AI 3000 / CS 5500 : Reinforcement Learning Review № 1

Date : 01/09/2022

Course Instructor : Easwar Subramanian 21/08/2022

Problem 1 : Markov Reward Process

A Markov chain with state space $\mathcal{S} = \{1, 2, 3\}$ has the transition probability given by

$$\mathcal{P} = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \end{pmatrix}.$$

Is there a terminal state for the above Markov chain ? If so, what is the terminal state and why is it a terminal state ? What are the expected times to reach the terminal state starting from any other state of the Markov process ?

State 3 is clearly an absorbing state as any transition from state 3 would end up in state 3 itself. There are no other absorbing states.

One approach is to model a suitable Markov reward process (MRP) with the $\mathcal{P}$ as given as above and $R = [1, 1, 0]^T$. The Bellman evaluation equation for an MRP is given by $V = (I - \gamma\mathcal{P})^{-1}\mathcal{R}$ which when solved for $V(s)$ would give the "expected number" of times required to reach state 3 from any other state of the MRP. The matrix $(I - \gamma P)$ becomes invertible if we set $V(s) = 0$ for state 3 even when we set $\gamma = 1$. One may find the inverse of the matrix $(I - \gamma\mathcal{P}_{2\times 2})$ and multiply with $\mathcal{R}_{2\times 1}$ to compute the expected time taken from other states of the MRP to reach state 3. The expected time to reach state 3 from state 1 and 2 are 2.5 and 2 respectively.

.

Problem 2 : Modelling of Markov Decision Process

Consider a *SUBWAY* outlet in your locality. Customers arrive to the store at times governed by an unknown probablity distribution. The outlet sells sandwiches with a certain type of bread (choice of 4 types) and filling (choice of 5 types). If a customer cannot get the desired sandwich, he/she is not going to visit the store again. Ingredients need to be discarded every 3 days after purchase. The store owner wants to figure out a policy for buying ingredients in such a way to maximize his long-time profit using reiforcement learning. To this end, we will formulate the problem as a MDP. You are free to make other assumptions regarding the probem setting. Please enumerate your assumptions while answering the questions below.

(a) Suggest a suitable state and action space for the MDP.

(b) Devise an appropriate reward function for the MDP

(c) Would you use discounted or undiscounted setting in your MDP formulation ? Justify your answer.

(d) Would you use dynamic progamming or reinforcement learning to solve the the problem ? Explain with reasons.

(e) Between MC and TD methods, which would you use for learning ? Why ?

(f) Is function approximation required to solve this problem ? Why or why not ?

We will leave out the first two questions as they are design choices. There is not one right answer. For example, the state space could be 4 variable to denote quantity of 4 types of bread and 5 other variables to denote volume of 5 types of filling. The reward function could indicate profit (or revenue) maximization. Exact details of the formula are needed. Depending on the choice of state, action and reward design, one can answer other questios as follows.

(c) Since the store owner is interested in maximizing long-time profit, the horizon is infinite horizon (or long finite time). So, it is good to use discounted setting

(d) DP is used in model based setting. It is difficult to get transition and reward function formulated accurately. Hence, model free RL approaches are to be used

(e) Since the time horizon is long (or even infinite), TD methods are to preferred.

(f) Depending on the state and action space formulation the choice of whether to use FA methods or not is to be decided. For example, if either state or action space is huge / continuous, then FA methods are to be used.

Problem 3 : Bellman Equations and Dynamic Programming

Let $M =< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ be an MDP with fintite state and action space. We further assume that the reward function $\mathcal{R}$ to be a deterministic function of current state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$. Let f be a arbitraty action value function mapping a state-action pair of the MDP to a real number, i.e. $f, : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Prove that

$$\left\| V^* - V^{\pi_f} \right\|_\infty \leq \frac{2\|f - Q^*\|_\infty}{1-\gamma}$$

with π_f being the greedy policy with respect to f.

