ECE433/COS435 Introduction to RL
Assignment 3: Value Iteration and Policy Gradient
Spring 2025

Fill me in

Kaixuan Huang

Due February 24, 2025

Collaborators

Fill me in

Please fill in the names and NetIDs of your collaborators in this section.

Instructions

Writeups should be typesetted in Latex and submitted as PDF. You can work with whatever tool you like for the code, but **please submit the asked-for snippet and answer in the solutions box as part of your writeup. We will only be grading your writeup.**

Grading. Questions 1-2 will collectively be worth total 50 points, and the coding assignment will also be worth 50 points, making the total score 100 points.

Question 0 (0 points). Feedback

How many hours did you spend on this homework?

Solution

Your solution here...

Question 1 (20 points). Finite-state Value Function

An agent is navigating a very simple environment structured as a straight path with three states labeled s_1, s_2, and s_3, where state s_3 is a terminal state. At each step, the agent can choose to move to the adjacent states or stay in the current state. Assume the discount factor $\gamma = 0.5$. The actions available in each state are:

- In state s_1: the agent can either "move" to state s_2 or "stay" in state s_1.
- In state s_2: the agent can "move" to state s_3, "stay" in state s_2, or "move back" to state s_1.

The rewards are as follows:

- Moving from state s_1 to state s_2 gives a reward of -1.
- Moving from state s_2 to state s_3 gives a reward of 0.
- Moving back from state s_2 to state s_1 gives a reward of -2.
- Staying in states s_1 or state s_2 gives a reward of -1.

Question 1.a (10 points)

Calculate the value function for each state when the agent always decides to "move" to the next state when possible (i.e., from s_1 to s_2, and from s_2 to s_3). (Note: The reward and value function at the terminal state s_3 is zero.)

Solution

Since s_3 is the terminal state, we have $V(s_3) = 0$.
When starting from state s_2, the agent will choose to move to state s_3 and receive an immediate reward 0. Therefore,

$$V(s_2) = 0 + \gamma V(s_3) = 0.$$

When starting from state s_1, the agent will choose to move to state s_2 and receive an immediate reward -1. Therefore,

$$V(s_1) = -1 + \gamma V(s_2) = -1.$$

In summary, the value function of the policy is

$$V(s_1) = -1, \quad V(s_2) = 0, \quad V(s_3) = 0.$$

Question 1.b (10 points)

Calculate the value function at each state when the agent always chooses to move to state s_2 when in state s_1, and always chooses to move back to state s_1 when in state s_2. (Note: The reward and value function at the terminal state s_3 is zero.)

Solution

Since s_3 is the terminal state, we have $V(s_3) = 0$.
By Bellman Equation, we have

$$V(s_1) = -1 + \gamma V(s_2)$$
$$V(s_2) = -2 + \gamma V(s_1).$$

Substituting $\gamma = 0.5$ and solving the linear equations, we obtain

$$V(s_1) = -\frac{8}{3}, \quad V(s_2) = -\frac{10}{3}.$$

In summary, the value function of the policy is

$$V(s_1) = -\frac{8}{3}, \quad V(s_2) = -\frac{10}{3} \quad V(s_3) = 0.$$

Question 2 (30 points). Properties of value functions

In this question, we consider an infinite horizon MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, p, \gamma)$. We aim to derive some interesting theoretical properties for its value function. Recall that a value function V is the optimal value function if and only if it satisfies the Bellman Optimal Equation:

$$V(s) = \max_a \left[r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)} V(s') \right], \quad \text{for all } (s,a).$$

Question 2.a (10 points)

Show that a policy π is optimal if and only if its corresponding value functions $V^\pi : \mathcal{S} \to \mathbb{R}$ and $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ satisfies $V^\pi(s) \geq Q^\pi(s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.

