AI 3000 / CS5500 : Reinforcement Learning
Exam № 1

Easwar Subramanian, IIT Hyderabad 30/09/2022, 4.00 PM

Problem 1 : Markov Reward Process

A fair coin is tossed repeatedly and independently. By formulating a suitable Markov reward process and using Bellman equations, find the expected number of tosses required for the pattern HTH to appear. (5 Points)

Call HTH our target. Consider a chain that starts from a state called nothing (denote by $\emptyset$) and is eventually absorbed at HTH. If we first toss H then we move to state H because this is the first letter of our target. If we toss a T then we move back to $\emptyset$ having expended 1 unit of time. Being in state H we either move to a new state HT if we bring T and we are 1 step closer to the target or, if we bring H, we move back to H: we have expended 1 unit of time, but the new H can be the beginning of a target. When in state HT we either move to HTH and we are done or, if T occurs then we move to $\emptyset$. The transition diagram looks like below.

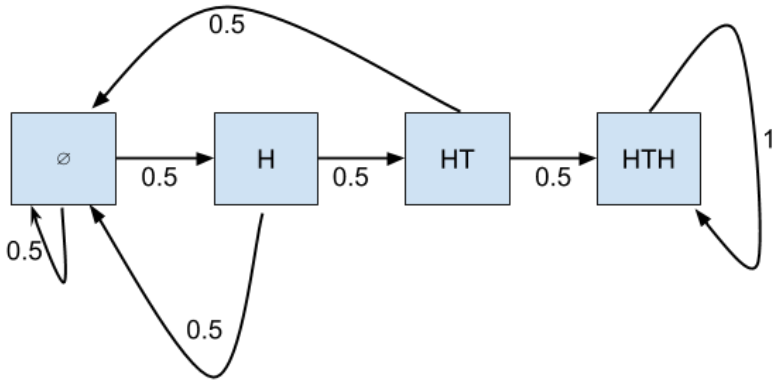


Figure 1: Suitable Markov Reward Process

Now we can write down the states of the MRP $< \mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma >$ as follows.

- The set of states $\mathcal{S} = \{\emptyset, H, HT, HTH\}$
- The transition matrix $\mathcal{P}$ is given by,

$$\begin{array}{c} \\ \emptyset \\ H \\ HT \\ HTH \end{array} \begin{array}{c} \begin{array}{cccc} \emptyset & H & HT & HTH \end{array} \\ \begin{pmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

- The absorbing state is HTH and this MRP is very similar to the snake and ladder problem discussed in the class. So, every time we toss a coin, we get a reward of -1 and when we reach the absorbing state we get a reward of 0. So, $\mathcal{R}(s) = -1 \quad \text{for} \quad s \in \{\phi, H, HT\}$ and $\mathcal{R}(HTH) = 0$.
- The discount factor $\gamma = 1$.

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 coin tosses required to reach state HTH from any other state s of the MRP. The matrix $(I - \gamma P)$ becomes invertible if we set $V(s) = 0$ for $s = HTH$. One may find the inverse of the matrix $(I - \gamma\mathcal{P}_{3\times3})$ and multiply with $\mathcal{R}_{3\times1}$ to compute the expected coin tosses from any given state of the MRP. Specifically, we are interested from state ϕ. Upon solving one can find that the expected number of coin tosses from state $\emptyset$ to reach HTH is 10.

Problem 2 : Bellman Equations and Dynamic Programming

Consider a MDP $M = < \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >)$ with 2 states $\mathcal{S} = \{s_1, s_2\}$. From each state there are 2 available actions $\mathcal{A} = \{a_1, a_2\}$. Choosing action a_1 from any state leaves you in same state and gives a reward of -1. Choosing action a_2 from state s_1 takes you to state s_2 by giving a reward -2, while choosing a_2 from state s_2 ends the episode giving reward 3.

