AI 3000 : Reinforcement Learning

Assignment № 2

Due Date : 09/10/2025

Easwar Subramanian, IIT Hyderabad 12/09/2025

Problem 1 : Importance Sampling

Consider a single state MDP with finite action space, such that $|\mathcal{A}| = K$. Assume the discount factor of the MDP γ and the horizon length to be 1. For taking an action $a \in \mathcal{A}$, let $\mathcal{R}^a(r)$ denote the unknown distribution of reward r, bounded in the range $[0, 1]$. Suppose we have collected a dataset consisting of action-reward pairs $\{(a, r)\}$ by sampling $a \sim \pi_b$, where π_b is a stochastic behaviour policy and $r \sim \mathcal{R}^a$. Using this dateset, we now wish to estimate $V^\pi = \mathbb{E}_\pi[r|a \sim \pi]$ for some target policy π. We assume that π is fully supported on π_b.

(a) Suppose the dataset consists of a single sample (a, r). Estimate V^π using importance sampling (IS). Is the obtained IS estimate of V^π is unbiased ? Explain. (2 Points)
The unbiased IS estimate of V^π is given by $\rho\, r$ where $\rho = \frac{\pi(a|s)}{\pi_b(a|s)}$. One can argue that the estimate is unbiased in the following way.

$$V^\pi(s) = \mathbb{E}_{a\sim\pi}(r) = \mathbb{E}_{a\sim\pi_b}\left(\frac{\pi(a|s)}{\pi_b(a|s)} r\right)$$

The entity $\rho\, r$ is sample estimate of the expecation in RHS

(b) Compute

$$\mathbb{E}_{\pi_b}\left[\frac{\pi(a|\cdot)}{\pi_b(a|\cdot)}\right]$$

(1 Point)

$$\mathbb{E}_{a\sim\pi_b}\left[\frac{\pi(a|\cdot)}{\pi_b(a|\cdot)}\right] = \sum_{a\in\mathcal{A}}\left[\frac{\pi(a|\cdot)}{\pi_b(a|\cdot)}\pi_b(a|\cdot)\right] = 1$$

(c) For the case that π_b is a uniformly random policy (all K actions are equiprobable) and π a deterministic policy, provide an expression for importance sampling ratio. (1 Point)

$$\rho = \frac{1_{a=\pi(s)}}{1/K}$$

(d) We consider a general multi-state (i.e $|\mathcal{S}| > 1$), multi-step MDP. We futher assume that $\mu(s_0)$ to be the initial start state distribution (i.e. $s_0 \sim \mu(s_0)$) where s_0 is the start state of the MDP.

Let τ denote a trajectory (state-action sequence) given by, $(s_0, a_0, s_1, a_1, \cdots, s_t, a_t, \cdots)$ with actions $a_{0:\infty} \sim \pi_b$. Let Q and P be joint distributions, over the entire trajectory τ induced by the behaviour policy π_b and a target policy π, respectively. Provide a compact expression for the importance sampling weight $\frac{P(\tau)}{Q(\tau)}$. (2 Points)

Let $\tau \sim \pi_\theta$ denote the state-action sequence given by $s_0, a_0, s_1, a_1, \cdots, s_t, a_t, \cdots$. Then, $P(\tau;\theta)$ be the probability of finding a trajectory τ with policy π

$$P(\tau;\pi) = P(s_0)\prod_{t=0}^{\infty}\pi(a_t|s_t)P(s_{t+1}|s_t,a_t)$$

$$\frac{\mathbf{P}(\tau|\pi)}{\mathbf{Q}(\tau|\pi_b)} = \frac{\mu(s_0)\prod_{t=0}^{\infty}P(s_{t+1}|s_t,a_t)\pi(a_t|s_t)}{\mu(s_0)\prod_{t=0}^{\infty}P(s_{t+1}|s_t,a_t)\pi_b(a_t|s_t)} = \prod_{t=0}^{\infty}\frac{\pi(a_t|s_t)}{\pi_b(a_t|s_t)}$$

The point is that the dynamics and start state distribution gets cancelled as they don't depend on policy.

Problem 2 : Q Learning

Consider the two dimensional grid world problem given below. The terminal states are shaded and reaching them completes a trajectory and provides the reward as shown in those states. As usual, an agent can move in all four directions, namely, $\{\text{Left},\text{Right},\text{Up},\text{Down}\}$. Actions that the take the agent off the grid leaves the state unchanged. All moves are realized as there is no stochasticity in the environment.

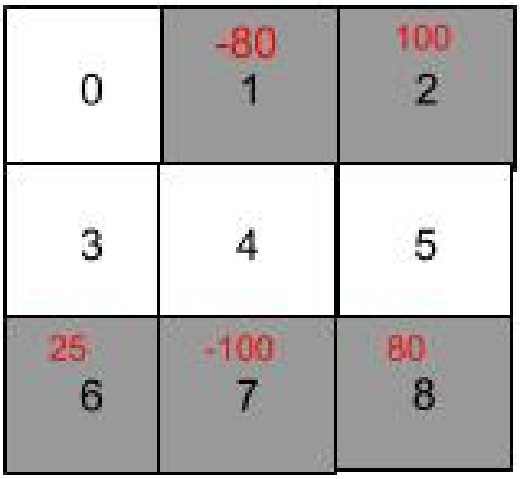

Figure 1: Two dimensional grid world

(a) What is the optimal V^* for each non-terminal state in the grid world ? (1 Point)

(b) An agent starts from the top left corner (square 0) and makes move following some policy π and the table below contains the 3 episodes generated by agent. Each line in an episode is a quadruple containing $(s_t, a_t, r_{t+1}, s_{t+1})$.