Solution

For one direction, we assume that π is optimal. This indicates that V^π satisfies the following Bellman Optimality Equation

$$V^\pi(s) = \max_a \left[r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)} V^\pi(s') \right].$$

Therefore,

$$\begin{aligned} V^\pi(s) &\geq r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)} V^\pi(s') \\ &= Q^\pi(s,a) \end{aligned}$$

for all a.
For the other direction, we assume that $V^\pi(s) \geq Q^\pi(s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. As

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)}[Q^\pi(s,a)],$$

we have $V^\pi(s) = Q^\pi(s,a)$ for all the actions a that satisfies $\pi(a|s) > 0$. This means

$$\begin{aligned} V^\pi(s) &= \max_a Q^\pi(s,a) \\ &= \max_a \left[r(s,a) + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a)} V^\pi(s') \right]. \qquad \text{(Bellman Equation for policy } \pi) \end{aligned}$$

Therefore, the value function of policy π satisfies Bellman Optimality Equation, which indicates that π is optimal.

Question 2.b (10 points)

Is the optimal policy of an MDP unique? Please answer by proof or a counter-example.

Solution

Not necessary. Here is a naive counter-example. Let's consider a two-state MDP with zero rewards for all transitions. For this case, any policy π will have an all-zero value function. Therefore, all policies are simultaneously optimal.

Question 2.c (5 points)

Suppose that $\mathcal{M}$ has no terminating state. The agent will work forever. Now someone decides to add a small reward bonus c to all transitions in the MDP, which results in a new reward $r'(s,a) = r(s,a) + c$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. Note that r is the original reward function. Could this change the optimal policy of $\mathcal{M}$?

Solution

No. As we have assumed that the agent will work forever, adding a constant reward bonus c to all transitions will increase the value function by a certain amount for **all** policies. It won't affect the relative relationship between any two policies, so the optimal policy of the original MDP will be also optimal in the modified MDP.
Specifically, for any policy π, by definition

$$V^\pi(s) = \mathbb{E}[r(s_0,a_0) + \gamma r(s_1,a_1) + \cdots + \gamma^t r(s_t,a_t) + \cdots \mid s_0 = s, a_t \sim \pi(\cdot|s_t)].$$

For the modified MDP $\mathcal{M}'$, we have

$$\begin{aligned} V'^{\pi}(s) &= \mathbb{E}[r'(s_0,a_0) + \gamma r'(s_1,a_1) + \cdots + \gamma^t r'(s_t,a_t) + \cdots \mid s_0 = s, a_t \sim \pi(\cdot|s_t)] \\ &= \mathbb{E}[r(s_0,a_0) + \gamma r(s_1,a_1) + \cdots + \gamma^t r(s_t,a_t) + \cdots \mid s_0 = s, a_t \sim \pi(\cdot|s_t)] \\ &\quad + \left(c + \gamma c + \gamma^2 c + \cdots\right) \\ &= V^{\pi}(s) + \frac{c}{1-\gamma}. \end{aligned}$$

Therefore, we see that for any π, $V'^{\pi} = V^{\pi} + \frac{c}{1-\gamma}$.

Question 2.d (5 points)

If $\mathcal{M}$ is allowed to have terminating states, does the answer in Question 2.c still hold? If not, provide an example MDP where your answers to Question 2.c and this one differs.

Solution

No. We can use the MDP in Question 1 as the counter-example. For the original MDP, as all the rewards are non-positive, the optimal policy is to reach the terminating state s_3 as soon as possible (the policy in Question 1.a). However, if we add a constant $c = 3$ for all (s, a) pairs to make all the rewards positive, the optimal policy will be change: when the agent is at s_2, staying at s_2 forever will give larger rewards than going to the terminal state s_3.

Question 3 (50 points). Coding Problems

3.1 Value Iterations (20 points)

a. Bellman Update (15 points)

Paste the code block implementing `Bellman Update` below.

