CS 5180

Reinforcement Learning

Exercise 2
Sep 27, 2024
Anuj Patel 002874710

Northeastern University

Que 1) part a: State Space (S)

The State space S consists of all possible positions the agent can occupy in the environment, that are represented as coordinates pair (x, y). here $x \in [0, 10]$ and $y \in [0, 10]$ and $y \in [0, 10]$. The agent cannot take occupied cells.

Action Space (A):

the action space A consists of four possible actions:-

UP, Down, LEFT, Right

Que 1(b)

(1) calculate the number of state:

(I will consider here four room example)

- Total grid :- 11 X 11 => 121 cells
- number of free cells :- 104 cells (17 walls)

(2) Total action => 4.

(3) Total number of State-action pairs

= S X A

=> 104 X 4

= 416 pairs

(4) we will consider a deterministic movement. the agent move to the intended adjacent cells.

-> Possible outcome per action
- Intended move
- blocked by wall.

so there are 2 possible next state.

(5) Reward function

- The agent will receive a reward of +1 when reache a goal and 0 otherwise.
- given s and a, the reward r is calculated by resulting s'

(6) Total non zero entries =
= |S| x |A| x possible s' per (s, a)
= 104 x 4 x 2
= 832 rows

- So approximately 832 rows in conditional probability table P(s', r | s, a) have non-zero probability

code.txt

```
start grid from the given array
define action space A with movements: Up, Down, Left, Right
Define state space S as all positions (i, j) where grid[i][j] == 0
Set goal_state = (10, 10)
Set start_state = (0, 0)

For each state s in S:
    For each action a in A:
        If s == goal_state:
            s_prime = start_state
            r = 0
            probability = 1
            Output (s, a, s_prime, r, probability)
            Continue to next action
        Compute intended position s_intended based on action a:
            s_intended = (s_i + delta_i, s_j + delta_j)
        If s_intended is within grid bounds and grid[s_intended] == 0:
            s_prime = s_intended
        Else:
            s_prime = s  // Collision with wall or boundary
        If s_prime == goal_state:
            r = +1
        Else:
            r = 0
        probability = 1
        Output (s, a, s_prime, r, probability)
```

Quez:- Part (a):

$\gamma = 0.5$, $T = 0.5$ and $R_1 = -1$, $R_2 = 2$, $R_3 = 6$, $R_4 = 3$, $R_5 = 2$

discounted return:- $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-t-1} R_T$

Backward calculation

$G_5 = 0$ (no rewards are received after time 5)

$G_4 = R_5 + \gamma G_5 \Rightarrow 2 + 0.5(0) = 2$

$G_3 = R_4 + \gamma G_4 \Rightarrow 3 + 0.5(2) = 4$

$G_2 = R_3 + \gamma G_3 \Rightarrow 6 + 0.5(4) = 8$

$G_1 = R_2 + \gamma G_2 \Rightarrow 2 + 0.5(8) = 6$

$G_0 = R_1 + \gamma G_1 \Rightarrow -1 + 0.5(6) = 2$

overall

$G_0 = 2$, $G_1 = 6$, $G_2 = 8$, $G_3 = 4$, $G_4 = 2$, $G_5 = 0$

Part (b)

$\gamma = 0.9$, $R_1 = 2$ followed by an infinite sequency of 7

For infinite sequence where rewards are constant after first reward we can use this formula:-

$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \ldots G_t$

$= R_{t+1} + \gamma (R_{t+2} + \gamma R_{t+3} + \ldots)$

If all reward from R_2 onward are 7.

$G_1 = 7 + \gamma \cdot 7 + \gamma^2 \cdot 7 + \ldots G_1$

$G_1 = 7(1 + \gamma + \gamma^2 \ldots)$

$G_1 = 7 \cdot \frac{1}{1-\gamma}$

$$G_1 = 7 \times \frac{1}{1-0.9} = 70$$

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

$\boxed{G_1 = 70}$ and $\boxed{G_0 = 65}$

Que 3 (a)