[**Note :** The max norm of an action value function f is defined as $\|f\|_\infty = \max_s \max_a |f(s,a)|$]

For all state $s \in \mathcal{S}$, we have,

$$V^*(s) - V^{\pi_f}(s) \quad = \quad Q^*(s, \pi^*(s)) - Q^{\pi_f}(s, \pi_f(s))$$

Adding and subtracting the term $Q^*(s, \pi_f(s))$ to the RHS, we have,

$$V^*(s) - V^{\pi_f}(s) \quad = \quad Q^*(s, \pi^*(s)) - Q^*(s, \pi_f(s)) + Q^*(s, \pi_f(s)) - Q^{\pi_f}(s, \pi_f(s))$$

Since the third and fourth term take the same action, $\pi_f(s)$, at state s, we can rewrite those terms as,

$$\begin{aligned} &= \quad Q^*(s, \pi^*(s)) - Q^*(s, \pi_f(s)) + R(s, \pi_f(s)) + \gamma \mathbb{E}_{s' \sim P(s, \pi_f(s))}[V^*(s')] \\ &- \quad R(s, \pi_f(s)) + \gamma \mathbb{E}_{s' \sim P(s, \pi_f(s))}[V^{\pi_f}(s')] \end{aligned}$$

Combining terms, we get,

$$V^*(s) - V^{\pi_f}(s) \quad = \quad Q^*(s, \pi^*(s)) - Q^*(s, \pi_f(s)) + \gamma \mathbb{E}_{s' \sim P(s, \pi_f(s))} \left[V^*(s') - V^{\pi_f}(s') \right]$$

Now consider the term $f(s, \pi_f(s)) - f(s, \pi^*(s))$. This term is clearly non-negative. That is $f(s, \pi_f(s)) - f(s, \pi^*(s)) \geq 0$ because the policy π_f is the greedy policy with respect to f. (Recall that after the first action, all other terms in $f(s, \pi_f(s))$ and $f(s, \pi^*(s))$ are same and the first action being greedy in $f(s, \pi_f(s))$ would fetch a better reward.)

$$\begin{aligned} V^*(s) - V^{\pi_f}(s) \quad &\leq \quad Q^*(s, \pi^*(s)) - f(s, \pi^*(s)) + f(s, \pi_f(s)) - Q^*(s, \pi_f(s)) \\ &+ \quad \gamma \mathbb{E}_{s' \sim P(s, \pi_f(s))} \left[V^*(s') - V^{\pi_f}(s') \right] \end{aligned}$$

Now, the first two terms take the same action and third and fourth terms take the same action and the last term is an average term. Hence, we get the inequality,

$$V^*(s) - V^{\pi_f}(s) \quad \leq \quad 2 \left\| f - Q^* \right\|_\infty + \gamma \left\| V^* - V^{\pi_f} \right\|_\infty$$

from which the result follows.

Bonus Question : Prove that the following two definitions of state value functions are equivalent.

$$\begin{aligned} V^\pi(s) \quad &= \quad \mathbb{E}_[\, G_t | s_t = s] \\ V^\pi(s) \quad &= \quad \mathbb{E}_[\, G_0 | s_0 = s] \end{aligned} \tag{1}$$

Problem 4 : Optimal Stopping Problem

Consider a dice game in which a player is eligible for a reward that is equal to $3x^2 + 5$ where x is the value of the face of the dice that comes on top. A player is allowed to roll the dice at most N times. Af every time step, after having observed the outcome of the dice roll, the player can pick the eligible reward and quit the game or roll the dice one more time with no immediate reward. If not having stopped before, then, at terminal time N, the game ends and the player gets the reward corresponding to the outcome of dice roll at time N.

The goal of this problem is to model the game as an MDP and formulate a policy that helps the player decide, at any time step $n < N$, whether to continue or quit the game. As a specific case, let's consider a fair four sided dice for this game. It then follows that one can model the

game as a finite horizon MDP (with horizon N) consisting of four states $\mathcal{S} = \{1, 2, 3, 4\}$ and two actions $\mathcal{A} = \{Continue, Quit\}$. One can assume that the discount factor (γ) is 1. For any $n \leq N$, denote $V^n(s)$ and $Q^n(s, a)$ as the state and action functions for state s and action a at time step n.

(a) Evaluate the value function $V^N(s)$ for each state s of the MDP.

(b) Compute $Q^{N-1}(s, a)$ for each state-action pair of the MDP.