Solution

```
    ############################
    # YOUR IMPLEMENTATION HERE #

    # step 1: compute the Q function:
    # Q(s,a) = E_{s'} [r(s,a, s') + gamma V(s')]

    Q_func = np.zeros((nS, nA))
    for current_state in range(0, nS):
      for current_action in range(0, nA):
        list_transition = P[current_state][current_action]
```

```
        for (prob, next_state, reward, is_terminal) in
    list_transition:
            total_reward = reward
            if not is_terminal:
              total_reward += gamma * value[next_state]
            Q_func[current_state, current_action] += prob *
    total_reward

     # step 2: take argmax from Q
     new_value = np.max(Q_func, axis=1)
     greedy_policy = np.argmax(Q_func, axis=1)

     ############################
```

b. Value Iteration (5 points)

Paste the code block implementing `Value Iteration` below.

Solution

```
     ############################
     # YOUR IMPLEMENTATION HERE #

     previous_value = None
     while (previous_value is None or np.max(np.abs(previous_value -
    value_function))>tol):
          previous_value = value_function
          value_function, policy = Bellman_Update(P, nS, nA,
    value_function, gamma)
     ############################
```

c. Results and Discussion (0 points)

Take a look at the document of the RL environment. What is the theoretical upper bound of the un-discounted cumulative reward for this environment? How close are the rewards of your learned policy to the upper bound?

Solution

Your solution here...

3.2 REINFORCE

a. Discounted Reward (15 points)

Please paste the code block implementing `discounted reward` below.

Solution

```
    ############################
    # YOUR IMPLEMENTATION HERE #
    R = 0
    for r in episode_rewards[::-1]:
        R = r + gamma * R
        discounted_rewards.insert(0, R)
    ############################

```

b. REINFORCE (15 points)

Please paste the code block implementing `REINFORCE` below.

Solution

```
    ### Q4: REINFORCE, 15 points ###
    ############################
    # YOUR IMPLEMENTATION HERE #
    policy_loss = torch.stack(log_probs) * discounted_rewards
    policy_loss = -policy_loss.sum()
    ############################

```

c. Results and Discussion (0 points)

Take a look at the document of the RL environment. What is the theoretical upper bound of the un-discounted cumulative reward for this environment? How close are the rewards of your learned policy to the upper bound?

Solution

Your solution here...

Question 4 (0 points). Bellman residual [Optional]

This problem **will not be graded**, but we encourage those who are interested in the theoretical aspect of reinforcement learning to try it out. We will not include questions like this in formal exams. In the lecture, we introduced the (optimal) Bellman operator for an infinite horizon MDP with discount factor γ and transition p:

$$(\mathbb{B}V)(s) = \max_{a \in \mathcal{A}} \left\{ r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) V(s') \right\},$$

and the Bellman operator with respect to a certain policy π:

$$(\mathbb{B}^\pi V)(s) = \sum_{a\in\mathcal{A}} r(s,a)\pi(a|s) + \gamma \sum_{s'\in\mathcal{S},a\in\mathcal{A}} p(s'|s,a)\pi(a|s)V(s').$$

We denote the optimal policy by π^* and the optimal value function by V^*. As we know from the lecture, learning V^* is equivalent to solving the following Bellman equation:

$$V(s) - \mathbb{B}V(s) = 0, \forall s \in \mathcal{S}.$$

For an arbitrary function $V : \mathcal{S} \to \mathbb{R}$, define the Bellman residual to be $(\mathbb{B}V - V)$, and its magnitude by $\|\mathbb{B}V - V\|_\infty$. Recall that for a vector $x = (x_i)_{i\in[d]}$, $\|\cdot\|_\infty$ is defined by $\max_{i\in[d]} |x_i|$. As we will see through the course, this Bellman residual is an important component of several important RL algorithms such as the Deep Q-network.

Question 4.a

Prove the following statements for an arbitrary $V : \mathcal{S} \to \mathbb{R}$ (not necessarily a value function for any policy):

$$\|V - V^\pi\|_\infty \le \frac{\|V - \mathbb{B}^\pi V\|_\infty}{1-\gamma}, \text{ for any policy } \pi$$
$$\|V - V^*\|_\infty \le \frac{\|V - \mathbb{B}V\|_\infty}{1-\gamma}.$$

(Hint: use Bellman equation to expand V^π, then apply triangle inequality.)