Reward R_t = zero at all times except $R_T = -1$ upon failure at time T

Episodic task

here episode ends upon failure. The return at each time step t is

$$G_t = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}$$

all the rewards are zero except at failure

$$G_t = \gamma^{T-t-1} \times (-1)$$
$$= -\gamma^{T-t-1}$$

b) continuing task

here the process goes on indefinitely without terminal state. failure is treated as a transition back to state.

$$G_T = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

non zero Rewards $R_T = -1$, then process continue after T

$$G_t = -\gamma^{T-t-1} + \sum_{k=T-t}^{\infty} \gamma^k \times 0$$

$$G_t = -\gamma^{T-t-1}$$

→ difference:

- both episodic and continuing tasks give the same result at each time step t

$$G_t = -\gamma^{T-t-1}$$

~~other episode~~

- for the episodic task, the episods ends at failure and no further records are considered
- for continuing task, the agent continues operate after failure. It potentially occur more records or penalties

Que 3

→ Part (b)

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} \ldots + R_T$$

what is wrong?

- The record structure does not differentiate between quick and slow solution. with $\gamma = 1$ and only a record at the end, every successful path through the maze gets the same total record of +1, Regardless of steps taken
- the agent has no incentive to find the shortest path.
- all successful trajectories appear equally good to the agent.

how to improove:

1) Introduce discounting factor
2) add small negative record for each step
3) Also add time-dependent record.

Que 4: $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ → eq 3.8

Now we will add c to all rewards
The new rewards are $R'_{t+1} = R_{t+1} + c$

→ New G'_t using new rewards R'_{t+1} is

$$G'_t = (R_{t+1} + c) + \gamma(R_{t+2} + c) + \gamma^2(R_{t+3} + c) + \ldots$$

$$G'_t = (R_{t+1} + \underbrace{\gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots}_{G_t}) + (c + \gamma c + \gamma^2 c + \ldots)$$

$$G'_t = G_t + \underbrace{\sum_{k=0}^{\infty} \gamma^k c}_{\text{geometric}}$$

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

This shows that adding a constant c to all rewards shift the return by constant $\frac{c}{1-\gamma}$,

but the relative difference between return states are same.

→ Adding c constant to all rewards does not affect the value of any states

→ The value function for any state s under the modified reward system shifts by $v_c = \frac{c}{1-\gamma}$. The relative order of state value remains the same.

Q-4
(b) Episodic value function

$$V_\pi(s) = E_\pi\left[\sum_{t=0}^{T} \gamma^t R_{t+1} \mid S_0 = s\right]$$

Add c constant to each reward.

$$V'_\pi(s) = E_\pi\left[\sum_{t=0}^{T} \gamma^t (R_{t+1} + c) \mid S_0 = s\right]$$

We can write as

$$V'_\pi(s) = \underbrace{E_\pi\left[\sum_{t=0}^{T} \gamma^t R_{t+1} \mid S_0 = s\right]}_{\Rightarrow V_\pi(s)} + \sum_{t=0}^{T} \gamma^t c$$

$$\sum_{t=0}^{T} \gamma^t c = c \times \frac{1 - \gamma^{T+1}}{1-\gamma}$$

Thus modified function value is

$$V'_\pi(s) = V_\pi(s) + c \times \frac{1 - \gamma^{T+1}}{1-\gamma}$$

→ In episodic task adding a constant c to all rewards does affect the total value because the sum of $\sum_{t=0}^{T} \gamma^t$ is finite.
The agent decision will not change because a relative difference between state value, but rewards will change.

Example:-

If we add $c=-1$ to every reward the agent will follow the same policy to reach the goal, but the total rewards will decrease by $\frac{1-\gamma^{T+1}}{1-\gamma}$. Also, if we add $c=+1$, the total reward increase by same amount, even though the agent follow the same optimal path.

(a)

Que 5 State values:- center = 0.7, up = 2.3, Right = 0.4
Down = -0.4, Left = 0.7

Probability of each action = $\frac{1}{4}$

$\gamma = 0.9$

The bellman eq for equipobable random policy at center state $V_\pi(s)$ (eq 3.14.)

