AI 3000 / CS 5500 : Reinforcement Learning

Assignment № 2

Due Date : 18/10/2024

Couse Instructor : Easwar Subramanian 20/09/2024

Problem 1 : Dynamic Programming and Model Free Methods

(a) Let M be an infinite horizon MDP and let π be a policy. Suppose if the value iteration algorithm to calculate V^π is terminated after $k+1$ iterations with $\left\|V^\pi_{k+1} - V^\pi_k\right\|_\infty < \epsilon$ for some chosen $\epsilon > 0$, how far is the estimate V_{k+1} from the true value function V^π ? Provide details of your derivation. (3 Points)

The last iterate of the algorithm is V_{k+1} and we know that $\|V_{k+1} - V_k\|_\infty \leq \varepsilon$. By using the triangular inequality (of norms) and by using the fact $BV_k = V_{k+1}$ where B is the Bellman evaluaiton backup, we have,

$$\begin{aligned}\|V_k - V\|_\infty &\leq \|V_k - V_{k+1}\|_\infty + \|V_{k+1} - V\|_\infty = \|V_k - V_{k+1}\|_\infty + \|BV_k - BV\|_\infty \\ &\leq \|V_k - V_{k+1}\|_\infty + \gamma\|V_k - V\|_\infty = \varepsilon + \gamma\|V_k - V\|_\infty\end{aligned}$$

Therefore, $\|V_k - V\|_\infty \leq \frac{\varepsilon}{1-\gamma}$. This allows us to conclude that,

$$\|V_{k+1} - V\|_\infty = \|BV_k - BV\|_\infty \leq \gamma\|V_k - V\|_\infty \leq \frac{\gamma\varepsilon}{1-\gamma}$$

(b) Prove that the Bellman evaluation operator $\mathcal{L}^\pi$ satisfies the monotoniciity property. That is, for any two value functions u and v such that $u \leq v$ (this means, $u(s) \leq v(s)$ for all $s \in \mathcal{S}$), we have $\mathcal{L}^\pi(u) \leq \mathcal{L}^\pi(v)$ (2 Points)

By the defintion of Bellman evaluation operator $\mathcal{L}^\pi$, for a fixed but arbitrary state $s \in \mathcal{S}$, we have,

$$L^\pi(u) = \mathcal{R}^\pi + \gamma\mathcal{P}^\pi u$$

where

$$\begin{aligned}\mathcal{P}^\pi(s'|s) &= \sum_{a\in\mathcal{A}} \pi(a|s)\mathcal{P}^a_{ss'} \\ \mathcal{R}^\pi(s) &= \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'} \mathcal{P}^a_{ss'}\mathcal{R}^a_{ss'} = \mathbb{E}(r_{t+1}|s_t = s)\end{aligned}$$

Since $u(s) \leq v(s)$, we have, $\gamma\mathcal{P}^\pi u \leq \gamma\mathcal{P}^\pi v$ and R^π is nothing but the expected reward from state s for taking an action using policy π, we are adding the same term to both sides of the inequality which proves the result.

(c) In the TD(λ) algorithm, we use λ returns as the target. The λ return target is given by,

$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

where $G_t^{(n)}$ is the n-step return defined as,

$$G_t^{(n)} = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1}r_{t+n} + \gamma^n V(s_{t+n}).$$

The parameter λ is used to determine the weights corresponding to each of the n-step returns in the λ-return target and the weights decay exponentially with n. Let $\eta(\lambda)$ denote the time by which the weighting sequence would have fallen to half of its initial value. Derive an expression that relates the parameter λ to $\eta(\lambda)$. Use the expression derived to compute the value of λ for which the weights would drop to half after 3 step returns. (4 Points)

The λ-return is defined as

$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

Defining $\tau = \eta(\lambda)$ to be the value of n such that

$$\lambda^\tau = \frac{1}{2} \implies \tau \approx \frac{\ln(\frac{1}{2})}{\ln(\lambda)}$$

Thus given λ, the half life τ is given by this expression. For example if $\lambda \approx \frac{1}{\sqrt{2}}$, then we compute that $\tau = 3$ so we are looking that many steps ahead before our weighting drops by one half.

(d) Assume an MDP where four transitions are provided as shown in the snapshot below. 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. 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. (3 Points)

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	-10	-10	-10	-10	-10	-10
Transition 1						
Transition 2						
Transition 3						
Transition 4						

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		

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

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

As there was a typo in the question (one place said Q-values are initialized to 0 and another said Q-values are initialized to 10, both answers are accepted.)

If the start with the initial Q-values as 10, then the Q-table values will be updated as

	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		-8				
Transition 2				-9.5		
Transition 3					- 10.75	
Transition 4				-9.25		

The greedy policy remians the same.