CSE564: Reinforcement Learning

Assignment-2

Divyajeet Singh (2021529)

September 23, 2023

Question-1

Consider ten bandit arms numbered $1, 2, \ldots, 10$. You pick an even numbered arm with probability twice that of picking an odd arm. Arm i gives a reward that is drawn from $N[i, 1]$. You pick an arm with this policy 10 times. Derive from first principles the expected sum reward at the end of picking ten times.

Solution

According to the given policy, say π, let the probability of picking an odd arm be p. Then,

$$5p + 5 \cdot 2p = 1 \implies p = \frac{1}{15}$$

$$P[A_t = a] = \pi(a) = \begin{cases} \frac{2}{15} & \text{if } a \bmod 2 = 0 \\ \frac{1}{15} & \text{if } a \bmod 2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

The expected sum reward at the end of picking ten times is given by,

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{10} R_t\right] &= \sum_{t=1}^{10} \mathbb{E}[R_t] \\
&= \sum_{t=1}^{10}\sum_{a=1}^{10} \mathbb{E}[R_t | A_t = a] P[A_t = a] \\
&= \sum_{t=1}^{10}\sum_{a=1}^{10} \pi(a)\mathbb{E}[N[a, 1]] = \sum_{t=1}^{10}\sum_{a=1}^{10} a\pi(a) \\
&= \sum_{t=1}^{10} \left[\frac{1}{15} \cdot (1 + 3 + 5 + 7 + 9) + \frac{2}{15} \cdot (2 + 4 + 6 + 8 + 10)\right] \\
&= \sum_{t=1}^{10} \left[\frac{25}{15} + \frac{60}{15}\right] = \frac{85}{15} \cdot 10 = \frac{170}{3} \approx 56.667
\end{aligned}$$

Question-2

We have 10 arms. Arms 1, 2, 4, 5, 7, 9, and 10 give a reward 0 with probability 0.5 and reward of 1 otherwise. The other arms give a reward of 0 with probability 0.3, a reward of 0.2 with probability 0.3, and a reward of 1 with probability 0.4. As always, you want to maximize the expected reward. Derive six optimal policies.

Solution

Let sets $A_1 = \{1,2,4,5,7,9\}$ and $A_2 = \{3,6,8\}$. Then, given the information about the rewards from each arm, we have

$$P[R=r|A \in A_1] = \begin{cases} 0.5 & \text{if } r = 0,1 \\ 0 & \text{otherwise} \end{cases} \qquad \implies \mathbb{E}[R|A \in A_1] = 0.5$$

$$P[R=r|A \in A_2] = \begin{cases} 0.3 & \text{if } r = 0, 0.2 \\ 0.4 & \text{if } r = 1 \\ 0 & \text{otherwise} \end{cases} \qquad \implies \mathbb{E}[R|A \in A_2] = 0.46$$

It is easy to observe that the expected reward is more if the arms in A_1 are picked. So, we can derive the following (non-exhaustive) optimal policies.

$$P[A=a] = \begin{cases} 0.5 & \text{if } a = 1,2 \\ 0 & \text{otherwise} \end{cases} \qquad P[A=a] = \begin{cases} 0.5 & \text{if } a = 2,4 \\ 0 & \text{otherwise} \end{cases}$$

$$P[A=a] = \begin{cases} 0.5 & \text{if } a = 4,5 \\ 0 & \text{otherwise} \end{cases} \qquad P[A=a] = \begin{cases} 0.5 & \text{if } a = 5,7 \\ 0 & \text{otherwise} \end{cases}$$

$$P[A=a] = \begin{cases} 0.5 & \text{if } a = 7,9 \\ 0 & \text{otherwise} \end{cases} \qquad P[A=a] = \begin{cases} 0..5 & \text{if } a = 9,1 \\ 0 & \text{otherwise} \end{cases}$$

In fact, any policy π_* that picks only the actions in A_1 with non-zero probability is an optimal policy.

Question-3

Consider a variant of the ϵ-greedy policy wherein we explore only over the non-greedy actions. Suppose that we have 3 bandit arms. Each arm gives a random reward that takes a value from the set $\{0,1\}$. Assume an initial estimate of $Q_1(a), a \in \{1,2,3\}$. Choose the initial estimates to be different and non-zero for the arms. Create and explain an example sequence of $Q_t(a)$, A_t, and R_t, for $t = 1,2,3,4,5,6$, obtained under the assumption that you explore at odd times and exploit at even times. Use sample mean estimates.

Solution

An example sequence of $Q_t(a)$, A_t, and R_t is given in the Table 1. It was assumed that the initial estimates of the arms are 1, -3, and 4 respectively. As estimates are obtained for the arm, the inital estimates are discarded and then the sample mean estimates are used. The arm with the highest estimate at $t = t_0$ is used for exploitation at $t = t_0 + 1$, i.e. $a = \arg\max_a Q_{t_0}(a)$ is used as the greedy arm. The other two arms are used for exploration in the odd time steps.

t	**Action**	A_t	R_t	$Q_t(1)$	$Q_t(2)$	$Q_t(3)$
$t=1$	-	-	-	1	-3	4
$t=2$	`Exploit`	3	0	1	-3	0
$t=3$	`Explore`	3	1	1	-3	$\frac{1+0}{2} = 0.5$
$t=4$	`Exploit`	1	1	$\frac{1+1}{2} = 1$	-3	0.5
$t=5$	`Explore`	2	0	1	0	0.5
$t=6$	`Exploit`	1	0	$\frac{1+1+0}{3} = \frac{2}{3}$	0	0.5

Table 1: Example sequence of $Q_t(a)$, A_t, and R_t

Question-4

Solve Exercise 3.4 and explain how you obtained the table.

Solution

According to the MDP table given in Example 3.3, the solution to Exercise 3.4 is Table 2. It was obtained by observing the probabilities of each state transition (s, a) to a state s' with reward r in the given MDP. In the table in Example 3.3, the same values can be seen as we fixed (s, a, s') tuples, which could only give rewards from a fixed set.

s	a	s'	r	$p(s', r \mid s, a)$
`high`	`search`	`high`	$r_{\texttt{search}}$	α
`high`	`search`	`low`	$r_{\texttt{search}}$	$1 - \alpha$
`high`	`wait`	`high`	$r_{\texttt{wait}}$	1
`low`	`search`	`high`	-3	$1 - \beta$
`low`	`search`	`low`	$r_{\texttt{search}}$	β
`low`	`wait`	`low`	$r_{\texttt{wait}}$	1
`low`	`recharge`	`high`	0	1

Table 2: The MDP of state-transition probabilities for the Cleaning Robot Problem

This has the same effects of renaming the columns $r(s, a, s')$ to r and $p(s'|s, a)$ to $p(s', r|s, a)$, because in this simple problem, there are no other rewards that can be obtained from a (s, a, s') tuple, i.e., only a deterministic reward is obtained when transitioning from a state s to another state s' using action a.

Question-5

Write code that solves the linear equations required to find $v_\pi(s)$ and generate the values in the table in Figure 3.2. Note that the policy π picks all valid actions in a state with equal probability. Add comments in your code that explain all the steps.

Solution

The solution to the linear equations required to find $v_\pi(s)$ can be found using the dynamic programming policy evaluation technique. The solution for this is given in `main.ipynb`.

Question-6

Solve Exercises 3.15 and 3.16.

Solution

Exercise 3.15

In the Grid-World problem, the signs of the rewards are not relevant. It is the intervals between the *good* rewards and *bad* rewards that matter. According to (3.8) in the text,

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Now we add a constant c to all rewards, then our new discounted return G'_t becomes

$$\begin{aligned} G'_t = \sum_{k=0}^{\infty} \gamma^k \cdot (R_{t+k+1} + c) &= G_t + \sum_{k=0}^{\infty} \gamma^k c \\ &= G_t + c \sum_{k=0}^{\infty} \gamma^k = G_t + c \cdot \frac{1}{1-\gamma} = G_t + v_c \end{aligned}$$

This proves that adding a constant to all returns adds a constant value to the return value. The value of this constant, v_c, in terms of c and γ is

$$v_c = \frac{c}{1-\gamma}$$

This can be used to calculate the value function for all states (in the Grid-World problem). For all $s \in \mathcal{S}$,

$$v'_\pi(s) = \mathbb{E}[G'_t | S_t = s] = \mathbb{E}[G_t + v_c | S_t = s] = \mathbb{E}[G_t | S_t = s] + v_c = v_\pi(s) + v_c$$

Hence, even the values get shifted by the same constant. Since the policy depends on these state-values and the relative difference among them is the same, the policy behaves in the same way. This reinforces the claim that the signs of the rewards are not important.

Exercise 3.16

Adding a constant c to the rewards in an episodic task, i.e. without discounting can very well change the learning task. Negative rewards are used to penalize being in an undersirable state. A negative reward also suggests the agent to try and accelerate finishing the task. This is because a negative reward lowers the expected return, but even a small positive reward increases it.
Let's say two policies π_1 and π_2 with the same total reward generate episodes of length L_1 and L_2 each such that $L_1 > L_2$. Adding a constant c to all rewards would increase the return of π_1 by $L_1 c$, which is more than the increase $L_2 c$ in π_2. This would make our agent prefer the policy that generates longer runs. So, the sign of the rewards in an episodic task is important.

