AI 3000 / CS 5500 : Reinforcement Learning

Assignment № 1

Due Date : 27/09/2021

Teaching Assistants : Shantam Gulati and Megha Gupta

Easwar Subramanian, IIT Hyderabad 15/09/2021

Problem 1 : Markov Reward Process

Consider a fair four sided dice with faces marked as $\{'1','2","3","4"\}$. The dice is tossed repeatedly and independently. By formulating a suitable Markov reward process (MRP) and using Bellman equation for MRP, find the expected number of tosses required for the pattern $'1234'$ to appear. Specifically, answer the following questions.

(a) Identify the states, transition probablities and terminal states (if any) of the MRP (3 Points)

(b) Construct a suitable reward function, discount factor and use the Bellman equation for MRP to find the 'average' number of tosses required for the pattern $'1234'$ to appear. (7 Points)

[**Explanation : For the target pattern to occur, four consequective tosses of the dice should result in different faces of the dice being on the top, in the specific order** $'1'$**,** $'2'$**,** $'3'$ **and** $'4'$]

Answer

Call 1234 our target. Consider a chain that starts from a state called nothing (denote by $\emptyset$) and is eventually absorbed at 1234. If we first toss 1 then we move to state 1 because this is the first letter of our target. If we toss any other facethen we move back to $\emptyset$ having expended 1 unit of time. Being in state 1 we either move to a new state 12 if we toss 2 and we are 1 step closer to the target or, if we toss 1 we move back to state 1. If any other face shows up, we move back to $\emptyset$: we have expended 1 more unit of time. We can construct the state sequence similarly and the transition diagram looks like below.

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

- The set of states $\mathcal{S} = \{\emptyset, 1, 12, 123, 1234\}$
- The transition matrix $\mathcal{P}$ is given by,

$$\begin{array}{c} \\ \emptyset \\ 1 \\ 12 \\ 123 \\ 1234 \end{array} \begin{array}{c} \begin{array}{ccccc} \emptyset & 1 & 12 & 123 & 1234 \end{array} \\ \begin{pmatrix} 0.75 & 0.25 & 0 & 0 & 0 \\ 0.5 & 0.25 & 0.25 & 0 & 0 \\ 0.5 & 0.25 & 0 & 0.25 & 0 \\ 0.5 & 0.25 & 0 & 0 & 0.25 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

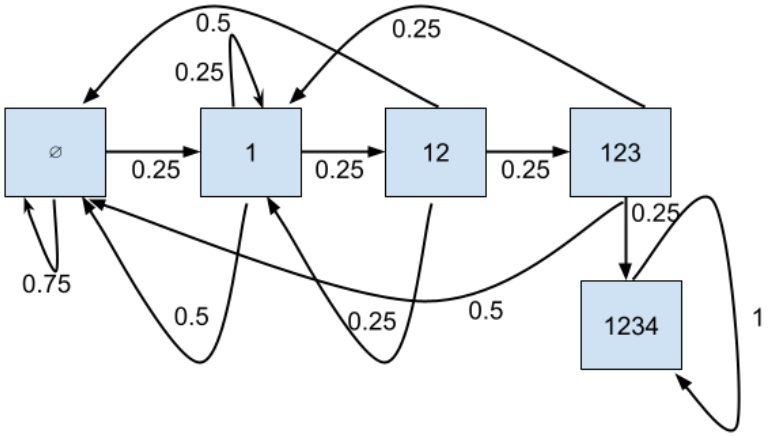

Figure 1: Suitable Markov Reward Process

- The absorbing state is 1234 and this MRP is very similar to the snake and ladder problem discussed in the class. So, every time we toss the dice, we get a reward of -1 and when we reach the absorbing state we get a reward of 0. So, $\mathcal{R}(s) = -1 \quad \text{for} \quad s \in \{\phi, 1, 12, 123\}$ and $\mathcal{R}(1234) = 0$.
- 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 dice throws required to reach state 1234 from any other state s of the MRP. The matrix $(I - \gamma P)$ becomes invertible if we set $V(s) = 0$ for $s = 1234$. One may find the inverse of the matrix $(I - \gamma\mathcal{P}_{4\times4})$ and multiply with $\mathcal{R}_{4\times1}$ to compute the expected dice throws from any given state of the MRP. Specifically, we are interested from state $\emptyset$. Upon solving one can find that the expected number of coin tosses from state $\emptyset$ to reach 1234 is 256.