Solution

By Bellman (Consistency) Equation, we know that for any policy π, we have $V^\pi = \mathbb{B}^\pi V^\pi$. Therefore, we have

$$\begin{aligned}|V(s) - V^\pi(s)| &= |V(s) - \mathbb{B}^\pi V^\pi(s)| \\ &\le |V(s) - \mathbb{B}^\pi V(s)| + |\mathbb{B}^\pi V(s) - \mathbb{B}^\pi V^\pi(s)| \\ &\le |V(s) - \mathbb{B}^\pi V(s)| + \gamma |V(s) - V^\pi(s)|,\end{aligned}$$

where the last inequality follows from the contractiveness of the Bellman Operator. Therefore, we have

$$(1-\gamma)|V(s)-V^\pi(s)| \le |V(s)-\mathbb{B}^\pi V(s)| \implies |V(s)-V^\pi(s)| \le \frac{1}{(1-\gamma)}|V(s)-\mathbb{B}^\pi V(s)|.$$

Hence we have proved $\|V - V^\pi\|_\infty \le \frac{\|V-\mathbb{B}^\pi V\|_\infty}{1-\gamma}$.
By Bellman Optimality Equation, we know that $V^* = \mathbb{B}V^*$. Therefore, we have

$$\begin{aligned}|V(s) - V^*(s)| &= |V(s) - \mathbb{B}V^*(s)| \\ &\le |V(s) - \mathbb{B}V(s)| + |\mathbb{B}V(s) - \mathbb{B}V^*(s)| \\ &\le |V(s) - \mathbb{B}V(s)| + \gamma |V(s) - V^*(s)|,\end{aligned}$$

where the last inequality follows from the contractiveness of the Bellman Operator. Therefore, we have

$$(1-\gamma)|V(s)-V^*(s)| \le |V(s)-\mathbb{B}V(s)| \implies |V(s)-V^*(s)| \le \frac{1}{(1-\gamma)}|V(s)-\mathbb{B}V(s)|.$$

Hence we have proved $\|V-V^*\|_\infty \le \frac{\|V-\mathbb{B}V\|_\infty}{1-\gamma}$.

Question 4.b

Now let's assume that π is the greedy policy extracted from V, and assume $\|V-\mathbb{B}V\|_\infty \le \epsilon$. Prove the following inequality by utilizing the results in Question 3.a:

$$V^* - V^\pi \le \frac{2\epsilon}{1-\gamma}.$$

This shows that as long as the Bellman residual of V is small, then the policy learned from V will not be too bad.

Solution

As π is the greedy policy extracted from V, one can verify that $\mathbb{B}^\pi V = \mathbb{B}V$:

$$\begin{aligned}
\mathbb{B}^\pi V(s) &= \mathbb{E}_{a\sim\pi(\cdot|s)}\left[r(s,a) + \mathbb{E}_{s'\sim p(\cdot|s,a)V(s')}\right] \\
&= \max_a \left[r(s,a) + \mathbb{E}_{s'\sim p(\cdot|s,a)V(s')}\right] && (\pi \text{ is greedy policy}) \\
&= \mathbb{B}V(s).
\end{aligned}$$

Therefore, the sub-optimality of V^π can be upper bounded by $\frac{2}{1-\gamma}$ times the Bellman error:

$$\begin{aligned}
\|V^* - V^\pi\|_\infty &\le \|V^* - V\|_\infty + \|V - V^\pi\|_\infty \\
&\le \frac{\|V-\mathbb{B}V\|_\infty}{1-\gamma} + \frac{\|V-\mathbb{B}^\pi V\|_\infty}{1-\gamma} \\
&= \frac{2\|V-\mathbb{B}V\|_\infty}{1-\gamma} \le \frac{2\epsilon}{1-\gamma}.
\end{aligned}$$