(c) Evaluate the value function $V^{N-1}(s)$ for each state s of the MDP.

(d) For any time $2 < n \leq N$, express $V^{n-1}(s)$ recursively in terms of $V^n(s)$.

(e) For any time $2 < n \leq N$, express $Q^{n-1}(s, "Continue")$ in terms of $Q^n(s, "Continue")$.

(f) What is the optimal policy at any time n that lets a player decide whether to continue or quit based on current state s ?

(g) Is the optimal policy stationary or non-stationary ? Explain.

(a) At terminal time N, $V^N(1) = 8$, $V^N(2) = 17$, $V^N(3) = 32$ and $V^N(4) = 53$

(b) Note that at terminal time, there is no action to be taken. So, Q-values can only be computed from time $N - 1$. We then have,

$$Q^{N-1}(i, "Continue") = \frac{1}{4} \sum_{j=1}^{4} V^N(j) = 27.5$$

and

$$Q^{N-1}(i, "Quit") = 3i^2 + 5$$

for $i \in \{1, 2, 3, 4\}$.

(c) $V_*^{N-1}(i) = \max(Q^{N-1}(i, "Continue"), Q^{N-1}(i, "Quit")) = \max(27.5, 3i^2+5)$ for $i \in \{1, 2, 3, 4\}$. Therefore, we have,

$$V^{N-1}(1) = \max(27.5, 8) = 27.5$$

$$V^{N-1}(2) = \max(27.5, 17) = 27.5$$

$$V^{N-1}(3) = \max(27.5, 32) = 32$$

$$V^{N-1}(4) = \max(27.5, 53) = 53$$

(d) The value function can be computed as

$$V^{n-1}(i) = \max(Q^{n-1}(i, "Continue"), Q^{n-1}(i, "Quit")) = \max(\frac{1}{4} \sum_{j=1}^{4} V^n(j), 3i^2 + 5)$$

(e) For any $n < N$, note that $Q^n(i, "Continue")$ does not depend on current state i. Once the player has decided to $"Continue"$ the number shown by the dice at time n is irrelevant. We can simplify the notation $Q^n(i, "Continue")$ as $Q(n)$

$$Q(n-1) = \frac{1}{4}\sum_{i=1}^{4} V^n(i) = \frac{1}{4}\sum_{i=1}^{4} \max(Q(n), Q^n(i, "Quit"))$$

$$= \frac{1}{4}\sum_{i=1}^{4} \max(Q(n), 3i^2 + 5)$$

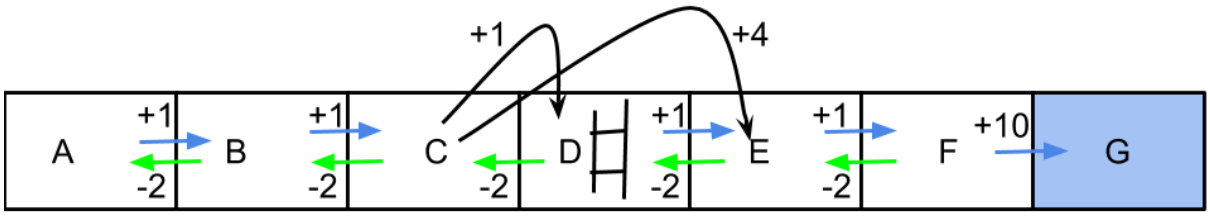
(f) The optimal policy at time n is given by

$$\pi^n(s) = \begin{cases} \text{Continue}, & \text{if } Q(n) > 3i^2 + 5 \\ \text{Quit}, & \text{otherwise} \end{cases}$$

(g) The policy is clearly non-stationary as it on time instant and the current state.

Problem 5 : Model Free Prediction and Control

Consider the MDP shown below with states $\{A, B, C, D, E, F, G\}$. Normally, an agent can either move *left* or *right* in each state. However, in state C, the agent has the choice to either move *left* or *jump* forward as the state D of the MDP has an hurdle. There is no *right* action from state C. The *jump* action from state C will place the agent either in square D or in square E with probability 0.5 each. The rewards for each action at each state s is depicted in the figure below alongside the arrow. The terminal state is G and has a reward of zero. Assume a discount factor of $\gamma = 1$.