(a) Depict the above MDP in figure. (1 Point)

It is useful to consider a notional terminal state like T, to represent episode termination, as

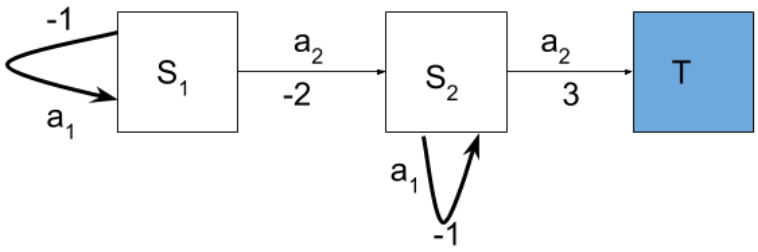


Figure 2: Markov Decision Process

that it will aid in value iteration algorithm. .

[Some of you might have drawn action a_2 ending in state s_2 with reward 3 and that is wrong and it will also not help you in correct value function computation.]

(b) For the MDP described above, initialize value function to zero and perform value iteration (for maximum of 5 iterations) and comment whether the value function converges to optimal value function. (2 Points)

Let us initialize value iteration as $V_0 = [0; 0]$. Then $V_1 = [-1; 3]$ and $V_2 = [1; 3]$. We also have $V^* = [1; 3]$. It only takes two iterations for the value function to converge.

(c) Does the successive value function iterates (from part (b)) are monotonic for all states $s \in \mathcal{S}$ of the MDP ? If not, would that contradict the fact the Bellman optimality operator is a contraction ? Explain. (3 Points)

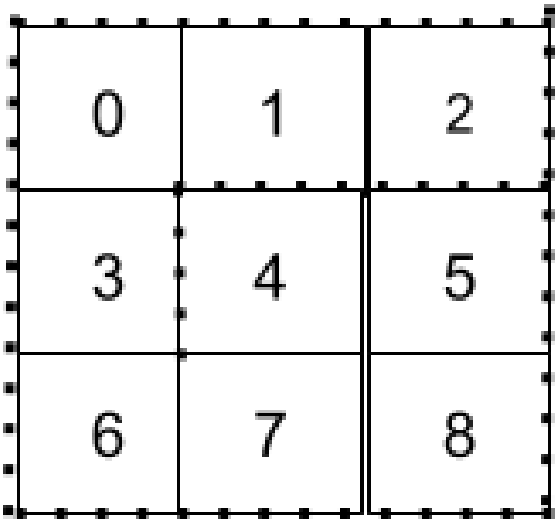
Clearly, for state S_1, convergence is clearly not monotonic. The Bellman operator is a contraction. This means that the maximum absolute value across all states of the error (difference of the value function compared to the optimal value function) must not increase between iterations of value iteration. However, for individual states it is possible that the error increases between iterations. In this example, $\|V_0 - V^*\|_\infty = 3, \quad \|V_1 - V^*\|_\infty = 2, \quad \|V_2 - V^*\|_\infty = 0$. So clearly there is no contradiction. The key idea is here is the use of infinity or max norm, so considering error estimates of value function on individual states is not the correct way to verify the contraction property.

(d) In general, for any MDP, why does the value iteration algorithm converge ? (2 Points)
Bellman operator (both evaluation and optimality) are contractions under max norm and then one can use Banach fixed point theorem.

(e) For an arbitrary MDP, would value and policy iteration algorithms converge to the same optimal policy π^* ? Justify the answer. (2 Points)
No, need not. Policy iteration can converge to a different (deterministic) optimal policy. But both will converge to same V^* as V^* is unique whereas there could be many different optimal policies agreeing on the value function.