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

$$= \frac{1}{4}[\gamma \times 2.3] + \frac{1}{4}[\gamma \times 0.4] + \frac{1}{4}[\gamma - (-0.4)] + \frac{1}{4}[\gamma \times 0.7]$$

~~$\frac{1}{4}[0.9 \times 2.3 + 0.9 \times 0.4 + (-0.4) + 0.9 \times 0.7]$~~

$$= \frac{1}{4}[0.9 \times 2.3 + 0.9 \times 0.4 + 0.9 \times (-0.4) + 0.9 \times 0.7]$$

$$= 0.675 \approx 0.7$$

This aligns with the center value given

Q-5 Part b (optimal policy)

State value: center = 17.8, Up = 18, Right = 16.4
Down = 16, Left = 18.2

$\gamma = 0.9$

Bellman optimally equation for v^* at center state.

$$v^*(s) = \max_a \left[\sum_{s',r} p(s', r | s, a) [r + \gamma v^*(s')] \right] \ldots 3.19$$

Assuming transitions are deterministic under the optimal policy and there are no explicit rewards

$$V^*(s) = \max [\gamma \cdot V^*(s')] \text{ for each action } a)$$

$$V^*(s) = \max [0.9 \times 18, 0.9 \times 16.4, 0.9 \times 16, 0.9 \times 18.2]$$

$$V^*(s) = 16.38$$

The highest value we got using calculated optimal value is 16.38 (left) This does not align with the stated value 17.8

here I divides equation into two
to simplify

Que 6::

(a) Bellman equation for state values:-

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

$p(s'|s,a)$ ⇒ the probability of transitioning to state s' from state s after taking action a

$R(s,a,s')$:- Reward received after transitioning from s to s', having action a

→ States and action for high

states:- high: battery is high
low:- battery is low

Actions:-
In high $(A(high))$:- {search, wait}
In low $(A(low))$:- {search, wait, recharge}

→ Bellman for high state

$$V^\pi(high) = \sum_a \pi(a|high) \sum_{s'} \{p(s'|high,a)$$
$$[R(high,a,s') + \gamma V^\pi(s')]$$

expand equation:

1) Action:- Search

probability of taking action $\pi(\text{search}|\text{high})$

-> Transitions and rewards

stay in high:-

probability α

Reward r_{search}

Next state value = $V^{\pi}(\text{high})$

Transition to low

Prob = $1-\alpha$

Reward = r_{search}

Next state value:- $V^{\pi}(\text{low})$

Eq:-

$$\pi(\text{search}|\text{high})[\alpha(r_{search}+\gamma V^{\pi}(\text{high})) + (1-\alpha)(r_{search}+\gamma V^{\pi}(\text{low}))]$$

2) Action:- wait

Prob:- $\pi(\text{wait}|\text{high})$

-> Transitions and rewards

stays in high

Prob = 1

Reward:- r_{wait}

Next state value:- $V^{\pi}(\text{high})$

eq

$$\pi(\text{wait}|\text{high})[r_{wait}+\gamma V^{\pi}(\text{high})]$$

$V^{\pi}_{final}(high) = \pi(search|high)[\alpha(r_{search} + \gamma V^{\pi}(high)) + (1-\alpha)(r_{search} + \gamma V^{\pi}(low))] + \pi(wait|high)[r_{wait} + \gamma V^{\pi}(high)]$

→ **For state low**

$$V^{\pi}(low) = \sum_{a} \pi(a|low) \sum_{s'} P(s'|low, a)[R(low, a, s') + \gamma V^{\pi}(s')]$$

expand eq:

1) Action: search

- Probability of taking action :- $\pi(search|low)$

Transition and rewards:

- Stay in low:
  probability :- β
  Reward: r_{search}
  Next state :- $V^{\pi}(low)$.
- Battery depletes.
  prob :- $1-\beta$
  Reward :- -3 penalty
  Next state :- $V^{\pi}(high)$

eq:

$\pi(search|low)[\beta(r_{search} + \gamma V^{\pi}(low)) + (1-\beta)(-3 + \gamma V^{\pi}(high))]$

2) Action :- wait

Prob :- π(wait|low)
Transition & rewards,

- Stay in low :-
  Prob = 1
  Reward = r_{wait}
  Next state value :- V^{π}(low)

π(wait|low) [$r_{wait} + \gamma V^{\pi}$(low)]

3) Action - Recharge.