Using the Watkin's Q-learning update rule what are the Q-values after running through the three episodes for the following state-action pairs. (2 Points)

Episode 1	Episode 2	Episode 3
$(0, \text{Down}, 3, 0)$	$(0, \text{Down}, 3, 0)$	$(0, \text{Down}, 3, 0)$
$(3, \text{Right}, 4, 0)$	$(3, \text{Right}, 4, 0)$	$(3, \text{Right}, 4, 0)$
$(4, \text{Down}, 7, -100)$	$(4, \text{Right}, 5, 0)$	$(4, \text{Right}, 5, 0)$
	$(5, \text{Up}, 2, 100)$	$(5, \text{Down}, 8, 80)$

Table 1: Episodes generated by agent

- $Q(5, \text{Up})$
- $Q(3, \text{Down})$
- $Q(4, \text{Right})$.

For this question, assume the Q-table is initialized to 0, discount factor $\gamma = 0.5$, learning rate $\alpha_t = 0.5$ is held constant and the update rule is given as,

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \left(r_{t+1} + \gamma \max_{a'} Q(s', a') - Q(s_t, a_t) \right).$$

- $V^*(\cdot) = 100$ for all non-terminal states if $\gamma = 1$
  - $Q(5, \text{Up}) = 50$
  - $Q(3, \text{Down}) = 0$
  - $Q(4, \text{Right}) = 12.5$.

(c) For the Q-Learning algorithm to converge to the optimal Q function, a necessary condition is that the learning rate, α_t, which is the learning rate at the t-th time step would need to satisfy the Robinns-Monroe condtion. In here, the time step t refers to the t-th time we are updating the value of the Q value of the state-action pair (s, a). Would the following values for learning rate α_t obey Robbins Monroe conditions ? (3 Points)

(i) $\alpha_t = \frac{1}{t}$

(ii) $\alpha_t = \frac{1}{t^2}$

The series $\sum_{i=1}^{\infty} \frac{1}{t}$ is harmonic series and it does not converge. In fact, one can rewrite the series in the following way (by re-grouping terms)

$$\begin{aligned} \sum_{i=1}^{\infty} \frac{1}{t} &= 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots \\ &> 1 + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} \cdots \\ &= 1 + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \cdots = \infty \end{aligned}$$

A generalization of the Harmonic series is the p-series (Hyperharmonic series) defined as $\sum_{i=1}^{\infty} \frac{1}{t^p}$ for any +ve real number p. The p-series converges for all $p > 1$ (overharmonic

series) and diverges for all $p \leq 1$. So, one can now use the above property to get the following results.

α_t	$\sum \alpha_t$	$\sum \alpha_t^2$	Algo converges
$\frac{1}{t^2}$	$< \infty$	$< \infty$	No
$\frac{1}{t}$	∞	$< \infty$	Yes
$\frac{1}{t^{\frac{2}{3}}}$	∞	$< \infty$	Yes
$\frac{1}{t^{0.5}}$	∞	∞	No

Problem 3 : Game of Tic-Tac-Toe

Consider a 3×3 Tic-Tac-Toe game. The aim of this problem is to implement a Tic-Tac-Toe agent using Q-learning. This is a two player game in which the opponent is part of the environment.

(a) Develop a Tic-Tac-Toe environment with the following methods. (4 Points)

(1) An **init** method that starts with an empty board position, assigns both player symbols ('X' or 'O') and determines who starts the game. For simplicity, you may assume that the agent always plays 'X' and the opponent plays 'O'.

(2) An **act** method that takes as input a move suggested by the agent. This method should check if the move is valid and place the 'X' in the appropriate board position.

(3) A **print** method that prints the current board position

(4) You are free add other methods inside the environment as you deem fit.

(b) Develop two opponents for the Q-learning agent to train against, namely, a random agent and safe agent (4 Points)

(1) A **random agent** picks a square among all available empty squares in a (uniform) random fashion

(2) A **safe agent** uses the following heuristic to choose a square. If there is a winning move for the safe agent, then the corresponding square is picked. Else, if there is a blocking move, the corresponding square is chosen. A blocking move obstructs an opponent from winning in his very next chance. If there are no winning or blocking moves, the safe agent behaves like the random agent.

(c) The Q-learning agent now has the task to learn to play Tic-Tac-Toe by playing several games against **safe** and **random** opponents. The training will be done using tabular Q-learning by playing 10,000 games. In each of these 10,000 games, a fair coin toss determines who makes the first move. After every 200 games, assess the efficacy of the learning by playing 100 games with the opponent using the full greedy policy with respect to the current Q-table. Record the number of wins in those 100 games. This way, one can study the progress of the training as a function of training epochs. Plot the training progress graph as suggested. In addition, after the training is completed (that is after 10,000 games of training is done),

the trained agent's performance is ascertained by playing 1000 games with opponents and recording the total number of wins, draws and losses in those 1000 games. The training and testing process is described below. (10 Points)

(1) Training is done only against the random player. But the learnt Q-table is tested against both random and safe player.

(2) Training is done only against the safe player. But the learnt Q-table is tested against both random and safe player.

(3) In every game of training, we randomly select our opponent. The learnt Q-table is tested against both random and safe player.

(4) Among the three agents developed, which agent is best ? Why ?

(5) Is the Q-learning agent developed unbeatable against any possible opponent ? If not, suggest ways to improve the training process.

[**Note** : A useful diagnostic could be to keep count of how many times each state-action pair is visited and the latest Q value for each state-action pair. The idea is that, if a state-action pair is visited more number of times, Q value for that state-action pair gets updated frequently and consequently it may be more close to the 'optimal' value. Although, it is not necessary to use the concept of **afterstate**, it may be useful to accelerate the training process]

ALL THE BEST