Question-7

Write code that generates the optimal state-value function and the optimal policy for Figure 3.5. You want to solve the corresponding system of non-linear equations. Explain all the steps.

Solution

To solve the non-linear equations, we use the policy iteration technique. The solution for this is given in `main.ipynb`. Within each iteration, the current policy is evaluated by finding its state-values, and then improve upon the policy in a greedy way. This is repeated until the policy converges to the optimal policy.

Question-8

Give an equation for v_* in terms of q_*.

Solution

By definition, the optimal value of a state is given by the maximum of optimal values of all the actions possible in that state. Thus,

$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_*(s,a) \quad \forall s \in \mathcal{S}$$

Question-9

Code policy iteration (PI) and value iteration (VI) to solve the Grid-World problem in Example 4.1. Your code must log the output of each iteration. Pick up a few sample iterations to show policy evaluation and PI at work. Similarly, show using a few obtained iterations that every iteration of VI improves the value function. Your code must include the fix to the bug mentioned in Exercise 4.4.

Solution

The bug in the given pseudocode was that it may never converge. This was because it may get stuck shuffling between two equally optimal policies, which is likely to occur in practice. To fix this, we can fix which of the equally optimal actions to select in each state. For example, instead of breaking ties arbitrarily, we can always select the action that has the smallest index.
Exactly this fix was used in the solution using `numpy.argmax()`, which returns the index of the first occurrence of the maximum value in the array. The solution for this is given in the code in `main.ipynb`.

Question-10

Code Exercise 4.7.

Solution

The solution for Jack's Car Rental problem is given in `main.ipynb`. The solution uses the policy iteration technique to find the optimal policy for the given problem. The state-values are found by simply summing over all possible next states to calculate the expected value of the return.

Question-11

When we defined a Markov Decision Process, we explicitly captured, using probability mass functions, the fact that the random variable R_{t+1} is dependent on the state S_t and the action A_t. Is the random variable R_{t+2} dependent on S_t and A_t? Support your answer using the PMFs used to define an MDP.

Solution

To find out whether R_{t+2} is dependent on S_t and A_t, we consider the conditional PMF of R_{t+2} conditioned on S_t and A_t. Let $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$. Then,

$$\begin{aligned}
P_{R_{t+2}|S_t,A_t}(r'|s,a) &= P[R_{t+2} = r'|S_t = s, A_t = a] \\
&= \sum_{s'} P[R_{t+2} = r', S_{t+1} = s'|S_t = s, A_t = a] \\
&= \sum_{s'} P[R_{t+2} = r'|S_{t+1} = s', S_t = s, A_t = a] P[S_{t+1} = s'|S_t = s, A_t = a]
\end{aligned}$$

By Markov property, S_{t+1} is a complete description required at time $t+1$, and any more information is redundant. Thus, we follow

$$\begin{aligned}
P_{R_{t+2}|S_t,A_t}(r'|s,a) &= \sum_{s'} P[R_{t+2} = r'|S_{t+1} = s'] P[S_{t+1} = s'|S_t = s, A_t = a] \\
&= \sum_{s'} P[R_{t+2} = r'|S_{t+1} = s'] \sum_{r} P[S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a] \\
&= \sum_{s',r} P[R_{t+2} = r'|S_{t+1} = s'] p(s', r|s, a) \\
&= \sum_{s',r} \sum_{s''} P[R_{t+2} = r', S_{t+2} = s''|S_{t+1} = s'] p(s', r|s, a)
\end{aligned}$$

$$= \sum_{s'',s',r} \sum_{a'} P[R_{t+2} = r', S_{t+2} = s'', A_{t+1} = a' | S_{t+1} = s'] p(s', r | s, a)$$
$$= \sum_{s'',s',a',r} P[R_{t+2} = r', S_{t+2} = s'' | S_{t+1} = s', A_{t+1} = a'] P[A_{t+1} = a' | S_{t+1} = s'] p(s', r | s, a)$$
$$= \sum_{s'',s',a',r} p(s'', r' | s', a') \pi(a' | s') p(s', r | s, a)$$

By the above equations[1], it is easy to notice that R_{t+2} is dependent on A_{t+1} and S_{t+1}, which in turn are dependent on A_t and S_t. So, R_{t+2} is dependent on A_t and S_t.

Question-12

Derive the expression for $\mathbb{E}[R_{t+2} | S_t = s, A_t = a]$ in terms of the PMF(s) that define an MDP.

Solution

We use first principles to find the expected value of R_{t+2} conditioned on $S_t = s$ and $A_t = a$.

$$\mathbb{E}[R_{t+2} | S_t = s, A_t = a] = \sum_{r'} r' P[R_{t+2} = r' | S_t = s, A_t = a]$$

Using the conditional PMF of R_{t+2} conditioned on $S_t = s$ and $A_t = a$ from the final result of Question-11, the expectation becomes

$$\mathbb{E}[R_{t+2} | S_t = s, A_t = a] = \sum_{s'',s',a',r',r} r' p(s'', r' | s', a') \pi(a' | s') p(s', r | s, a)$$

Question-13

We know that the state-value function $v_\pi(s) = \mathbb{E}[G_t | S_t = s]$. Use this definition of $v_\pi(s)$ to derive the Bellman equation for $v_\pi(s)$ for all $s \in \mathcal{S}$. The Bellman equation will use the PMF corresponding to the policy π and the PMF $p(s', r | s, a)$ and will provide a recursive method of calculating $v_\pi(s)$.

Solution

We start with the given equation and derive the recursive form of the Bellman equation.

$$\begin{aligned}
v_\pi(s) &= \mathbb{E}[G_t | S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma G_{t+1} | S_t = s] \\
&= \mathbb{E}[R_{t+1} | S_t = s] + \gamma \mathbb{E}[G_{t+1} | S_t = s] \\
&= \mathbb{E}[R_{t+1} | S_t = s] + \gamma \sum_{g'} g' P[G_{t+1} = g' | S_t = s] \\
&= \mathbb{E}[R_{t+1} | S_t = s] + \gamma \sum_{g',s'} g' P[G_{t+1} = g', S_{t+1} = s' | S_t = s] \\
&= \mathbb{E}[R_{t+1} | S_t = s] + \gamma \sum_{g',s'} g' P[G_{t+1} = g' | S_{t+1} = s'] P[S_t = s | S_t = s'] \\
&= \mathbb{E}[R_{t+1} | S_t = s] + \gamma \sum_{s'} v_\pi(s') P[S_t = s' | S_t = s] \\
&= \mathbb{E}[R_{t+1} | S_t = s] + \gamma \mathbb{E}[v_\pi(S_{t+1}) | S_t = s]
\end{aligned}$$

[1] **Abuse of Notation:** It is implicit that $s', s'' \in \mathcal{S}$, $a' \in \mathcal{A}(s')$, and $r, r' \in \mathcal{R}$. Moreover, the summations are over all possible values of the variables under them.

We again get a sum of two scaled expectations. So, we get

$$\begin{aligned}
v_\pi(s) &= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s] \\
&= \sum_{s',r} \left[r + \gamma v_\pi(s')\right] P[S_{t+1} = s', R_{t+1} = r|S_t = s] \\
&= \sum_{s',r,a} \left[r + \gamma v_\pi(s')\right] P[S_{t+1} = s', R_{t+1} = r, A_t = a|S_t = s] \\
&= \sum_{s',r,a} \left[r + \gamma v_\pi(s')\right] P[S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a]P[A_t = a|S_t = s] \\
&= \sum_{s',r,a} \left[r + \gamma v_\pi(s')\right] p(s', r|s, a)\pi(a|s)
\end{aligned}$$

The above equations hold true for each state $s \in \mathcal{S}$.

Question-14

Suppoes an agent receives the sequence of rewards $R_1 = 2$, $R_2 = -1$, $R_3 = 10$, and $R_4 = -3$. Calculate the γ-discounted return/reward for each step for $\gamma = 0.5$. Also, show that, if agent receives a constant reward c, at every time step, for $\gamma < 1$, the infinite horizon discounted return is given by

$$G_t = \frac{c}{1 - \gamma}$$

Solution

We start from the last received reward. The γ-discounted return/reward for each step is given by

$$\begin{aligned}
G_3 &= R_4 = -3 \\
G_2 &= R_3 + \gamma G_3 = 10 + 0.5 \cdot (-3) = 8.5 \\
G_1 &= R_2 + \gamma G_2 = -1 + 0.5 \cdot 8.5 = 3.25 \\
G_0 &= R_1 + \gamma G_1 = 2 + 0.5 \cdot 3.25 = 3.625
\end{aligned}$$

Now, we assume that the agent receives a constant reward c at every time step, given that $\gamma < 1$. Then, the infinite horizon discounted return is given by

$$\begin{aligned}
G_t &= R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \\
&= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = c \sum_{k=0}^{\infty} \gamma^k = \frac{c}{1 - \gamma}
\end{aligned}$$

This proves the claim.

Question-15

Suppose you are given the optimal state-value function $v_*(s)$, for all $s \in \mathcal{S}$. How will you find the optimal policy? Show your steps.