Proba :- π(recharge|low)

Transition and reward.
- Transition to high
  Prob :- 1
  Reward :- 0
  Next state value :- V^{π}(high)

π(recharge|low)[$0 + \gamma V^{\pi}$(high)]

final eq :-

V^{π}(low) :- π(search|low)[$\beta(r_{search} + \gamma V^{\pi}$(low)) + (1-$\beta$)(-3 + γV^{π}(high))]
+ π(wait|low)[$r_{wait} + \gamma V^{\pi}$(low)]
+ π(recharge|low)[$r V^{\pi}$(high)]

final equation

Que 6

→ Part (b)

given Parameters

$\gamma = 0.9$

- Transition probabilities

$\alpha = 0.8$

$\beta = 0.6$

- Reward

$r_{search} = 10$

$r_{wait} = 3$

Rescue reward $= -3$

- Policy

$\pi(search|high) = 1$

$\pi(wait|low) = 0.5$

$\pi(recharge|low) = 0.5$

$\pi(search|low) = 0$

→ Bellman equation for high.

→ will put into the above question find eq.

$V(high) = \alpha[r_{search} + \gamma V(high)] + (1-\alpha)[r_{search} + \gamma V(low)]$

↓

$0.2 V(high) - 0.1 \& V(low) = 10 \rightarrow c1)$

→ Bellman for low

$V(low) = \pi(wait|low)[r_{wait} + \gamma V(low)] + \pi(recharge|low)[0 + \gamma V(high)]$

$0.5 V(low) - 0.45 V(high) = 1.5 \rightarrow c2)$

→ Multiply eq c1) by 55 and eq c2) by 25

$15.4V(high) - 9.9V(low) = 550$
$15.4V(low) - 12.6V(high) = 642$

After solving will get

$V(high) \cong 79.04$
$V(low) \cong 67.40$

→ Check that it satisfies the bellman equation

$V(high) = 10 + 0.72 \times V(high) + 0.18 V(low))$
$= 79.04.$

$V(low) = 1.5 + 0.45 \times V(low) + 0.45 \times V(high)$
$= 69.38$

→ This shows that expected cumulative records for the robot starting in each state is under given policy

→ when battery is high, cumulative record is 79.04

→ when battery is low, cumulative record is 67.40

Extra credit

Ques (c)

⇒ bellman equation with θ

$V(high) = r_{search} + \gamma[\theta V(high) + (1-\theta) V(low)]$

$= 10 + 0.9[0.8 \cdot V(high) + 0.2 V(low)]$

$= 10 + 0.72 V(high) + 0.18 V(low)$

$\theta = 0.28 V(high) - 0.18 V(low) = 10 \rightarrow (1)$

$V(low) = \theta[r_{recharge} + \gamma V(low)] + (1-\theta)[\gamma V(high)]$

after simplify

$\Rightarrow V(low)(1 - 0.9\theta) = 3\theta + 0.9(1-\theta) V(high) \rightarrow (2)$

add $V(high)$ in above and then solve we get

$0.28 V(high) - 0.18\left(\frac{3\theta + 0.9(1-\theta) V(high)}{1 - 0.9\theta}\right) = 10$

⇒ To maximize $V(low)$ we consider $\theta = 0$ and $\theta = 1$

⇒ I) At $\theta = 0$

$IT(recharge | low) = 3$

$V(low) = 0 + \frac{0.9 \times 1 \times V(high)}{1 - 0}$

$= 0.9 V(high)$

Substitute $V(low)$ into (1)

$0.28V(high) - 0.18 \times 0.9V(high) = 10$

$V(high) = 84.75$

$V(low) = 76.28$

(2) At $\theta = 1$

$\Pi(wait|low) = 1$

Put in eq 2

$V(low) = \frac{3}{1 - 0.1} = 30$

→ Substitute $V(low)$ in eq (1)

$V(high) = 0.28V(high) = 15.4$

$V(high) \approx 55$

→ Overall:

$V(low)$ is higher when $\theta = 0$

→ optimal policy

$\theta = 0$; choose recharge in the low state

→ value function at $\theta = 0$

$V(high) = 84.75$

$V(low) = 76.28$

To maximize $V(low)$, I set $\theta = 0$. So that robot will choose always recharge in low state.