HW 8 171

Eugene Francisco

December 2024

Problem 3

Part A

State Space: The space $\mathcal{S}$ are just the numbers $\{0, \ldots, n\}$ for the different lilypads the frog could be on. Of these, the non-terminating sets $\mathcal{N}$ are the numbers $\{1, \ldots, n-1\} \subset S$, since these are the lilypads for which we keep playing the game.

Action Space: There are two possible actions at any lilypad, so $\mathcal{A} = \{A, B\}$, where each action represents the corresponding croak.

Transition Function:

We are interested in $\mathcal{P} : \mathcal{N} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, where $\mathcal{P}(s, a, s') := \sum_{r \in \mathcal{D}} \mathcal{P}_{\mathcal{R}}(s, a, r, s') = \mathbb{P}(S_{t+1} = s' | A_t = a, S_t = s)$.

$$\begin{aligned}
\mathcal{P}(s, a, s') &= \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a) \\
&= \begin{cases} s/n & : A_t = A, s' = s - 1 \\ (n-s)/n & : A_t = A, s' = s + 1 \\ 0 & : A_t = A, s' \neq s \pm 1 \\ 1/n & : A_t = B. \end{cases}
\end{aligned}$$

Reward Function:

Since there isn't a given reward structure, we will set up rewards as follows: Whenever an agent transitions to state s, they receive reward s, the motivation being to incentivize getting to the nth lilypad. First, the reward transition function:

$$\begin{aligned}
\mathcal{R}_T(s, a, s') &:= \mathbb{E}(R_{t+1} | (S_{t+1} = s', S_t = s, A_t = a)) \\
&= s'
\end{aligned}$$

We note that because of this reward structure, $\mathcal{P}_{\mathcal{R}}(s, a, r, s') = \mathbb{P}((T_{t+1} = r, S_{t+1} = s') | S_t = s, A_t = a) = \mathcal{P}(s, a, s')$, since going to state s' guarantees the reward.

Also the reward function:

$$\begin{aligned}\mathcal{R}(s,a) &:= \mathbb{E}(R_{t+1}|S_t = a, A_t = a)\\ &= \begin{cases}\sum_{s'\in\mathcal{S}} r\mathcal{P}_{\mathcal{R}}(s,A,r,s') & : a = A\\ \sum_{s'\in\mathcal{S}} r\mathcal{P}_{\mathcal{R}}(s,B,r,s') & : a = B.\end{cases}\end{aligned}$$

By the remark above and our reward setup, this simplifies to

$$\begin{aligned}\mathcal{R}(s,a) &= \begin{cases}(s-1)\frac{s}{n} + (s+1)\frac{n-s}{n} & : a = A\\ \sum_{s'\in\mathcal{S}} s'\frac{1}{n} & : a = B\end{cases}\\ &= \begin{cases}\frac{-2s+sn+n}{n} & : a = A\\ \frac{1}{n}\frac{n(n+1)}{2} & : a = B\end{cases}\\ &= \begin{cases}\frac{-2s+sn+n}{n} & : a = A\\ \frac{n+1}{2} & : a = B.\end{cases}\end{aligned}$$

Question 4

Part A:

As the question suggests, we begin with $v_0(s_1) = 10$, $v_0(s_2) = 2$, $v_0(s_3) = 0$. Let's calculate what our greedy policy π_D^0 is for this initialized valuation. First,

$$\pi_D^0(s_1) = \text{argmax}_{a\in A}\{q_0(s_1,a_1), q_0(s_1,a_2)\}$$

where

$$\begin{aligned}q_0(s_1,a_1) &= R(s_1,a_1) + p(s_1,a_1,s_1)v_0(s_1) + p(s_1,a_1,s_2)v_0(s_2)\\ &= 8 + 0.25\cdot 10 + 0.65\cdot 1\\ &= 11.15\end{aligned}$$

while (with a similar calculation)

$$\begin{aligned}q_0(s_1,a_2) &= 10 + 0.1\cdot 10 + 0.4\cdot 1\\ &= 11.4.\end{aligned}$$

Since $q_0(s_1,a_2)$ is higher, we pick a_2 for the strategy when we are on state s_1. Similarly

$$\begin{aligned}q_0(s_2,a_1) &= 1 + 0.3\cdot 10 + 0.15\cdot 1\\ &= 4.15\\ q_0(s_2,a_2) &= -1 + 0.25\cdot 10 + 0.55\cdot 1\\ &= 2.05\end{aligned}$$

so we choose a_1 as the policy action for state s_2. To recap, $\boxed{\pi_D^0(s_1) = a_2, \pi_D^0(s_2) = a_1}$.