Solution

An optimal policy, π_*, is a policy that always picks the action with the highest value in each state. So, to pick an action using the state values, we can use the following equation.

$$\begin{aligned}
\pi_*(s) &= \underset{a \in \mathcal{A}(s)}{\text{argmax}}\ q_*(s, a) \\
&= \underset{a \in \mathcal{A}(s)}{\text{argmax}}\ \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1})|S_t = s, A_t = a] \\
&= \underset{a \in \mathcal{A}(s)}{\text{argmax}} \sum_{s',r} \left[r + \gamma v_*(s')\right] p(s', r|s, a)
\end{aligned}$$

where $q_*(s,a)$ is the optimal action-value function.

Question-16

Problem too long to dump here.

Solution

1

The set of states $\mathcal{S} = \{\texttt{fresh}, \texttt{stale}\}$ and actions $\mathcal{A} = \{\texttt{query}, \texttt{stay silent}\}$. So, the given MDP can be tabulated as follows.

s	a	s'	r	$p(s', r \mid s, a)$
fresh	stay	fresh	+4	0.5
fresh	stay	stale	+4	0.5
fresh	query	fresh	-4	0.9
fresh	query	stale	-4	0.1
stale	stay	fresh	-	0.0
stale	stay	stale	+4	1.0
stale	query	fresh	-8	0.8
stale	query	stale	-8	0.2

Table 3: The MDP table for the Server Query Problem

2

The future rewards are discounted with a factor of $\gamma = 0.5$. The agent gets to optimize over three time steps. Given the three time steps, we can calculate the optimal state values for each state for the time steps $t = 0, 1, 2, 3$ in reverse order. Let $v_{t_*}(s)$ denote the optimal state value for state s at time t. For brevity, we will also assume

$$\text{State } \texttt{S}_\texttt{F} = \texttt{fresh} \qquad \text{Action } \texttt{A}_\texttt{Q} = \texttt{query}$$
$$\text{State } \texttt{S}_\texttt{S} = \texttt{stale} \qquad \text{Action } \texttt{A}_\texttt{S} = \texttt{stay}$$

At the terminal step $t = 3$, if the agent ends in $\texttt{S}_\texttt{F}$, then it gets a reward of +10, and if it ends in $\texttt{S}_\texttt{S}$, then it gets a reward of -10. So,

$$v_{3_*}(\texttt{S}_\texttt{F}) = +10 \qquad v_{3_*}(\texttt{S}_\texttt{S}) = -10$$

For the other time steps, we repeatedly make use of the following formula

$$v_{t_*}(s) = \max_{a \in \mathcal{A}(s)} q_t(s,a) \qquad \pi_{t_*}(s) = \operatorname*{argmax}_{a \in \mathcal{A}(s)} q_t(s,a)$$

where we can calculate $q_t(s,a)$ as follows

$$\begin{aligned}
q_t(s,a) &= \sum_{s',r} \left[r + \gamma v_{t+1_*}(s') \right] p(s', r \mid s, a) \\
&= \sum_r r \; p(r \mid s, a) + \gamma \sum_{s'} v_t(s') \; p(s' \mid s, a) \\
&= r(s,a) + 0.5 \cdot \left[v_{t+1_*}(\texttt{S}_\texttt{F}) \; p(\texttt{S}_\texttt{F} \mid s, a) + v_{t+1_*}(\texttt{S}_\texttt{S}) \; p(\texttt{S}_\texttt{S} \mid s, a) \right]
\end{aligned}$$

So, for time $t = 2$, we have

$$\begin{aligned}
\pi_{2_*}(\mathtt{S}_\mathtt{F}) &= \operatorname*{argmax}_{a \in \mathcal{A}(\mathtt{S}_\mathtt{F})} \begin{cases} \mathtt{A}_\mathtt{Q} : & -4 + 0.5 \cdot \big[v_{3_*}(\mathtt{S}_\mathtt{F}) \cdot 0.9 + v_{3_*}(\mathtt{S}_\mathtt{S}) \cdot 0.1\big] = 0.0 \\ \mathtt{A}_\mathtt{S} : & +4 + 0.5 \cdot \big[v_{3_*}(\mathtt{S}_\mathtt{F}) \cdot 0.5 + v_{3_*}(\mathtt{S}_\mathtt{S}) \cdot 0.5\big] = 4.0 \end{cases} \\
&= \mathtt{A}_\mathtt{S} & \implies v_{2_*}(\mathtt{S}_\mathtt{F}) = 4.0 \\
\pi_{2_*}(\mathtt{S}_\mathtt{S}) &= \operatorname*{argmax}_{a \in \mathcal{A}(\mathtt{S}_\mathtt{S})} \begin{cases} \mathtt{A}_\mathtt{Q} : & -8 + 0.5 \cdot \big[v_{3_*}(\mathtt{S}_\mathtt{F}) \cdot 0.8 + v_{3_*}(\mathtt{S}_\mathtt{S}) \cdot 0.2\big] = -5.0 \\ \mathtt{A}_\mathtt{S} : & +4 + 0.5 \cdot \big[v_{3_*}(\mathtt{S}_\mathtt{F}) \cdot 0.0 + v_{3_*}(\mathtt{S}_\mathtt{S}) \cdot 1.0\big] = -1.0 \end{cases} \\
&= \mathtt{A}_\mathtt{S} & \implies v_{2_*}(\mathtt{S}_\mathtt{S}) = -1.0
\end{aligned}$$

For time $t = 1$,

$$\begin{aligned}
\pi_{1_*}(\mathtt{S}_\mathtt{F}) &= \operatorname*{argmax}_{a \in \mathcal{A}(\mathtt{S}_\mathtt{F})} \begin{cases} \mathtt{A}_\mathtt{Q} : & -4 + 0.5 \cdot \big[v_{2_*}(\mathtt{S}_\mathtt{F}) \cdot 0.9 + v_{2_*}(\mathtt{S}_\mathtt{S}) \cdot 0.1\big] = -2.25 \\ \mathtt{A}_\mathtt{S} : & +4 + 0.5 \cdot \big[v_{2_*}(\mathtt{S}_\mathtt{F}) \cdot 0.5 + v_{2_*}(\mathtt{S}_\mathtt{S}) \cdot 0.5\big] = 4.75 \end{cases} \\
&= \mathtt{A}_\mathtt{S} & \implies v_{1_*}(\mathtt{S}_\mathtt{F}) = 4.75 \\
\pi_{1_*}(\mathtt{S}_\mathtt{S}) &= \operatorname*{argmax}_{a \in \mathcal{A}(\mathtt{S}_\mathtt{S})} \begin{cases} \mathtt{A}_\mathtt{Q} : & -8 + 0.5 \cdot \big[v_{2_*}(\mathtt{S}_\mathtt{F}) \cdot 0.8 + v_{2_*}(\mathtt{S}_\mathtt{S}) \cdot 0.2\big] = -6.5 \\ \mathtt{A}_\mathtt{S} : & +4 + 0.5 \cdot \big[v_{2_*}(\mathtt{S}_\mathtt{F}) \cdot 0.0 + v_{2_*}(\mathtt{S}_\mathtt{S}) \cdot 1.0\big] = 3.5 \end{cases} \\
&= \mathtt{A}_\mathtt{S} & \implies v_{1_*}(\mathtt{S}_\mathtt{S}) = 3.5
\end{aligned}$$

Finally, for time $t = 0$,

$$\begin{aligned}
\pi_{0_*}(\mathtt{S}_\mathtt{F}) &= \operatorname*{argmax}_{a \in \mathcal{A}(\mathtt{S}_\mathtt{F})} \begin{cases} \mathtt{A}_\mathtt{Q} : & -4 + 0.5 \cdot \big[v_{1_*}(\mathtt{S}_\mathtt{F}) \cdot 0.9 + v_{1_*}(\mathtt{S}_\mathtt{S}) \cdot 0.1\big] = -1.6875 \\ \mathtt{A}_\mathtt{S} : & +4 + 0.5 \cdot \big[v_{1_*}(\mathtt{S}_\mathtt{F}) \cdot 0.5 + v_{1_*}(\mathtt{S}_\mathtt{S}) \cdot 0.5\big] = 6.0625 \end{cases} \\
&= \mathtt{A}_\mathtt{S} & \implies v_{0_*}(\mathtt{S}_\mathtt{F}) = 6.0625 \\
\pi_{0_*}(\mathtt{S}_\mathtt{S}) &= \operatorname*{argmax}_{a \in \mathcal{A}(\mathtt{S}_\mathtt{S})} \begin{cases} \mathtt{A}_\mathtt{Q} : & -8 + 0.5 \cdot \big[v_{1_*}(\mathtt{S}_\mathtt{F}) \cdot 0.8 + v_{1_*}(\mathtt{S}_\mathtt{S}) \cdot 0.2\big] = -5.75 \\ \mathtt{A}_\mathtt{S} : & +4 + 0.5 \cdot \big[v_{1_*}(\mathtt{S}_\mathtt{F}) \cdot 0.0 + v_{1_*}(\mathtt{S}_\mathtt{S}) \cdot 1.0\big] = 5.75 \end{cases} \\
&= \mathtt{A}_\mathtt{S} & \implies v_{0_*}(\mathtt{S}_\mathtt{S}) = 5.75
\end{aligned}$$

