

8C)  $V_{\pi}(high) = a (hsearch + V_{V_{\pi}}(high)) + (1-X)(-3+1/V_{\pi}(high))$   $V_{\pi}(how) = \theta (hsearch + 1/V_{\pi}(how)) + (1-\theta) (1/V_{\pi}(high))$   $V_{high} = 2.4 + 0.18 V_{\pi}(how)$   $V_{high} = 2.4 + 0.18 V_{\pi}(high)(1-\theta)$   $V_{high} = 3.4 + 0.18 V_{\pi}(high)(1-\theta)$   $V_{high} = 2.4 +$