Problem 2 : Finite Horizon MDP

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.

[**Hint :** A finite horizon MDP is solved backwards in time. One first computes the value of a state at terminal time and then use it to compute the value of a state at intermediate times. Note that the value of a state at any intermediate time is equal to the best action value possible for that state at that time. The best action value for a state, at any time, is evaluated by considering all possible actions from that state at that time.

(a) Evaluate the value function $V^N(s)$ for each state s of the MDP. (1 Point)

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

(b) Compute $Q^{N-1}(s, a)$ for each state-action pair of the MDP. (2 Points)

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

$V_*^{N-1}(i) = \max(Q^{N-1}(i, "Continue"), Q^{N-1}(i, "Quit")) = \max(27.5, 3i^2+5)$ 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) For any time $2 < n \leq N$, express $V^{n-1}(s)$ recursively in terms of $V^n(s)$. (2 Points)

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

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

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) What is the optimal policy at any time n that lets a player decide whether to continue or quit based on current state s ? (2 Points)

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) Is the optimal policy stationary or non-stationary ? Explain. (2 Points)

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

Problem 3 : Value Iteration

Let M be a MDP given by $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ with $|\mathcal{S}| < \infty$ and $|\mathcal{A}| < \infty$ and $\gamma \in [0, 1)$]. Let $\hat{M} =< \mathcal{S}, \mathcal{A}, \mathcal{P}, \hat{\mathcal{R}}, \gamma >$ be another MDP with a modified reward function $\hat{\mathcal{R}}$ such that

$$\left|\mathcal{R}(s, a, s') - \hat{\mathcal{R}}(s, a, s')\right| = \varepsilon.$$

Given a policy π, let V^π and $\hat{V}^\pi$ be value functions under policy π for MDPs M and $\hat{M}$ respectively.

(a) Derive an expression that relates $V^\pi(s)$ to $\hat{V}^\pi(s)$ for any state $s \in \mathcal{S}$ of the MDP. (5 Points)

Considering the definition of $V^\pi(s)$, the state value function under policy π, we have

$$V^\pi(s) = \mathbb{E}_\pi \left(\sum_{k=0}^{\infty} \gamma^k r^{t+k+1} \right)$$

We assume that each reward has a constant added to it. That is we consider the reward $\hat{r}_{t+k+1}$ in terms of r_{t+k+1} by

$$\hat{r}_{t+k+1} = r_{t+k+1} + \varepsilon$$

Then, the state value function for this new sequence of rewards is given by,

$$\begin{aligned}
\hat{V}^{\pi}(s) &= \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty}\gamma^{k}\hat{r}^{t+k+1}\right) \\
&= \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty}\gamma^{k}\left(r_{t+k+1}+\varepsilon\right)\right) \\
&= \mathbb{E}_{\pi}\left(\sum_{k=0}^{\infty}\gamma^{k}r_{t+k+1}\right)+\mathbb{E}_{\pi}\left(\gamma^{k}\epsilon\right) \\
&= V^{\pi}(s)+\mathbb{E}_{\pi}\left(\gamma^{k}\varepsilon\right)=V^{\pi}(s)+\varepsilon\sum_{k=0}^{\infty}\gamma \\
&= V^{\pi}(s)+\frac{\varepsilon}{1-\gamma}
\end{aligned}$$

The alternate relation

$$\hat{V}^{\pi}=V^{\pi}(I-\gamma P)^{-1}\varepsilon$$

is dependent on the model of the MDP.