3

Now, we need to show upto 4 iterations of value iteration. The update rule for value iteration is given by

$$v_{k+1}(s) = \max_{a \in \mathcal{A}(s)} \sum_{s',r} \big[r + \gamma v_k(s')\big]\, p(s', r \mid s, a)$$

Let us assume the initial values of the states to be 0. Then, for iteration 1

$$\begin{aligned}
v_1(\mathtt{S}_\mathtt{F}) &= \max_{a \in \mathcal{A}(\mathtt{S}_\mathtt{F})} \begin{cases} \mathtt{A}_\mathtt{Q} : & -4 + 0.5 \cdot \big[v_0(\mathtt{S}_\mathtt{F}) \cdot 0.9 + v_0(\mathtt{S}_\mathtt{S}) \cdot 0.1\big] = -4.0 \\ \mathtt{A}_\mathtt{S} : & +4 + 0.5 \cdot \big[v_0(\mathtt{S}_\mathtt{F}) \cdot 0.5 + v_0(\mathtt{S}_\mathtt{S}) \cdot 0.5\big] = 4.0 \end{cases} \\
&= 4.0 \\
v_1(\mathtt{S}_\mathtt{S}) &= \max_{a \in \mathcal{A}(\mathtt{S}_\mathtt{S})} \begin{cases} \mathtt{A}_\mathtt{Q} : & -8 + 0.5 \cdot \big[v_0(\mathtt{S}_\mathtt{F}) \cdot 0.8 + v_0(\mathtt{S}_\mathtt{S}) \cdot 0.2\big] = -8.0 \\ \mathtt{A}_\mathtt{S} : & +4 + 0.5 \cdot \big[v_0(\mathtt{S}_\mathtt{F}) \cdot 0.0 + v_0(\mathtt{S}_\mathtt{S}) \cdot 1.0\big] = 4.0 \end{cases} \\
&= 4.0
\end{aligned}$$

For iteration 2,

$$\begin{aligned}
v_2(\mathtt{S}_\mathtt{F}) &= \max_{a \in \mathcal{A}(\mathtt{S}_\mathtt{F})} \begin{cases} \mathtt{A}_\mathtt{Q} : & -4 + 0.5 \cdot \big[v_1(\mathtt{S}_\mathtt{F}) \cdot 0.9 + v_1(\mathtt{S}_\mathtt{S}) \cdot 0.1\big] = -2.0 \\ \mathtt{A}_\mathtt{S} : & +4 + 0.5 \cdot \big[v_1(\mathtt{S}_\mathtt{F}) \cdot 0.5 + v_1(\mathtt{S}_\mathtt{S}) \cdot 0.5\big] = 6.0 \end{cases} \\
&= 6.0 \\
v_2(\mathtt{S}_\mathtt{S}) &= \max_{a \in \mathcal{A}(\mathtt{S}_\mathtt{S})} \begin{cases} \mathtt{A}_\mathtt{Q} : & -8 + 0.5 \cdot \big[v_1(\mathtt{S}_\mathtt{F}) \cdot 0.8 + v_1(\mathtt{S}_\mathtt{S}) \cdot 0.2\big] = -6.0 \\ \mathtt{A}_\mathtt{S} : & +4 + 0.5 \cdot \big[v_1(\mathtt{S}_\mathtt{F}) \cdot 0.0 + v_1(\mathtt{S}_\mathtt{S}) \cdot 1.0\big] = 6.0 \end{cases} \\
&= 6.0
\end{aligned}$$

For iteration 3,

$$\begin{aligned} v_3(\mathtt{S_F}) = &\max_{a\in\mathcal{A}(\mathtt{S_F})} \begin{cases} \mathtt{A_Q}: & -4 + 0.5\cdot\big[v_2(\mathtt{S_F})\cdot 0.9 + v_2(\mathtt{S_S})\cdot 0.1\big] = -1.0 \\ \mathtt{A_S}: & +4 + 0.5\cdot\big[v_2(\mathtt{S_F})\cdot 0.5 + v_2(\mathtt{S_S})\cdot 0.5\big] = 7.0 \end{cases} \\ = &\ 7.0 \\ v_3(\mathtt{S_S}) = &\max_{a\in\mathcal{A}(\mathtt{S_S})} \begin{cases} \mathtt{A_Q}: & -8 + 0.5\cdot\big[v_2(\mathtt{S_F})\cdot 0.8 + v_2(\mathtt{S_S})\cdot 0.2\big] = -5.0 \\ \mathtt{A_S}: & +4 + 0.5\cdot\big[v_2(\mathtt{S_F})\cdot 0.0 + v_2(\mathtt{S_S})\cdot 1.0\big] = 7.0 \end{cases} \\ = &\ 7.0 \end{aligned}$$

Finally, for iteration 4,

$$\begin{aligned} v_4(\mathtt{S_F}) = &\max_{a\in\mathcal{A}(\mathtt{S_F})} \begin{cases} \mathtt{A_Q}: & -4 + 0.5\cdot\big[v_2(\mathtt{S_F})\cdot 0.9 + v_2(\mathtt{S_S})\cdot 0.1\big] = -0.5 \\ \mathtt{A_S}: & +4 + 0.5\cdot\big[v_2(\mathtt{S_F})\cdot 0.5 + v_2(\mathtt{S_S})\cdot 0.5\big] = 7.5 \end{cases} \\ = &\ 7.5 \\ v_4(\mathtt{S_S}) = &\max_{a\in\mathcal{A}(\mathtt{S_S})} \begin{cases} \mathtt{A_Q}: & -8 + 0.5\cdot\big[v_2(\mathtt{S_F})\cdot 0.8 + v_2(\mathtt{S_S})\cdot 0.2\big] = -4.5 \\ \mathtt{A_S}: & +4 + 0.5\cdot\big[v_2(\mathtt{S_F})\cdot 0.0 + v_2(\mathtt{S_S})\cdot 1.0\big] = 7.0 \end{cases} \\ = &\ 7.5 \end{aligned}$$

We can see that the values are converging. We must return an optimal policy at the end of value iteration. Clearly,

$$\pi_{4_*}(s) = \mathtt{A_S} \quad s = \mathtt{S_F},\ \mathtt{S_S}$$

Now, we do the same thing with policy iteration. We begin with an equiprobable random policy. We evaluate the policy and then improve it. The following is the policy evaluation step for iteration 1.

$$\begin{aligned} v_{\pi_0}(\mathtt{S_F}) &= \pi_0(\mathtt{A_Q} \mid \mathtt{S_F})\cdot\big[-4 + 0.5\cdot\big(v_{\pi_0}(\mathtt{S_F})\cdot 0.9 + v_{\pi_0}(\mathtt{S_S})\cdot 0.1\big)\big] \\ &+ \pi_0(\mathtt{A_S} \mid \mathtt{S_F})\cdot\big[+4 + 0.5\cdot\big(v_{\pi_0}(\mathtt{S_F})\cdot 0.5 + v_{\pi_0}(\mathtt{S_S})\cdot 0.5\big)\big] \\ &= 0.5\cdot\big[-4 + 0.45v_{\pi_0}(\mathtt{S_F}) + 0.05v_{\pi_0}(\mathtt{S_S})\big] + 0.5\cdot\big[+4 + 0.25v_{\pi_0}(\mathtt{S_F}) + 0.25v_{\pi_0}(\mathtt{S_S})\big] \\ &= -2 + 2 + (0.225 + 0.125)v_{\pi_0}(\mathtt{S_F}) + (0.025 + 0.125)v_{\pi_0}(\mathtt{S_S}) \\ &= 0.35v_{\pi_0}(\mathtt{S_F}) + 0.15v_{\pi_0}(\mathtt{S_S}) \\ v_{\pi_0}(\mathtt{S_S}) &= \pi_0(\mathtt{A_Q} \mid \mathtt{S_S})\cdot\big[-8 + 0.5\cdot\big(v_{\pi_0}(\mathtt{S_F})\cdot 0.8 + v_{\pi_0}(\mathtt{S_S})\cdot 0.2\big)\big] \\ &+ \pi_0(\mathtt{A_S} \mid \mathtt{S_S})\cdot\big[+4 + 0.5\cdot\big(v_{\pi_0}(\mathtt{S_F})\cdot 0.0 + v_{\pi_0}(\mathtt{S_S})\cdot 1.0\big)\big] \\ &= 0.5\cdot\big[-8 + 0.4v_{\pi_0}(\mathtt{S_F}) + 0.1v_{\pi_0}(\mathtt{S_S})\big] + 0.5\cdot\big[+4 + 0.5\cdot\big(v_{\pi_0}(\mathtt{S_S})\big)\big] \\ &= -4 + 2 + 0.2v_{\pi_0}(\mathtt{S_F}) + (0.05 + 0.5)v_{\pi_0}(\mathtt{S_S}) \\ &= 0.2v_{\pi_0}(\mathtt{S_F}) + 0.55v_{\pi_0}(\mathtt{S_S}) - 2 \end{aligned}$$

