WIDS Project – RL Theory Week 3
Assignment 3

Arya Patil

Question 1: The "Cliff Walker"

1.1 Return G_0

The agent follows the trajectory:

$$S_1 \xrightarrow{R} S_2 \xrightarrow{L} S_1 \xrightarrow{R} S_2 \xrightarrow{R} S_{Term}$$

Rewards:

- Each non-terminal transition: -1
- Transition to terminal state: $+10$
- Discount factor: $\gamma = 0.9$

The return is defined as:

$$G_0 = R_1 + \gamma R_2 + \gamma^2 R_3 + \gamma^3 R_4$$

Substituting values:

$$\begin{aligned} G_0 &= (-1) + 0.9(-1) + 0.9^2(-1) + 0.9^3(10) \\ &= -1 - 0.9 - 0.81 + 7.29 \\ &= \boxed{4.58} \end{aligned}$$

1.2 Bellman Expectation Equation for $v_\pi(S_2)$

Under the random policy π, the agent chooses Left or Right with probability 0.5.
Transitions from S_2:

- Left $\to S_1$ with reward -1
- Right $\to S_{Term}$ with reward $+10$

Since $v_\pi(S_{Term}) = 0$, the Bellman equation is:

$$\boxed{v_\pi(S_2) = 0.5[-1 + 0.9 v_\pi(S_1)] + 0.5[10]}$$

Question 2: Philosophy of Reward

The agent learns to maximize the number of "dust sucked" detection events rather than keeping the room clean. A likely exploit is that the robot repeatedly redistributes or releases dust after sucking it up, allowing it to re-suck the same dust multiple times.

This is an example of **reward hacking**, where the agent optimizes the proxy reward instead of the true task objective.

Question 3: The Discount Factor

Part A: Mathematical Necessity

If rewards are always +1, the task is infinite-horizon, and $\gamma = 1$, then:

$$v_\pi(s) = \sum_{t=0}^{\infty} 1 = \infty$$

Thus, the value function diverges. A discount factor $\gamma < 1$ ensures convergence:

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

Part B: Intuition

- $\gamma = 0$: The agent only cares about immediate rewards and behaves impulsively.
- $\gamma = 0.99$: The agent values long-term outcomes and behaves strategically.

Question 4: The Brain Teaser

Originally, each step gives a reward of -1 with $\gamma = 1$, leading the agent to minimize the number of steps.

After adding a constant $C = +2$, each step gives a reward of $+1$. Since $\gamma = 1$, the agent can collect infinite reward by never reaching the goal.

Conclusion: The optimal policy changes. The agent avoids the goal indefinitely to maximize cumulative reward.

Question 5: Bellman Expectation Equation Derivation

By definition:

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

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

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} \mid S_t = s] + \gamma \mathbb{E}_\pi[G_{t+1} \mid S_t = s] \end{aligned}$$

Expanding over actions and transitions:

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

Question 6: Linear Algebra of RL

6.1 Matrix Form

The Bellman equation in vector form is:

$$v_\pi = R_\pi + \gamma P_\pi v_\pi$$

Rearranging:

$$(I - \gamma P_\pi) v_\pi = R_\pi$$

$$\boxed{v_\pi = (I - \gamma P_\pi)^{-1} R_\pi}$$

6.2 Computational Complexity

For Backgammon, $N \approx 10^{20}$ states.
Matrix inversion requires $O(N^3) \approx 10^{60}$ operations.
At 10^{18} FLOPs/sec:

$$\frac{10^{60}}{10^{18}} = 10^{42} \text{ seconds} \approx 3 \times 10^{34} \text{ years}$$

6.3 Conclusion

Exact dynamic programming is computationally infeasible, motivating approximate methods such as Monte Carlo learning.

Question 7: Model-Free Control

7.1 Greedy Policy using $v^*(s)$

$$\boxed{\pi'(s) = \arg\max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v^*(s')]}$$

7.2 Greedy Policy using $q^*(s, a)$

$$\boxed{\pi'(s) = \arg\max_a q^*(s, a)}$$

7.3 Comparison

In model-free environments, transition probabilities are unknown. Therefore, $v^*(s)$ alone is insufficient for action selection, whereas $q^*(s, a)$ directly provides the optimal action without requiring a model.