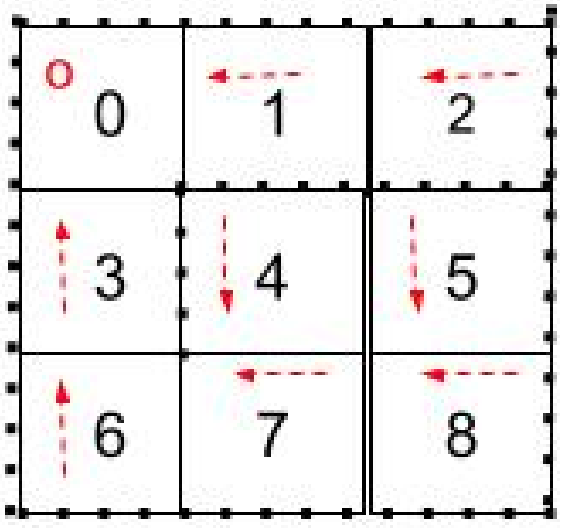
Problem 3 : On Discounting

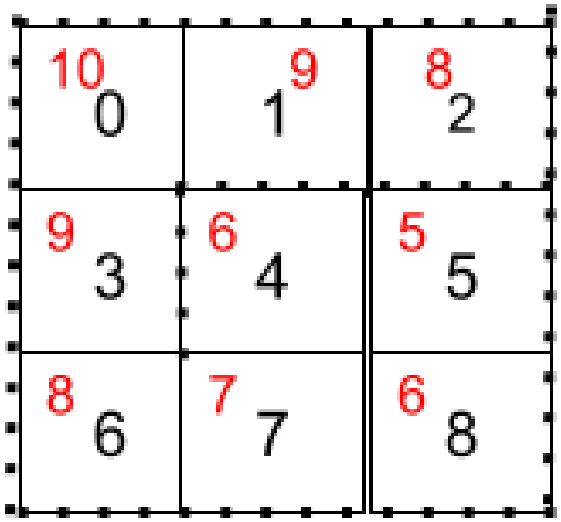
(a) Given a MDP, does scaling the discount factor using a scale factor $\kappa \in (0, 1)$, change the optimal policy ? Explain. (2 Points)
Yes, it does change the optimal policy. A classic example, is the exercise given by in Assignment 1 (Effect of Noise and Discounting)....



For the rest of the sub-questions, consider the 3×3 grid world shown below. Each square is a state of the MDP and there are five possible actions given by $\mathcal{A} = \{\text{Left}, \text{Right}, \text{Up}, \text{Down}, \text{Stay}\}$. The "Stay" action at any state $s \in \mathcal{S}$ of the MDP let the agent reside in the same state. The agent navigating the grid world cannot move through the walls. The walls are on the four corners of the grid world and between squares $(1,4), (2,5), (3,4)$. The agent gets a reward of +1 for any action taken at state 0 and there are no rewards for actions taken at any other state. More precisely, $R(0,a) = 1$ for any action $a \in \mathcal{A}$ and $R(s',a) = 0$ for all $a \in \mathcal{A}$ and $s' \neq 0$.

(a) Consider horizon length of $H = 10$ and no discounting. Fill in the optimal policy and optimal value function for each state of the MDP. (2 Points)





(b) For the value function filled in part (a), write out $Q(0, \text{Right})$ and $Q(4, \text{Right})$. (1 Point)

We note that, for any state s,

$$Q_{s,a}^{10 \text{ steps to go}} = r_s + V_{s'}$$

where s' is the sucessor state after taking action a in state s. Hence, $Q(0, \text{Right}) = 9$ and $Q(4, \text{Right}) = 4$

(c) For this sub-question, assume that the actions that make the agent hit the wall and the stay action are not available. For example, at state 0, the only available actions are $\{\text{Right}, \text{Down}\}$ (2 Points)

(1) How many optimal actions are there in state $s = 0$? What are they ? Right and down are optimal actions...

(2) What is the (optimal) value of state $s = 0$? The optimal value at state 0 is 5....

(d) Assume infinite horizon and $\gamma = 0.9$. What is the (optimal) value of the state $s = 0$? (2 Points) Using the property of geometric series $(1 + \gamma + \gamma^2 + \cdots$ converges to $\frac{1}{1-\gamma}$ we see that the optimal value at state 0 is 10. .

(e) Assume infinte horizon and no discounting. At every time step, after the agent takes an action and gets the reward, a power outage can occur with probablity $p = 0.1$ and terminates the game. Calculate the (optimal) value of the state $s = 0$. (2 Points) $\frac{1}{\alpha}$...

Problem 4 : Sample Based Estimation

Let $M =< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >)$ be a finite state, finite action, infiinte horizon MDP with $\gamma < 1$ with unknown reward function $\mathcal{R}$ and transition probablities $\mathcal{P}$. For simplicity, let $\mathcal{R}$ be bounded within $[0, R_{\max}]$.