Solving these equations, we get $v_{\pi_0}(\mathtt{S_F}) = -1.1$ and $v_{\pi_0}(\mathtt{S_S}) = -4.9$. Now, we improve the policy

$$\begin{aligned} \pi_1(\mathtt{S_F}) = &\operatorname*{argmax}_{a\in\mathcal{A}(\mathtt{S_F})} \begin{cases} \mathtt{A_Q}: & -4 + 0.5\cdot\big[v_{\pi_0}(\mathtt{S_F})\cdot 0.9 + v_{\pi_0}(\mathtt{S_S})\cdot 0.1\big] = -4.7 \\ \mathtt{A_S}: & +4 + 0.5\cdot\big[v_{\pi_0}(\mathtt{S_F})\cdot 0.5 + v_{\pi_0}(\mathtt{S_S})\cdot 0.5\big] = -2.5 \end{cases} \\ = &\ \mathtt{A_S} \\ \pi_1(\mathtt{S_S}) = &\operatorname*{argmax}_{a\in\mathcal{A}(\mathtt{S_S})} \begin{cases} \mathtt{A_Q}: & -8 + 0.5\cdot\big[v_{\pi_0}(\mathtt{S_F})\cdot 0.8 + v_{\pi_0}(\mathtt{S_S})\cdot 0.2\big] = -8.9 \\ \mathtt{A_S}: & +4 + 0.5\cdot\big[v_{\pi_0}(\mathtt{S_F})\cdot 0.0 + v_{\pi_0}(\mathtt{S_S})\cdot 1.0\big] = 1.5 \end{cases} \\ = &\ \mathtt{A_S} \end{aligned}$$

Now we have a deterministic policy π_1. We evaluate it again

$$\begin{aligned} v_{\pi_1}(\mathtt{S_F}) &= \pi_1(\mathtt{S_F})\cdot\big[+4 + 0.5\cdot\big(v_{\pi_1}(\mathtt{S_F})\cdot 0.5 + v_{\pi_1}(\mathtt{S_S})\cdot 0.5\big)\big] \\ &= 4 + 0.25v_{\pi_1}(\mathtt{S_F}) + 0.25v_{\pi_1}(\mathtt{S_S}) \\ v_{\pi_1}(\mathtt{S_S}) &= \pi_1(\mathtt{S_S})\cdot\big[+4 + 0.5\cdot\big(v_{\pi_1}(\mathtt{S_F})\cdot 0.0 + v_{\pi_1}(\mathtt{S_S})\cdot 1.0\big)\big] \\ &= 4 + 0.5v_{\pi_1}(\mathtt{S_S}) \end{aligned}$$

Solving these equations, we get $v_{\pi_1}(\texttt{S}_\texttt{F}) = 8.0$ and $v_{\pi_1}(\texttt{S}_\texttt{S}) = 8.0$. Also, notice that any deterministic policy that chooses $\pi_k(\texttt{S}_\texttt{S}) = \texttt{A}_\texttt{S}$ will give the same value for $\texttt{S}_\texttt{S}$. Now, we improve the policy

$$\begin{aligned}
\pi_2(\texttt{S}_\texttt{F}) &= \underset{a\in\mathcal{A}(\texttt{S}_\texttt{F})}{\text{argmax}} \begin{cases} \texttt{A}_\texttt{Q}: & -4 + 0.5\cdot\left[v_{\pi_1}(\texttt{S}_\texttt{F})\cdot 0.9 + v_{\pi_1}(\texttt{S}_\texttt{S})\cdot 0.1\right] = 0.0 \\ \texttt{A}_\texttt{S}: & +4 + 0.5\cdot\left[v_{\pi_1}(\texttt{S}_\texttt{F})\cdot 0.5 + v_{\pi_1}(\texttt{S}_\texttt{S})\cdot 0.5\right] = 8.0 \end{cases} \\
&= \texttt{A}_\texttt{S} \\
\pi_2(\texttt{S}_\texttt{S}) &= \underset{a\in\mathcal{A}(\texttt{S}_\texttt{S})}{\text{argmax}} \begin{cases} \texttt{A}_\texttt{Q}: & -8 + 0.5\cdot\left[v_{\pi_1}(\texttt{S}_\texttt{F})\cdot 0.8 + v_{\pi_1}(\texttt{S}_\texttt{S})\cdot 0.2\right] = -4.0 \\ \texttt{A}_\texttt{S}: & +4 + 0.5\cdot\left[v_{\pi_1}(\texttt{S}_\texttt{F})\cdot 0.0 + v_{\pi_1}(\texttt{S}_\texttt{S})\cdot 1.0\right] = 8.0 \end{cases} \\
&= \texttt{A}_\texttt{S}
\end{aligned}$$

We observe that $\pi_2 = \pi_1$. Hence, this is the optimal policy, π_*, which was obtained in two iterations.

Question-17

Assume an infinite horizon discounted costs problem. Prove that the policy impovement step either improves the current policy or the current policy is optimal. Show all your steps and support them using the properties of dynamic programming.

Solution

The above given statement is the Policy Improvement Theorem. Let our current policy be π_k. Then we need to show the following

$$\begin{aligned}
v_{\pi_{k+1}}(s) &\geq v_{\pi_k}(s) \quad \forall s\in\mathcal{S} \\
v_{\pi_{k+1}}(s) &= v_{\pi_k}(s) \quad \forall s\in\mathcal{S} \implies \pi_k = \pi_*
\end{aligned}$$

where π_{k+1} is the next policy, given by policy improvement.
We improve the policy by considering the fact that if it is better to select an action a^* in a state s once, then it is best to select a^* in s every time. According to the policy improvement theorem, we change to policy π_{k+1} because

$$\begin{aligned}
v_{\pi_k}(s) &\leq q_{\pi_k}(s, \pi_{k+1}(s)) \quad \forall s\in\mathcal{S} \\
&= \mathbb{E}[R_{t+1} + \gamma v_{\pi_k}(S_{t+1}) | S_t = s, A_t = \pi_{k+1}(s)] \\
&= \mathbb{E}_{\pi_{k+1}}[R_{t+1} + \gamma v_{\pi_k}(S_{t+1}) | S_t = s]
\end{aligned}$$

Now we use the fact that the state value of any state is at most the value of the best action in that state. So, we have

$$\begin{aligned}
v_{\pi_k}(S_{t+1}) &\leq q_{\pi_k}(S_{t+1}, \pi_{k+1}(S_{t+1})) \\
&= \mathbb{E}[R_{t+2} + \gamma v_\pi(S_{t+2}) | S_{t+1}, A_{t+1} = \pi_{k+1}(S_{t+1})] = \mathbb{E}[\cdots] \\
v_{\pi_k}(s) &\leq \mathbb{E}_{\pi_{k+1}}[R_{t+1} + \gamma q_{\pi_k}(S_{t+1}, \pi_{k+1}(S_{t+1})) | S_t = s] \\
&= \mathbb{E}_{\pi_{k+1}}[R_{t+1} + \gamma \mathbb{E}[\cdots] | S_t = s] \\
&= \mathbb{E}_{\pi_{k+1}}[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_{\pi_k}(S_{t+2}) | S_t = s] \\
&\leq \mathbb{E}_{\pi_{k+1}}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_{\pi_k}(S_{t+3}) | S_t = s] \\
&\leq \mathbb{E}_{\pi_{k+1}}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots | S_t = s] \\
&= \mathbb{E}_{\pi_{k+1}}[G_t | S_t = s] = v_{\pi_{k+1}}(s)
\end{aligned}$$

Therefore, we have shown that the value for all states under policy π_{k+1} is greater than or equal to the value for all states under policy π_k. This proves the first part of the claim. Now, suppose that

$v_{\pi_{k+1}}(s) = v_{\pi_k}(s) \quad \forall s \in \mathcal{S}$. Then, we have

$$\begin{aligned} v_{\pi_{k+1}}(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_k}(s,a) \\ &= \max_{a \in \mathcal{A}(s)} \mathbb{E}[R_{t+1} + \gamma v_{\pi_k}(S_{t+1}) | S_t = s, A_t = a] \\ &= \max_{a \in \mathcal{A}(s)} \mathbb{E}[R_{t+1} + \gamma v_{\pi_{k+1}}(S_{t+1}) | S_t = s, A_t = a] \end{aligned}$$

which is the Bellman Optimality Equation. Hence, if the policy does not improve, it is the optimal policy.

Question-18

Problem too long to dump here.