Consider the following samples of Markov chain trajectories with rewards to answer the questions below

- $A \xrightarrow{+1} B \xrightarrow{+1} C \xrightarrow{-2} B \xrightarrow{+1} C \xrightarrow{+1} D \xrightarrow{+1} E \xrightarrow{+1} F \xrightarrow{+10} G$
- $A \xrightarrow{+1} B \xrightarrow{+1} C \xrightarrow{+1} D \xrightarrow{+1} E \xrightarrow{+1} F \xrightarrow{+10} G$
- $A \xrightarrow{+1} B \xrightarrow{+1} C \xrightarrow{+4} E \xrightarrow{+1} F \xrightarrow{+10} G$
- $A \xrightarrow{+1} B \xrightarrow{+1} C \xrightarrow{+4} E \xrightarrow{-2} D \xrightarrow{+1} E \xrightarrow{+1} F \xrightarrow{+10} G$
- $A \xrightarrow{+1} B \xrightarrow{+1} C \xrightarrow{+4} E \xrightarrow{-2} D \xrightarrow{+1} E \xrightarrow{+1} F \xrightarrow{-2} E \xrightarrow{+1} F \xrightarrow{+10} G$

(a) Evaluate $V(s)$ using first visit Monte-Carlo method for all states s of the MDP.

(b) Which states are likely to have different value estimates if evaluated using every visit MC as compared to first visit MC ? Why ?

(c) Now consider a policy π_f that always move forward (using actions *right* or *jump*). Compute **true** values of $V^{\pi_f}(s)$ for all states of the MDP.

(d) Suppose, using policy π_f, we collect infinitely many trajectories of the above MDP. If we compute the value function V^{π_f} using Monte Carlo and TD(0) evaluations, would the two methods converge to the same value function ? Justify your answer.

(e) Fill in the blank cells of the table below with the Q-values that result from applying the Q-learning update for the 4 transitions specified by the episode below. You may leave Q-values that are unaffected by the current update blank. Use learning rate $\alpha = 0.5$. Assume all Q-values are initialized to 0.

s	a	r	s	a	r	s	a	r	s	a	r	s
C	jump	4	E	right	1	F	left	-2	E	right	+1	F

	Q(C, left)	Q(C, jump)	Q(E, left)	Q(E, right)	Q(F, left)	Q(F, right)
Initial	0	0	0	0	0	0
Transition 1						
Transition 2						
Transition 3						
Transition 4						

(f) After running the Q-learning algorithm using the four transitions given above, construct a greedy policy using the current values of the Q-table in states C, E and F.

(a) The values are computed as follows
(i) $V(A) = (14 + 15 + 17 + 16 + 15)/5 = 77/5 = 15.4$
(ii) $V(B) = (13 + 14 + 16 + 15 + 14)/5 = 72/5 = 14.4$
(iii) $V(C) = (12 + 13 + 15 + 14 + 13)/5 = 67/5 = 13.4$
(iv) $V(D) = (12 + 12 + 12 + 11)/4 = 47/4 = 11.75$
(v) $V(E) = (11 + 11 + 11 + 10 + 9)/5 = 47/4 = 10.4$
(vi) $V(F) = (10 + 10 + 10 + 10 + 9)/5 = 9.8$ and $V(G) = 0$

(b) States $\{B, C, E, F\}$ are likely to have different value estimates when evaluated using every visit MC as these are visited more than once in a single rollout.

(c) For the forward policy π_f, we would have $V(G) = 0, V(F) = 10, V(E) = 11, V(D) = 12, V(C) = (13 + 15)/2 = 14, V(B) = 15, V(A) = 16$

(d) If we have infinitely many trajectories from a policy π, then the value function estimates using the MC method will converge to the true value function due to law of large numbers. For the TD(0), the MDP needs to be Markovian which in our example holds true. If the choice of the learning rate of the TD(0) algorithm, obeys the Robbins-Monroe condition, generally TD(0) methods converge to the true value function.