(a) Prove that the MDP M has bounded returns (2 Points)
Lower bound is 0; Upper bound is $\frac{R_{\max}}{1-\gamma}$ where $R_{\max}$ is the maximum reward possible. (Check it out)...

Suppose that we are given MLE estimates $\hat{\mathcal{R}}$ and $\hat{\mathcal{P}}$ such that

$$\begin{aligned} \max_{s,a} \left| \hat{R}(s,a) - R(s,a) \right| &< \varepsilon_R \\ \max_{s,a} ||\hat{P}(s'|s,a) - P(s'|s,a)||_1 &< \varepsilon_P \end{aligned}$$

where $||\cdot||_1$ is the L_1 norm. Define the approximate MDP $\widehat{M}$ as $< \mathcal{S}, \mathcal{A}, \hat{\mathcal{P}}, \hat{\mathcal{R}}, \gamma >)$. In addition, for a given deterministic policy π, let the error in state value, between the two MDPs, be bounded as,

$$||V_M^\pi - V_{\widehat{M}}^\pi||_\infty \leq \frac{\varepsilon_R}{1-\gamma} + \gamma \varepsilon_P \frac{R_{\max}}{2(1-\gamma^2)}$$

(b) Prove that for all $s \in \mathcal{S}$ of the MDP

$$V_M^{\pi^*}(s) - V_M^{\widehat{\pi}^*}(s) \leq 2 \left(\frac{\varepsilon_R}{1-\gamma} + \gamma \varepsilon_P \frac{R_{\max}}{2(1-\gamma^2)} \right)$$

where π^* and $\widehat{\pi}^*$ are (deterministic) optimal policies of M and $\widehat{M}$ respectively. (4 Points)

[**Note :** The RHS measures the loss incured by using the optimal policy for $\widehat{M}$ instead of using the optimal policy for M. Lesser the loss, the better is the approximation]

There was a typo in the question paper due to typesetting oversight. The left hand side of (b) should have read $V_M^{\pi^*}(s) - V_M^{\widehat{\pi}^*}(s)$ instead of $V_M^{\pi^*}(s) - V_{\widehat{M}}^{\widehat{\pi}^*}(s)$. Apologies for the mistake. Appopriate adjustment will be made to the overall socre to all students.

From the definition of value function, we get,

$$\begin{aligned} V_M^{\pi^*}(s) - V_M^{\widehat{\pi}^*}(s) &= V_M^{\pi^*}(s) - V_{\widehat{M}}^{\pi^*}(s) + V_{\widehat{M}}^{\pi^*}(s) - V_{\widehat{M}}^{\widehat{\pi}^*}(s) \\ &\leq ||V_M^{\pi^*} - V_{\widehat{M}}^{\pi^*}||_\infty + V_{\widehat{M}}^{\pi^*}(s) - V_{\widehat{M}}^{\widehat{\pi}^*}(s) \\ &\leq ||V_M^{\pi^*} - V_{\widehat{M}}^{\pi^*}||_\infty + V_{\widehat{M}}^{\widehat{\pi}^*}(s) - V_{\widehat{M}}^{\widehat{\pi}^*}(s) \\ &\leq ||V_M^{\pi^*} - V_{\widehat{M}}^{\pi^*}||_\infty + ||V_{\widehat{M}}^{\widehat{\pi}^*} - V_{\widehat{M}}^{\widehat{\pi}^*}||_\infty \\ &\leq 2\left(\frac{\varepsilon_R}{1-\gamma} + \gamma\varepsilon_P \frac{R_{\max}}{2(1-\gamma^2)}\right) \end{aligned}$$

(b) We are helping a robot learn a policy π which is helpful in performing a real world application. Assume that the real world task is modelled using the MDP M. We only have access to a simulation software to generate samples and not from the robot directly. The simulation software uses a transition model $\hat{P}(s'|s,a)$ instead of the real transition model given by $P(s'|s,a)$. Provide an expression for the update rule for learning Q^π using samples generated by the simulator. (3 Points)

Off-policy learning will come to aid. Use of importance sampling in the Q^π update rule is the key

(b) Continuing with set up from previous question; now the scenario is that we need to train n robots simulataneously. Each robot has its own transition model P (assume other elements of the MDP are same). Is it possible to make all robots simultaneously learn about policy π by using samples generated by the simulator ? If yes, would the learnt action value function Q^π be identical for all robots ? (3 Points)

Yes, possible to simultanously learn. Again using IS. The learnt Q^π would be different as iS ratios would be different for different robots.