(b) Derive an expression that relates the optimal value functions V_* and $\hat{V}_*$. (3 Points)

Althoug, one can independently derive the relationship between V_* and $\hat{V}_*$, one can use the above relationship and argue as follows. Since the above relation holds for any policy π, it also holds for optimal policy pi_* and hence we have,

$$\hat{V}_*=V_*(s)+\frac{\varepsilon}{1-\gamma}$$

(c) Will M and $\hat{M}$ have the same optimal policy ? Explain briefly. (2 Points)

The MDPs M and $\hat{M}$ will have the same optimal policy as :

$$\arg\max_a\left[\hat{r}(s,a,s')+\gamma\sum_s^{'}P(s'|s,a)\hat{V}_*(s')\right]=\arg\max_a\left[r(s,a,s')+\varepsilon+\gamma\sum_s^{'}P(s'|s,a)V_*(s')+\frac{\varepsilon}{1-\gamma}\right]$$

$$\arg\max_a\left[r(s,a,s')+\gamma\sum_s^{'}P(s'|s,a)V_*(s')+\varepsilon+\frac{\varepsilon}{1-\gamma}\right]=\arg\max_a\left[r(s,a,s')+\gamma\sum_s^{'}P(s'|s,a)V_*(s')\right]$$

Problem 4 : Effect of Noise and Discounting

Consider the grid world problem shown in Figure 2. The grid has two terminal states with positive payoff (+1 and +10). The bottom row is a cliff where each state is a terminal state with negative payoff (-10). The greyed squares in the grid are walls. The agent starts from the yellow state

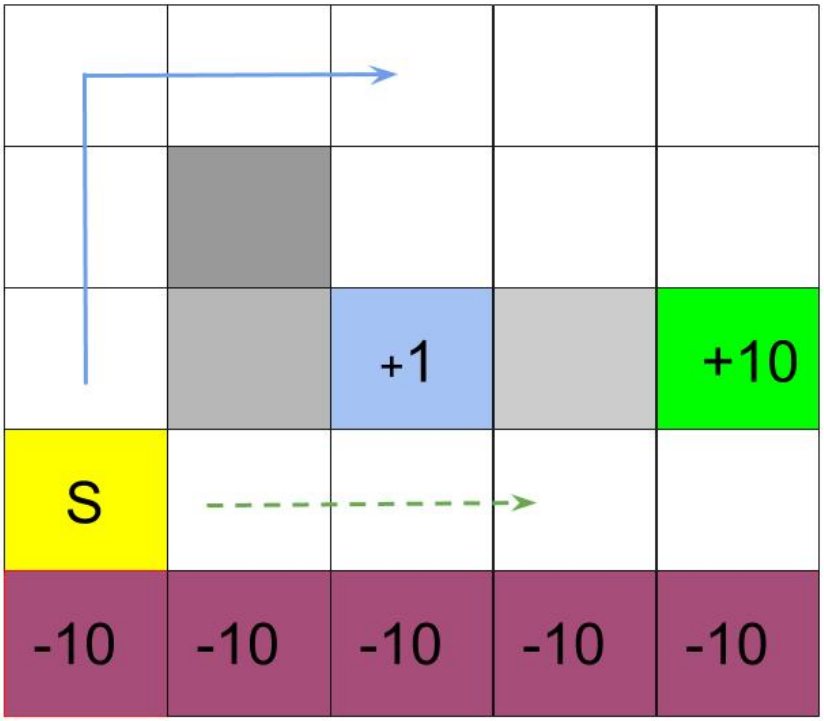

Figure 2: Modified Grid World

S. As usual, the agent has four actions $\mathcal{A} =$ (Left, Right, Up, Down) to choose from any non-terminal state and the actions that take the agent off the grid leaves the state unchanged. Notice that, if agent follows the dashed path, it needs to be careful not to step into any terminal state at the bottom row that has negative payoff. There are four possible (optimal) paths that an agent can take.