Solution

The given *Stairwell* problem can be modelled as an MDP as follows. The set of states is $\mathcal{S} = \{\texttt{G}, \texttt{1}, \texttt{2}, \texttt{F}\}$. Since `G` and `F` are terminal states, the set of actions in these states is $\mathcal{A}(\texttt{G}) = \mathcal{A}(F) = \emptyset$. The set of actions in the other states is $\mathcal{A}(\texttt{1}) = \mathcal{A}(\texttt{2}) = \{\uparrow, \downarrow\}$. The MDP is tabulated in Table 4.

s	a	s'	r	$p(s', r \mid s, a)$
1	↑	2	-1	1.0
1	↓	G	+1	1.0
2	↑	F	+4	1.0
2	↓	1	+2	0.5
2	↓	1	0	0.5

Table 4: The MDP table for the *Stairwell* Problem

Moreover, it is given that our initial policy, π_0, chooses to go up or down an equal probability of 0.5. To find the optimal policy, we first find the state values. We use the Bellman equations to form linear equations in the state values. This is the policy evaluation step.

$$v_{\pi_k}(\texttt{G}) = v_{\pi_k}(\texttt{F}) = 0 \quad \forall\ k$$

since the value of terminal states is zero by convention. The other equations are

$$\begin{aligned} v_{\pi_0}(\texttt{1}) &= \pi_0(\uparrow \mid \texttt{1}) \cdot (-1 + v_{\pi_0}(\texttt{2}))\ p(\texttt{2}, -1 | \texttt{1}, \uparrow) + \pi_0(\downarrow \mid \texttt{1}) \cdot (+1 + v_{\pi_0}(\texttt{G}))\ p(\texttt{G}, +1 | \texttt{1}, \downarrow) \\ &= 0.5 \cdot (-1 + v_{\pi_0}(\texttt{2})) + 0.5 \cdot (+1) \\ &= -0.5 + 0.5 + 0.5\ v_{\pi_0}(\texttt{2}) = 0.5\ v_{\pi_0}(\texttt{2}) \\ v_{\pi_0}(\texttt{2}) &= \pi_0(\uparrow \mid \texttt{2}) \cdot (+4 + v_{\pi_0}(\texttt{F}))\ p(\texttt{F}, +4 | \texttt{2}, \uparrow) \\ &+ \pi_0(\downarrow \mid \texttt{2}) \cdot [(+2 + v_{\pi_0}(\texttt{1}))\ p(\texttt{1}, +2 | \texttt{2}, \downarrow) + (0 + v_{\pi_0}(\texttt{1}))\ p(\texttt{1}, 0 | \texttt{2}, \downarrow)] \\ &= 0.5 \cdot (+4) + 0.5 \cdot [(+2 + v_{\pi_0}(\texttt{1})) \cdot 0.5 + (0 + v_{\pi_0}(\texttt{1})) \cdot 0.5] \\ &= 2 + 0.5 \cdot (+1 + v_{\pi_0}(\texttt{1})) \\ &= 2 + 0.5 + 0.5\ v_{\pi_0}(\texttt{1}) = 2.5 + 0.5\ v_{\pi_0}(\texttt{1}) \end{aligned}$$

So, the Bellman equations give us the following linear equations

$$\begin{aligned} v_{\pi_0}(\texttt{1}) &= 0.5\ v_{\pi_0}(\texttt{2}) \\ v_{\pi_0}(\texttt{2}) &= 2.5 + 0.5\ v_{\pi_0}(\texttt{1}) \end{aligned}$$

This system of linear equations can be solved to get $v_{\pi_0}(\mathtt{1}) = 1.667$ and $v_{\pi_0}(\mathtt{2}) = 3.334$. Now, to improve to a policy π_1, we select the action that provides the highest value in each state. This is the policy improvement step.

$$\begin{aligned}
\pi_1(\mathtt{1}) &= \underset{a \in \mathcal{A}(\mathtt{1})}{\operatorname{argmax}} \begin{cases} \uparrow: & -1 + v_{\pi_0}(\mathtt{2}) = 2.334 \\ \downarrow: & +1 + v_{\pi_0}(\mathtt{G}) = 1 \end{cases} \\
&= \uparrow \\
\pi_1(\mathtt{2}) &= \underset{a \in \mathcal{A}(\mathtt{2})}{\operatorname{argmax}} \begin{cases} \uparrow: & +4 + v_{\pi_0}(\mathtt{F}) = 4 \\ \downarrow: & (+2 + v_{\pi_0}(\mathtt{1})) \cdot 0.5 + (0 + v_{\pi_0}(\mathtt{1})) \cdot 0.5 = 2.667 \end{cases} \\
&= \uparrow
\end{aligned}$$

This completes one full cycle of policy iteration. We now use the Bellman equations to find the state values for the new policy π_1. This is again, the policy evaluation step for the new policy, π_1. These are

$$\begin{aligned}
v_{\pi_1}(\mathtt{1}) &= \pi_1(\uparrow \mid \mathtt{1}) \cdot (-1 + v_{\pi_1}(\mathtt{2}))\ p(\mathtt{2}, -1 | \mathtt{1}, \uparrow) \\
&= 1 \cdot (-1 + v_{\pi_1}(\mathtt{2})) + 0 \cdot (+1) \\
&= -1 + v_{\pi_1}(\mathtt{2}) \\
v_{\pi_1}(\mathtt{2}) &= \pi_1(\uparrow \mid \mathtt{2}) \cdot (+4 + v_{\pi_1}(\mathtt{F}))\ p(\mathtt{F}, +4 | \mathtt{2}, \uparrow) \\
&= 1 \cdot (+4) + 0 \cdot [(+2 + v_{\pi_1}(\mathtt{1})) \cdot 0.5 + (0 + v_{\pi_1}(\mathtt{1})) \cdot 0.5] \\
&= 4 \\
\implies v_{\pi_1}(\mathtt{1}) &= 3
\end{aligned}$$

And to improve to a policy π_2, we select the greedy action (policy improvement step). This gives us

$$\begin{aligned}
\pi_2(\mathtt{1}) &= \underset{a \in \mathcal{A}(\mathtt{1})}{\operatorname{argmax}} \begin{cases} \uparrow: & -1 + v_{\pi_1}(\mathtt{2}) = 3 \\ \downarrow: & +1 + v_{\pi_1}(\mathtt{G}) = 1 \end{cases} \\
&= \uparrow \\
\pi_2(\mathtt{2}) &= \underset{a \in \mathcal{A}(\mathtt{2})}{\operatorname{argmax}} \begin{cases} \uparrow: & +4 + v_{\pi_1}(\mathtt{F}) = 4 \\ \downarrow: & (+2 + v_{\pi_1}(\mathtt{1})) \cdot 0.5 + (0 + v_{\pi_1}(\mathtt{1})) \cdot 0.5 = 4 \end{cases} \\
&= \uparrow
\end{aligned}$$

We see that the policy $\pi_2 = \pi_1$, so, by the Policy Improvement Theorem, we have found the optimal policy. The optimal policy is actually

$$\pi_*(s) = \uparrow \quad \forall\ s \in \mathcal{S} \setminus \mathcal{S}^+$$

Here, $n = 2$, and we cnverged to the optimal policy in 2 iterations. As n increases, it is expected that the number of iterations till convergence will increase by a factor of n, i.e. we converge in $O(n)$ iterations.

Question-19 to Question-22

Problem setting too long to dump here.

Solution-19

The given problem can be modelled as an MDP as follows. The set of states and actions are

$$\begin{aligned}
\mathcal{S} &= \{\mathtt{healthy}, \mathtt{sick}\} \\
\mathcal{A}(\mathtt{healthy}) &= \{\mathtt{go\text{-}out}, \mathtt{stay\text{-}home}\} \\
\mathcal{A}(\mathtt{sick}) &= \{\mathtt{take\text{-}med}, \mathtt{no\text{-}med}\}
\end{aligned}$$

The MDP is tabulated in Table 5. For all the following parts, we use the value $\gamma = 0.9$ for the discounting factor. First, we note that we are given $p(r|s,a,s')$ and $p(s'|s,a)$. We can use this to find $p(s',r|s,a)$ using the following equations.

$$\begin{aligned} p(r|s,a,s') &= \frac{p(s',r,s,a)}{p(s,a,s')} \\ &= \frac{p(s',r|s,a)\cdot p(s,a)}{p(s'|s,a)\cdot p(s,a)} \\ \implies p(s',r|s,a) &= p(r|s,a,s')\cdot p(s'|s,a) \end{aligned}$$

s	a	s'	r	$p(r\|s,a,s')$	$p(s'\|s,a)$	$p(s',r\|s,a)$
`healthy`	`go-out`	`healthy`	0	0.1	0.7	0.07
`healthy`	`go-out`	`healthy`	+20	0.9	0.7	0.63
`healthy`	`go-out`	`sick`	0	0.1	0.3	0.03
`healthy`	`go-out`	`sick`	-10	0.9	0.3	0.27
`healthy`	`stay-home`	`healthy`	0	0.05	0.95	0.0475
`healthy`	`stay-home`	`healthy`	+10	0.95	0.95	0.9025
`healthy`	`stay-home`	`sick`	0	0.1	0.05	0.005
`healthy`	`stay-home`	`sick`	-10	0.9	0.05	0.045
`sick`	`take-med`	`healthy`	-1	1.0	0.9	0.9
`sick`	`take-med`	`sick`	-2	1.0	0.1	0.1
`sick`	`no-med`	`healthy`	0	1.0	0.6	0.6
`sick`	`no-med`	`sick`	-1	1.0	0.4	0.4

Table 5: The MDP table for the Sick-Healthy Person Problem

Solution-20

We are required to consider a policy that picks all actions in each state with equal probability. So, we have

$$\begin{aligned} \pi(\texttt{go-out} \mid \texttt{healthy}) = \pi(\texttt{stay-home} \mid \texttt{healthy}) &= 0.5 \\ \pi(\texttt{take-med} \mid \texttt{sick}) = \pi(\texttt{no-med} \mid \texttt{sick}) &= 0.5 \end{aligned}$$