Problem 5 : Q-Learning

A RL agent collects experiences of the form $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$ to update Q values. At each time step, to choose an action, the agent follows a fixed policy π with probablity 0.5 or chooses an action in uniform random fashion. Assume the updates are applied infinitely often, state-action pairs are visited infintely often, the discount factor $\gamma < 1$ and the learning rate scheduling is appropriate.

(a) The Q learning agent performs following update

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)]$$

Will this update converge to the optimal Q function ? Why or Why not ? If not, will it converge to anything at all ? (3 Points)

Yes. Q-learning is an off-policy control algorithm and the target is based on Bellman optimality condition. Provided other conditions as stated in the question are true, this update will converge to Q^*.

(b) Another reinforcemnt learning called SARSA agent, performs the following update

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

Will this update converge to the optimal Q function ? Why or Why not ? If not, will it converge to anything at all ? (3 Points)

No, it will not converge to optimal Q function. Rather it will be converge $Q^{\pi'}$ where π' is a policy that, at each time step, chooses an action based on policy π with probablity 0.5 or chooses an action in uniform random fashion. Recall that SARSA update is on-policy.

(c) Consider an MDP with three states $\{s_1, s_2, s_3\}$ and three actions $\{a_1, a_2, a_3\}$ with discount factor $\gamma = 0.5$. There is no noise in the environment and therefore all actions result in intended state transitions. The reward for transitioning into a state s_i is i. For example, if any action $a_k, k\{1, 2, 3\}$ pushes the agent into state s_3, then the reward is 3. We consider a Q-learning agent that uses the ϵ-greedy strategy. When the Q-values from a particular state are same for more than one action, the agent breaks ties by choosing the action a_k with lowest k. Let us initialize the Q table to zeros for all state-action pairs and let the learning rate be set to $\alpha = 0.7$. For $\epsilon \neq 0$, could the Q-learning agent generate the following trajectory given by

$$(s_1, a_1, 1, s_1, a_2, 2, s_2)$$

If yes, reason out, which of the two action is greedy and which of it is random ? (3 Points)

Clearly, it is possible. For the first experience tuple $(s_1, a_1, 1, s_1)$, we start at $s1$ and select action a_1, which returns the reward as 1 and stay at the same state. The initial Q for all state-action is zeros. Since that ties are broken by choosing a_i with the smallest index i, the greedy action is thus a_1. Also, the random action can also pick the greedy action. For the second experience tuple $(s_1, a_2, 2, s_2)$, we start at s_2 and select action a_2, which returns the reward 2 and goes to state s_2. Now, we've updated the $Q(s_1; a_1)$ based on the first experience tuple, and $Q(s_1; a1) = 0.7$. Thus the optimal action for s_1 is a_1 now. The action a_2 could be taken only when the action is randomly selected.

(d) Consider a single state state MDP with two actions. That is, $\mathcal{S} = \{s\}$ and $\mathcal{A} = \{a_1, a_2\}$. Assume the discount factor of the MDP γ and the horizon length to be 1. Both actions yield random rewards, with expected reward for action a_1 being constant $c \geq 0$ and that of acton a_2 is given by $c + \mathcal{N}(-0.2, 1)$ (normal distribution with mean -0.2 and unit variance). Which is the better action to take in expectation ? Would the TD control agents like Q-learning or SARSA control, trained using finite samples, always favor the action that is best in expectation ? Explain (3 Points)

Action a_1 is a better action in expectation. But Q-learning and SARSA control can choose action a_2 because they use sample estimates to decide the best action

ALL THE BEST