- Prefer the close exit (state with reward +1) but risk the cliff (dashed path to +1)
- Prefer the distant exit (state with reward +10) but risk the cliff (dashed path to +10)
- Prefer the close exit (state with reward +1) by avoiding the cliff (solid path to +1)
- Prefer the distant exit (state with reward +10) by avoiding the cliff (solid path to +10)

There are two free parameters to this problem. One is the discount factor γ and the other is the noise factor (η) in the environment. Noise makes the environment stochastic. For example, a noise of 0.2 would mean the action of the agent is successful only 80 % of the times. The rest 20 % of the time, the agent may end up in an unintended state after having chosen an action.

(a) Identify what values of γ and η lead to each of the optimal paths listed above with reasoning. If necessary, you could implement the value iteration algorithm on this environment and observe the optimal paths for various choices of γ and η. (10 Points)

[**Hint** : For the discount factor, try high and low γ values like 0.9 and 0.1 respectively. For noise, consider deterministic and stochastic environment with noise level η being 0 or 0.5 respectively]

Answer

1. When γ is low, RL agent is 'short sighted' and better rewards available in the distant future is not given importance. Further, when noise is zero in the environment, there is no danger of tripping to the cliff. Therefore, for low γ and low η, the agent would prefer the close exit and risk the cliff.

2. When γ is low, RL agent is 'short sighted' and better rewards available in the distant future is not given importance. Further, when noise is high or moderate in the environment, there is danger of tripping to the cliff. Therefore, for low γ and low η, the agent would prefer the close exit and not risk the cliff.

3. When γ is high, RL agent is 'far sighted' and better rewards available in the distant future is given importance. Further, when noise is low or zero in the environment, there is less or no danger of tripping to the cliff. Therefore, for high γ and low η, the agent would prefer the distant exit and risk the cliff.

4. When γ is high, RL agent is 'far sighted' and better rewards available in the distant future is given importance. Further, when noise is high or medium in the environment, there is danger of tripping to the cliff. Therefore, for high γ and high η, the agent would prefer the distant exit and not risk the cliff.

Problem 5 : On Value Iteration Algorithm

Let M be an MDP given by $< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ with $|\mathcal{S}| < \infty$ and $|\mathcal{A}| < \infty$ and $\gamma \in [0, 1)$. We are given a policy π and the task is to evalute $V^{\pi}(s)$ for every state $s \in \mathcal{S}$ of the MDP. To this end, we use the iterative policy evalution algorithm. It is the analog of the algorithm described in **slide 9 of Lecture 6** for the policy evaluation case. We start the iterative policy evaluation algorithm with an initial guess V_1 and let V_{k+1} be the $k+1$-th iterate of the value function corresponding to policy π. Our constraint on compute infrastructure does not allow us to wait for the successive iterates of the value function to converge to the true value function V^{π} given by $V^{\pi} = (I - \gamma P)^{-1} R$. Instead, we let the algorithm terminate at time step $k+1$ when the distance between the sucessive iterates given by $\|V_{k+1} - V_k\|_{\infty} \leq \varepsilon$ for a given $\varepsilon > 0$.

(a) Prove that the error estimate between the obtained value function estimate V_{k+1} and true value function V^{π} is given by

$$\|V_{k+1} - V^{\pi}\|_{\infty} \leq \frac{\varepsilon\gamma}{1-\gamma}$$

(5 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 iterative policy evaluation algorithm converges geometrically, i.e.

$$\|V_{k+1} - V^\pi\|_\infty \leq \gamma^k \|V_1 - V_\infty^\pi\|$$

(5 Points)