To find the value for the policy, we need to calculate the state values of both states with respect to the policy. We use the Bellman equations to form linear equations in the state values.

$$\begin{aligned} v_\pi(\texttt{H}) &= \pi(\texttt{GO} \mid \texttt{H}) \sum_{s',r} \left[r + \gamma v_\pi(s')\right] p(s',r \mid \texttt{H},\texttt{GO}) + \pi(\texttt{SH} \mid \texttt{H}) \sum_{s',r} \left[r + \gamma v_\pi(s')\right] p(s',r \mid \texttt{H},\texttt{SH}) \\ &= 0.5 \cdot \Big[0.9v_\pi(\texttt{H}) \cdot 0.07 + (20 + 0.9v_\pi(\texttt{H})) \cdot 0.63 + 0.9v_\pi(\texttt{S}) \cdot 0.03 + (-10 + 0.9v_\pi(\texttt{S})) \cdot 0.27\Big] \\ &+ 0.5 \cdot \Big[0.9v_\pi(\texttt{H}) \cdot 0.0475 + (10 + 0.9v_\pi(\texttt{H})) \cdot 0.9025 + 0.9v_\pi(\texttt{S}) \cdot 0.005 + (-10 + 0.9v_\pi(\texttt{S})) \cdot 0.045\Big] \\ &= 0.5 \cdot \Big[0.063v_\pi(\texttt{H}) + 12.6 + 0.567v_\pi(\texttt{H}) + 0.027v_\pi(\texttt{S}) - 2.7 + 0.243v_\pi(\texttt{S})\Big] \\ &+ 0.5 \cdot \Big[0.04275v_\pi(\texttt{H}) + 9.025 + 0.81225v_\pi(\texttt{H}) + 0.0045v_\pi(\texttt{S}) - 0.45 + 0.0405v_\pi(\texttt{S})\Big] \\ &= 0.0315v_\pi(\texttt{H}) + 6.3 + 0.2835v_\pi(\texttt{H}) + 0.0135v_\pi(\texttt{S}) - 1.35 + 0.1215v_\pi(\texttt{S}) \\ &+ 0.021375v_\pi(\texttt{H}) + 4.5125 + 0.406125v_\pi(\texttt{H}) + 0.00225v_\pi(\texttt{S}) - 0.225 + 0.02025v_\pi(\texttt{S}) \\ &= 0.7425v_\pi(\texttt{H}) + 0.15775v_\pi(\texttt{S}) + 9.2375 \end{aligned}$$

$$\begin{aligned}
v_\pi(\texttt{S}) &= \pi(\texttt{TM} \mid \texttt{S}) \sum_{s',r} \big[r + \gamma v_\pi(s')\big]\, p(s', r \mid \texttt{S}, \texttt{TM}) + \pi(\texttt{NM} \mid \texttt{S}) \sum_{s',r} \big[r + \gamma v_\pi(s')\big]\, p(s', r \mid \texttt{S}, \texttt{NM}) \\
&= 0.5 \cdot \Big[(-1 + 0.9 v_\pi(\texttt{H})) \cdot 0.9 + (-2 + 0.9 v_\pi(\texttt{S})) \cdot 0.1\Big] \\
&+ 0.5 \cdot \Big[0.9 v_\pi(\texttt{H}) \cdot 0.6 + (-1 + 0.9 v_\pi(\texttt{S})) \cdot 0.4\Big] \\
&= -0.45 + 0.405 v_\pi(\texttt{H}) - 0.1 + 0.045 v_\pi(\texttt{S}) + 0.27 v_\pi(\texttt{H}) - 0.2 + 0.18 v_\pi(\texttt{S}) \\
&= 0.675 v_\pi(\texttt{H}) + 0.225 v_\pi(\texttt{S}) - 0.75
\end{aligned}$$

Therefore, the two Bellman Equations give us the following linear equations

$$\begin{gathered}
0.2575 v_\pi(\texttt{H}) - 0.15775 v_\pi(\texttt{S}) = 9.2375 \\
0.675 v_\pi(\texttt{H}) - 0.775 v_\pi(\texttt{S}) = 0.75
\end{gathered}$$

Solving these equations, we get $v_\pi(\texttt{healthy}) = 75.641$ and $v_\pi(\texttt{sick}) = 64.913$.

Solution-21

To find the expected return $\mathbb{E}[G_t | S_t = \texttt{H}, S_{t+1} = \texttt{S}]$, we use the following

$$\begin{aligned}
\mathbb{E}[G_t | S_t = \texttt{H}, S_{t+1} = \texttt{S}] &= \mathbb{E}[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = \texttt{H}, S_{t+1} = \texttt{S}] \\
&= \mathbb{E}[R_{t+1} | S_t = \texttt{H}, S_{t+1} = \texttt{S}] + \gamma \cdot \mathbb{E}[v_\pi(S_{t+1}) | S_t = \texttt{H}, S_{t+1} = \texttt{S}]
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[v_\pi(S_{t+1}) | S_t = \texttt{H}, S_{t+1} = \texttt{S}] &= \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i R_{t+2+i} \;\middle|\; S_t = \texttt{H}, S_{t+1} = \texttt{S}\right] \\
&= \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i R_{t+2+i} \;\middle|\; S_{t+1} = \texttt{S}\right] \\
&= v_\pi(\texttt{S})
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[R_{t+1} | S_t = \texttt{H}, S_{t+1} = \texttt{S}] &= \sum_r r P[R_{t+1} = r | S_t = \texttt{H}, S_{t+1} = \texttt{S}] \\
&= \sum_{a \in \mathcal{A}(\texttt{H}), r} r P[R_{t+1} = r | S_t = \texttt{H}, S_{t+1} = \texttt{S}, A_t = a] P[A_t = a | S_t = \texttt{H}, S_{t+1} = \texttt{S}] \\
&= \sum_{a \in \mathcal{A}(\texttt{H}), r} r \, \frac{P[S_{t+1} = \texttt{S}, R_{t+1} = r | S_t = \texttt{H}, A_t = a]}{P[S_{t+1} = \texttt{S} | S_t = \texttt{H}, A_t = a]} \, P[A_t = a | S_t = \texttt{H}, S_{t+1} = \texttt{S}] \\
&= \sum_{a \in \mathcal{A}(\texttt{H}), r} r \, \frac{p(\texttt{S}, r \mid \texttt{H}, a)}{p(\texttt{S} \mid \texttt{H}, a)} \, P[A_t = a | S_t = \texttt{H}, S_{t+1} = \texttt{S}]
\end{aligned}$$

Next, we simplify the conditional probability of action $A_t = a$ conditioned on the states $S_t = \texttt{H}$ and $S_{t+1} = \texttt{S}$.

$$\begin{aligned}
P[A_t = a | S_t = \texttt{H}, S_{t+1} = \texttt{S}] &= \frac{P[A_t = a, S_{t+1} = \texttt{S} | S_t = \texttt{H}]}{P[S_{t+1} = \texttt{S} | S_t = \texttt{H}]} \\
&= \frac{P[A_t = a | S_t = \texttt{H}] \; P[S_{t+1} = \texttt{S} | S_t = \texttt{H}, A_t = a]}{\sum_{\hat{a} \in \mathcal{A}(\texttt{H})} P[A_t = \hat{a}, S_{t+1} = \texttt{S} | S_t = \texttt{H}]} \\
&= \frac{\pi(a \mid \texttt{H}) \; p(\texttt{S} \mid \texttt{H}, a)}{\sum_{\hat{a} \in \mathcal{A}(\texttt{H})} \pi(\hat{a} \mid \texttt{H}) \; p(\texttt{S} \mid \texttt{H}, \hat{a})} \\
&= \frac{p(\texttt{S} \mid \texttt{H}, a)}{p(\texttt{S} \mid \texttt{H}, \texttt{GO}) + p(\texttt{S} \mid \texttt{H}, \texttt{SH})} \qquad \text{since } \pi \text{ is an equiprobable random policy} \\
&= \frac{p(\texttt{S} \mid \texttt{H}, a)}{0.3 + 0.05} = \frac{p(\texttt{S} \mid \texttt{H}, a)}{0.35}
\end{aligned}$$

So, we get the following expression for the conditional expectation of reward $R_{t+1} = r$ conditioned on the states $S_t = \mathtt{H}$ and $S_{t+1} = \mathtt{S}$

$$\begin{aligned}
\mathbb{E}[R_{t+1}|S_t = \mathtt{H}, S_{t+1} = \mathtt{S}] &= \sum_{a \in \mathcal{A}(\mathtt{H})} \frac{P[A_t = a|S_t = \mathtt{H}, S_{t+1} = \mathtt{S}]}{p(\mathtt{S} \mid \mathtt{H}, a)} \sum_r r\ p(\mathtt{S}, r \mid \mathtt{H}, a) \\
&= \sum_{a \in \mathcal{A}(\mathtt{H})} \frac{p(\mathtt{S} \mid \mathtt{H}, a)}{0.35 \cdot p(\mathtt{S} \mid \mathtt{H}, a)} \sum_r r\ p(\mathtt{S}, r \mid \mathtt{H}, a) \\
&= \frac{1}{0.35} \sum_{a \in \mathcal{A}(\mathtt{H})} \sum_r r\ p(\mathtt{S}, r \mid \mathtt{H}, a)
\end{aligned}$$