(e) Q-Evaluations are provided in the table. A state-action is only updated when a transition is made from it. Q(C; left), Q(E; left), and Q(F; right) state-actions are never experienced and so these values are never updated. The Q-learning update rule is given by,

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

Using the above update rule, the four updates are given by,

$$\begin{aligned} 2 &= 0 + 0.5(4 + 0 - 0) \\ 0.5 &= 0 + 0.5(1 + 0 - 0) \\ -0.75 &= 0 + 0.5(-2 + 0.5 - 0) \\ 0.75 &= 0.5 + 0.5(1 + 0 - 0.5) \end{aligned}$$

On transition 2, the Qs for F are still both 0, so the update increases the value by the reward +1 times the learning rate. On transition 3, the reward of -2 and $Q(E; right) = 0 : 5$ are included in the update. On transition 4, $Q(F; left)$ is now -0:75 but $Q(F; right)$ is still 0 so the next update to $Q(E; right)$ uses 0 in the max over the next state's action

	Q(C, left)	Q(C, jump)	Q(E, left)	Q(E, right)	Q(F, left)	Q(F, right)
Initial	0	0	0	0	0	0
Transition 1		2				
Transition 2				0.5		
Transition 3					- 0.75	
Transition 4				0.75		

(f) The greedy policy in states C, E and F is given by,

$$\pi(s) = \left\{ \begin{array}{ll} \text{jump}, & \text{for } s = C \\ \text{right}, & \text{for } s = E \\ \text{right}, & \text{for } s = F \end{array} \right\}$$

Problem 6 : Short Questions

(a) Explain if it is possible to parallelize the value iteration algorithm

Since V_{t+1} depends only on V_t, it is easy parallelize the back up operation for all states of the MDP

(b) For the policy evaluation problem, what are the advantages of using Monte-Carlo over DP methods ?

MC algorithms are model free, unbiasaed and work in non-Markovian setting as well. It doesn't require full back up to evaluate the value of the state.

(c) For the policy evaluation problem, what are the advantages of using Monte Carlo over TD methods ?

Non-Markovian, unbiased estimate of value function...

(d) For the policy evaluation problem, what are the advantages of using TD methods over Monte Carlo ?

Online, can work with partial sequences.

(e) What are the reasons for having a discount factor in defining a MDP ?

Convergence of Bellman updates, finite reward sum and variance reduction are some reasons to have discount factor

(f) Why does TD methods for policy evaluation yields an biased estimate of the true value of the policy ?

TD methods use bootstrapping and hence they yield a biased estimate of the value function

(g) Why are we studying TD)(λ) methods ?

Better Bias variance trade off

(h) Consider the grid world navigation problem. Following are two choices for reward function. One reward formulation assigns a reward of +5 at the goal state and -1 elsewhere. The other reward function is identical to the first one but assigns an additional reward of +2 to states that are two squares away from the goal state. Which reward formulation is suitable for the task to make the RL agent navigate the agent from a start state in the grid to the goal state in as few a steps as possible ? Explain.

The first reward formulation. Since, in the second reward formulation the agent can get carried away by the reward of +2 and keep looping around that state which is not task we want the agent to learn

(i) Given a MDP, does scaling of rewards using a positive scale factor, change the optimal policy ?

No. Explanation is required using expanding expectation of discounted reweards with and without scaling of rewards.

(j) Consider an MDP with finite state and finite action. How do we understand that the policy iteration converges ?

Successive iterates of the policy iteration yield a better policy. If the improvement stops, then we are at optimal policy. Except for the initial policy, all sucessive policies are deterministic.

There are only finite number of deterministic policies possible in a finite state finite action MDP. In fact, the total number of deterministic policies in this case are $|\mathcal{A}|^{|\mathcal{S}|}$

(k) Why don't we use importance sampling for 1-step Q-learning?

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

That update rule actually matches exactly what the "target policy" (greedy policy in this case) is doing; we update the Q value for the state-action pair for which we've just obtained a new observation (s_t, a_t), with the new reward being R_{t+1}) under the assumption that we follow up with the greedy / target policy immediately afterwards (resulting in $\max_a Q(s_{t+1}, a)$). In this equation, the only action that we may not have taken according to the target policy is the action a_t, but that's fine because precisely that same action is the one for which we're updating the Q-value.