We already proved that,

$$\|V_{k+1} - V\|_\infty \leq \gamma \|V_k - V\|_\infty$$

,

Applying this inequality recursively, we get the desired result

(c) Let v denote a value function and consider the Bellman optimallity operator given by,

$$L(v) = \max_{a \in \mathcal{A}} \left[\mathcal{R}^a + \gamma \mathcal{P}^a v\right].$$

Prove that the Bellman optimality operator ($\mathcal{L}$) 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 $\mathcal{L}(u) \leq \mathcal{L}(v)$ (3 Points)

By the defintion of Bellman optimality operator $\mathcal{L}$, for a fixed state $s \in \mathcal{S}$ and for action set $\mathcal{A}$ finite, one can conlude that there exists $a_1, a_2 \in \mathcal{A}$ (with a_1 and a_2 possibly different) such that,

$$L(u(s)) = \left[\mathcal{R}(s, a_1) + \gamma \sum_{s'} P(s'|s, a_1) u(s)\right]$$

and

$$L(v(s)) = \left[\mathcal{R}(s, a_2) + \gamma \sum_{s'} P(s'|s, a_2) v(s)\right]$$

It is then easy to observe (using the defintion of optimality operator) that,

$$L(v(s)) \geq \left[\mathcal{R}(s, a_1) + \gamma \sum_{s'} P(s'|s, a_1) v(s)\right]$$

Now, we have,

$$L(u(s)) - L(v(s)) \leq \left[\gamma \sum_{s'} P(s'|s, a_1) \left(u(s) -)v(s)\right)\right]$$

Since, $u(s) \leq v(s)$, we have,

$$L(u(s)) - L(v(s)) \leq \left[\gamma \sum_{s'} P(s'|s, a_1) \left(u(s) -)v(s)\right)\right] \leq \left[\gamma \sum_{s'} P(s'|s, a_1) \left(v(s) - v(s)\right)\right] = 0$$

Since s was choosen arbitarily, we have the desired result.

Problem 6 : On Contractions

(a) Let P and Q be two contractions defined on a normed vector space $< \mathcal{V}, \|\cdot\| >$. Prove that the compositions $P \circ Q$ and $Q \circ P$ are contractions on the same normed vector space. (5 Points)

Indeed, for all , $v, u \in \mathcal{V}$, we have,

$$\|P \circ Q(v) - P \circ Q(u)\| = \|P(Q(v)) - P(Q(u))\| \leq \gamma_p \|Q(v) - Q(u)\| \leq \gamma_p \gamma_q \|v - u\|$$

It is importrant to note that since γ_p and γ_q belong to $[0, 1)$, hence the product $\gamma_p \gamma_q$ also belong to $[0, 1)$.

(b) What can be suitable contraction (or LIpschitz) coeffecients for the contractions $P \circ Q$ and $Q \circ P$? (1 point)

$\gamma_p \gamma_q$ is the contraction co-efficient

(c) Define operator $\mathcal{B}$ as $\mathcal{F} \circ \mathcal{L}$ where $\mathcal{L}$ is the Bellman optimality operator and $\mathcal{F}$ is any other suitable operator. For example, $\mathcal{F}$ could play the role of a function approximator to the Bellman backup $\mathcal{L}$. Under what conditions would the value iteration algorithm converge to a unique solution if operator $\mathcal{B}$ is used in place of $\mathcal{L}$ (in the value iteration algorithm) ? Explain your answer. (2 Points)

For unique solution to exist, the operator $\mathcal{F} \circ \mathcal{L}$ must be a contraction. This composition can be contraction only when both $\mathcal{F}$ and $\mathcal{L}$ is a contraction under the max-norm.

Problem 7 : Programming Value and Policy Iteration

Implement value and policy iteration algorithm and test it on **'Frozen Lake'** environment in openAI gym. **'Frozen Lake'** is a grid-world like environment available in gym. The purpose of this exercise is to to help you get hands on with using gym and to understand the implementation details of value and policy iteration algorithm(s)

This question will not be graded but **will** still come in handy for future assignments.

ALL THE BEST