Hence, the final expression and value for the required conditional return is

$$\begin{aligned}
\mathbb{E}[G_t|S_t = \mathtt{H}, S_{t+1} = \mathtt{S}] &= \gamma v_\pi(\mathtt{S}) + \frac{1}{0.35} \sum_{a \in \mathcal{A}(\mathtt{H})} \sum_r r\ p(\mathtt{S}, r \mid \mathtt{H}, a) \\
&= 0.9 \cdot 64.913 + \frac{1}{0.35} \left[\left(\sum_r r\ p(\mathtt{S}, r \mid \mathtt{H}, \mathtt{GO}) \right) + \left(\sum_r r\ p(\mathtt{S}, r \mid \mathtt{H}, \mathtt{SH}) \right) \right] \\
&= 58.4217 + \frac{1}{0.35} \left[(-10 \cdot 0.27) + (-10 \cdot 0.045) \right] \\
&= 58.4217 - \frac{3.15}{0.35} = 58.4217 - 9 = 49.4217
\end{aligned}$$

Solution-22

We are required to find the optimal policy π_* and the optimal state values $v_*(s)$ for all states. This is done through Policy iteration, with a maximum of three iterations. We start with the equiprobable random policy π_0, which is the same as the policy used in the previous solutions.

We first find $v_{\pi_0}(s)$ for the states. This is the policy evaluation step. From Solution-20, we have

$$\begin{aligned}
v_{\pi_0}(\mathtt{H}) &= 75.641 \\
v_{\pi_0}(\mathtt{S}) &= 64.913
\end{aligned}$$

Now, we improve the policy to π_1 by selecting the greedy action in each state. This is the policy improvement step.

$$\begin{aligned}
\pi_1(\mathtt{H}) &= \operatorname*{argmax}_{a \in \mathcal{A}(\mathtt{H})} \begin{cases}
\mathtt{GO}: & (0 + 0.9 v_{\pi_0}(\mathtt{H})) \cdot 0.07 + (+20 + 0.9 v_{\pi_0}(\mathtt{H})) \cdot 0.63 \\
& +(0 + 0.9 v_{\pi_0}(\mathtt{S})) \cdot 0.03 + (-10 + 0.9 v_{\pi_0}(\mathtt{S})) \cdot 0.027 \\
& = 63.315 \\
\mathtt{SH}: & (0 + 0.9 v_{\pi_0}(\mathtt{H})) \cdot 0.0475 + (+10 + 0.9 v_{\pi_0}(\mathtt{H})) \cdot 0.9025 \\
& +(0 + 0.9 v_{\pi_0}(\mathtt{S})) \cdot 0.005 + (-10 + 0.9 v_{\pi_0}(\mathtt{S})) \cdot 0.045 \\
& = 74.641
\end{cases} \\
&= \ \mathtt{SH} \\
\pi_1(\mathtt{S}) &= \operatorname*{argmax}_{a \in \mathcal{A}(\mathtt{S})} \begin{cases}
\mathtt{TM}: & (-1 + 0.9 v_{\pi_0}(\mathtt{H})) \cdot 0.9 + (-2 + 0.9 v_{\pi_0}(\mathtt{S})) \cdot 0.1 = 66.011 \\
\mathtt{NM}: & (0 + 0.9 v_{\pi_0}(\mathtt{H})) \cdot 0.6 + (-1 + 0.9 v_{\pi_0}(\mathtt{S})) \cdot 0.4 = 63.815
\end{cases} \\
&= \ \mathtt{TM}
\end{aligned}$$

This means that our deterministic policy is

$$\begin{aligned}
\pi_1(\mathtt{H}) &= \mathtt{SH} \\
\pi_1(\mathtt{S}) &= \mathtt{TM}
\end{aligned}$$

Now, we evaluate the updated/improved policy π_1 to get $v_{\pi_1}(s)$. This is the policy evaluation step in the second iteration. We get the following equations for our deterministic policy.

$$\begin{aligned}
v_{\pi_1}(\texttt{H}) &= \pi(\texttt{SH} \mid \texttt{H}) \sum_{s',r} \left[r + \gamma v_{\pi_1}(s')\right] p(s', r \mid \texttt{H}, \pi(\texttt{H})) = \sum_{s',r} \left[r + \gamma v_{\pi_1}(s')\right] p(s', r \mid \texttt{H}, \texttt{SH}) \\
&= (0 + 0.9v_{\pi_1}(\texttt{H})) \cdot 0.0475 + (+10 + 0.9v_{\pi_1}(\texttt{H})) \cdot 0.9025 \\
&+ (0 + 0.9v_{\pi_1}(\texttt{S})) \cdot 0.005 + (-10 + 0.9v_{\pi_1}(\texttt{S})) \cdot 0.045 \\
&= 0.04275v_{\pi_1}(\texttt{H}) + 9.025 + 0.81225v_{\pi_1}(\texttt{H}) + 0.0045v_{\pi_1}(\texttt{S}) - 0.45 + 0.0405v_{\pi_1}(\texttt{S}) \\
&= 0.855v_{\pi_1}(\texttt{H}) + 0.045v_{\pi_1}(\texttt{S}) + 8.575 \\
v_{\pi_1}(\texttt{S}) &= \pi(\texttt{TM} \mid \texttt{S}) \sum_{s',r} \left[r + \gamma v_{\pi_1}(s')\right] p(s', r \mid \texttt{S}, \pi(\texttt{S})) = \sum_{s',r} \left[r + \gamma v_{\pi_1}(s')\right] p(s', r \mid \texttt{S}, \texttt{TM}) \\
&= (-1 + 0.9v_{\pi_1}(\texttt{H})) \cdot 0.9 + (-2 + 0.9v_{\pi_1}(\texttt{S})) \cdot 0.1 \\
&= -0.9 + 0.81v_{\pi_1}(\texttt{H}) - 0.2 + 0.09v_{\pi_1}(\texttt{S}) \\
&= 0.81v_{\pi_1}(\texttt{H}) + 0.09v_{\pi_1}(\texttt{S}) - 1.1
\end{aligned}$$

Therefore, the two Bellman Equations give us the following linear equations

$$\begin{aligned}
0.145v_{\pi_1}(\texttt{H}) - 0.045v_{\pi_1}(\texttt{S}) &= 8.575 \\
0.81v_{\pi_1}(\texttt{H}) - 0.91v_{\pi_1}(\texttt{S}) &= 1.1
\end{aligned}$$

Solving these equations, we get $v_{\pi_1}(\texttt{healthy}) = 81.191$ and $v_{\pi_1}(\texttt{sick}) = 71.060$. Now, we improve the policy to π_2 by selecting the greedy action in each state. This is the policy improvement step of iteration 2.

$$\begin{aligned}
\pi_2(\texttt{H}) &= \operatorname*{argmax}_{a \in \mathcal{A}(\texttt{H})} \begin{cases} \texttt{GO}: & (0 + 0.9v_{\pi_1}(\texttt{H})) \cdot 0.07 + (+20 + 0.9v_{\pi_1}(\texttt{H})) \cdot 0.63 \\ & +(0 + 0.9v_{\pi_1}(\texttt{S})) \cdot 0.03 + (-10 + 0.9v_{\pi_1}(\texttt{S})) \cdot 0.027 \\ & = 67.125 \\ \texttt{SH}: & (0 + 0.9v_{\pi_1}(\texttt{H})) \cdot 0.0475 + (+10 + 0.9v_{\pi_1}(\texttt{H})) \cdot 0.9025 \\ & +(0 + 0.9v_{\pi_1}(\texttt{S})) \cdot 0.005 + (-10 + 0.9v_{\pi_1}(\texttt{S})) \cdot 0.045 \\ & = 81.191 \end{cases} \\
&= \texttt{SH} \\
\pi_2(\texttt{S}) &= \operatorname*{argmax}_{a \in \mathcal{A}(\texttt{S})} \begin{cases} \texttt{TM}: & (-1 + 0.9v_{\pi_1}(\texttt{H})) \cdot 0.9 + (-2 + 0.9v_{\pi_1}(\texttt{S})) \cdot 0.1 = 71.060 \\ \texttt{NM}: & (0 + 0.9v_{\pi_1}(\texttt{H})) \cdot 0.6 + (-1 + 0.9v_{\pi_1}(\texttt{S})) \cdot 0.4 = 69.024 \end{cases} \\
&= \texttt{TM}
\end{aligned}$$

This clearly indicates that the policies $\pi_1 = \pi_2$. So, by the Policy Improvement Theorem, we have found the optimal policy. Therefore, the optimal value function and policy are

$$v_*(s) = \begin{cases} 81.191 & \text{if } s = \texttt{H} \\ 71.060 & \text{if } s = \texttt{S} \end{cases}$$

$$\pi_*(s) = \begin{cases} \texttt{SH} & \text{if } s = \texttt{H} \\ \texttt{TM} & \text{if } s = \texttt{S} \end{cases}$$