Online and Reinforcement Learning (2025)
Home Assignment 1

Your name and student ID

Contents

1 Short Questions

All four questions are True. Here is the explanation behind the answers:

1. In any finite MDP setting, every policy π has its own equivalent $\pi' \in \Pi^{SD}$ (which is optimal, even), which determines the action to take **in each state**. So, when a policy is fixed, the MDP reduces to a Markov Reward Process, where the transitions and rewards depend **only on the current state**.

2. Actually, quite similar explanation to the above one. There always exists an optimal static policy π, which means that it doesn't depend on the history.

3. A greedy policy with respect to the optimal Q^* selects actions that maximize the function Q, which by definition leads to optimal behavior in the MDP.

   This, in fact, follows from the Bellman optimality equations: being greedy w.r.t. Q^* yields the same value as V^*, meaning that no other policy can achieve higher returns.

4. The coverage assumption is crucial here for the convergence of wIS estimator. It states that π_b must cover all state-action pairs that π might visit. This ensures that π_b can generate all possible trajectories that π might take. Therefore, wIS estimator is a **consistent estimator of** V^* (as the number of samples increases, the estimator converges to the true value function with probability 1).

   In a more mathematical way:

   For every state-action pair (s,a) such that $\pi(a|s) > 0$, we must have $\pi_b(a|s) > 0$. The wIS estimator is defined as $\widehat{V}_{\text{wIS}} := \frac{\sum_{t=0}^{T} \rho_{0:t} G_t}{\sum_{t=0}^{T} \rho_{0:t}}$. Now we can use the law of large numbers:

$$\frac{1}{N} \sum_{i=1}^{N} \rho_{0:t}^{(i)} G_t^{(i)} \xrightarrow{\text{a.s.}} \mathbb{E}_{\pi_b}[\rho_{0:t} G_t] = V^\pi$$

$$\frac{1}{N} \sum_{i=1}^{N} \rho_{0:t}^{(i)} \xrightarrow{\text{a.s.}} \mathbb{E}_{\pi_b}[\rho_{0:t}] = 1$$

   So we got the wanted result: $\mathbb{E}_{\pi_b}[\widehat{V}_{\text{wIS}}] = V^\pi$.

2 MDPs with Similar Parameters Have Similar Values

As the two given expressions are bounded by the same value, they seem to be equivalent. So, for $\pi \in \Pi^{SD}$ it holds that $a = \pi(s)$ and therefore $Q^{\pi}(s,a) = Q^{\pi}(s, \pi(s)) = V^{\pi}(s)$.

Now, we start with writing out the left side of the expression (ii) using the Bellman equation:

$$V_1^{\pi}(s) - V_2^{\pi}(s) = [r_1(s,\pi(s)) - r_2(s,\pi(s))] + \gamma \sum_{x \in S} P_1[x|s,\pi(s)]V_1^{\pi}(x) - P_2[x|s,\pi(s)]V_2^{\pi}(x) \tag{1}$$

We can simplify the notation with $r_i(s,\pi(s)) = r_i$ and $P_i[x|s,\pi(s)]V_i^{\pi}(x) = P_iV_i$:

$$\begin{aligned} V_1 - V_2 &= [r_1 - r_2] + \gamma \sum P_1V_1 - P_2V_2 & (2) \\ &= [r_1 - r_2] + \gamma \sum P_1V_1 - P_1V_2 - P_2V_2 + P_1V_2 & (3) \end{aligned}$$

If we mix the components in the sum, and then use $|r_1 - r_2| \leq \alpha$ and $||Py - P2||_2 \leq \beta$, we get:

$$|V_1 - V_2| \leq \alpha + \gamma |\sum P_1(V_1 - V_2) + V_2\beta| \tag{4}$$

Now we can apply the triangle inequality $|\sum a| \leq \sum |a|$:

$$|V_1 - V_2| \leq \alpha + \gamma \sum P_1|(V_1 - V_2)| + V_2\beta. \tag{5}$$

Next up, two ingredients:

First, we can use the rough bound $V_i \leq \frac{R_{max}}{1-\gamma}$.

Secondly, we would like to get the $|V_1 - V_2|$ expression out of the sum, so we can simplify further with $\sum P_1 = 1$.
Since $V_1^{\pi}(s) - V_2^{\pi}(s) \leq |V_1^{\pi}(s) - V_2^{\pi}(s)|$ holds for all $s \in S$, it definitely holds for the s' for which the value function is the greatest - we can bound difference $V_1 - V_2$ with the maximum such difference, reached in state s'.
We can call this $\max_{s' \in S} |V_1^{\pi}(s') - V_2^{\pi}(s')|$. So it has to hold that $|V_1^{\pi}(s) - V_2^{\pi}(s)| \leq \max_{s' \in S} |V_1^{\pi}(s') - V_2^{\pi}(s')|$.

Now, since this new expression doesn't depend on $x \in S$ (from the sum), we can say:

$$|V_1 - V_2| \leq \alpha + \gamma |(V_1 - V_2)| + \gamma \frac{R_{max}\beta}{1-\gamma}. \tag{6}$$

What's left is to solve for $|V_1 - V_2|$:

$$|V_1 - V_2| - \gamma|(V_1 - V_2)| \leq \alpha + \frac{\gamma R_{max}\beta}{1-\gamma} \tag{7}$$

$$|V_1 - V_2| \leq \frac{\alpha + \frac{\gamma R_{max}\beta}{1-\gamma}}{1-\gamma} \tag{8}$$

$$|V_1 - V_2| \leq \frac{\alpha(1-\gamma) + \gamma R_{max}\beta}{(1-\gamma)^2} \tag{9}$$

$$|V_1 - V_2| \leq \frac{\alpha + \gamma R_{max}\beta}{(1-\gamma)^2} \tag{10}$$

NB: In the step between (9) and (10) we can use the fact that since $\gamma, (1-\gamma) \in [0,1]$, $\alpha(1-\gamma) \leq \alpha$.

Therefore, it holds that $|V_1^{\pi}(s) - V_2^{\pi}(s)| \leq \frac{\alpha+\gamma R_{max}\beta}{(1-\gamma)^2}$ ■

3 Policy Evaluation in RiverSwim

Code for this task can be found in the 3.*py* file. Here I will paste the most important code snippet:

```
...
for s in range(nS):

total = 0.0
for _ in range(n):

    env.reset()
    env.s = s  # set the starting state
    current_s = s

    discounted_sum = 0.0

    for t in range(T):
        # defining our policy as per HA2.pdf
        if current_s in [0, 1, 2]:
            action = np.random.choice([0, 1], p=[0.35, 0.65])
        else:
            action = 1 # when s = 4 or 5

        # deciding the reward and the next state s_t+1
        next_s, reward = env.step(action)


        discounted_sum += (gamma ** t) * reward
        current_s = next_s

    total += discounted_sum

V_hat[s] = total / n

return V_hat
```

3.1 Monte Carlo Simulation

Sequence of states = [0, 1, 2, 2, 2, 3, 3, 3, 3, 3, 2]
Monte Carlo Approximation of V^{π}: [4.3199, 4.875, 6.7515, 10.626, 11.0751]

3.2 Exact value

In the code where I use `np.linalg.solve` to calculate the exact matrix. Also worth noting is that

Exact Value Function V^{π}: [4.1209, 4.7112, 6.3346, 9.738, 11.1778]

It seems that the Monte Carlo approximation of the value function V^{π} is quite accurate, as it resembles the exact vector quite a lot.

4 Solving a Discounted Grid-World

-

5 Off-Policy Evaluation in Episode-Based River Swim

TBD by February 26th.