First value iteration: Note that since the policies are deterministic, $R^{\pi^0}(s_1) = R(s_1, a_1) = 8$ and $R^{\pi^0}(s_2) = R(s_2, a_2) = -1$. But this means that the Bellman update equation $v_1(s) = R^{\pi^0}(s) + \sum_{s \in N} p^{\pi}(s, s')V_i(s')$ simplifies (because we always take policy $\pi_D^0(s)$) to

$$v_1(s) = R(s, \pi_D^0(s)) + \sum_{s' \in N} p(s, \pi_D^0(s), s')v_0(s')$$

which is just the quality of taking the greedy action at a state s. That means that, as long as our policy is greedy,

$$v_{t+1}(s) = q_t(s, \pi_D^t(s))$$

which will heavily simplify later iterations. So $\boxed{v_1(s_1) = 11.4 \text{ and } v_1(s_2) = 4.15}$ and $v_1(s_3) = 0$. (Note that $v_k(s_3) = 0$ for all k because it is a terminating state whose value was initialized at 0).

First greedy policy: As before, we calculate the relevant $q_k(\cdot, \cdot)$.

$$\pi_D^1(s_1) = \text{argmax}_{a \in A}\{q_1(s_1, a_1), q_1(s_1, a_2)\}$$

where

$$\begin{aligned} q_1(s_1, a_1) &= 8 + 0.25 \cdot 11.4 + 0.65 \cdot 4.15 \\ &= 13.55 \\ q_1(s_1, a_2) &= 10 + 0.1 \cdot 11.4 + 0.4 \cdot 4.15 \\ &= 12.8 \end{aligned}$$

so we pick a_1 as the policy choice when on state s_1. Similarly,

$$\pi_D^1(s_2) = \text{argmax}_{a \in A}\{q_1(s_2, a_1), q_1(s_2, a_2)\}$$

where

$$\begin{aligned} q_1(s_2, a_1) &= 1 + 0.3 \cdot 11.4 + 0.15 \cdot 4.15 \\ &= 5.04 \\ q_1(s_2, a_2) &= -1 + 0.25 \cdot 11.4 + 0.55 \cdot 4.15 \\ &= 4.13 \end{aligned}$$

so we pick a_1 as the policy pick for state s_2. To recap then, $\boxed{\pi_D^1(s_1) = \pi_D^1(s_2) = a_1}$.

Second Value Iteration: Using the same trick as above,

$$\boxed{v_2(s_1) = 12.8,\ v_2(s_2) = 5.04, \text{ and } v_2(s_3) = 0.}$$

Second policy iteration:

$$\pi_D^2(s_1) = \text{argmax}_{a \in A}\{q_2(s_1, a_1), q_2(s_1, a_2)\}$$

where

$$\begin{aligned}
q_2(s_1, a_1) &= 8 + 0.25 \cdot 12.8 + 0.65 \cdot 5.04 \\
&= 14.48 \\
q_2(s_1, a_2) &= 10 + 0.1 \cdot 12.8 + 0.4 \cdot 5.04 \\
&= 13.30
\end{aligned}$$

so we pick $\pi_D^2(s_1) = a_1$. Similarly,

$$\pi_D^2(s_2) = \text{argmax}_{a \in A}\{q_2(s_2, a_1), q_2(s_2, a_2)\}$$

where

$$\begin{aligned}
q_2(s_2, a_1) &= 1 + 0.3 \cdot 12.8 + 0.15 \cdot 5.04 \\
&= 5.60 \\
q_2(s_2, a_2) &= -1 + 0.25 \cdot 12.8 + 0.55 \cdot 5.04 \\
&= 4.97
\end{aligned}$$

so we pick $\pi_D^2(s_2) = a_1$, ending with

$$\boxed{\pi_D^2(s_1) = \pi_D^2(s_2) = a_1.}$$

(For use in the next section, note the final valuation values of $v_3(s_1) = 14.48$, $v_3(s_2) = 5.60$ and $v_3(s_3) = 0$.)

Part B

To show stability, we wish to show that $\pi_D^k(s) = a_1$ for all $k \geq 3$. In other words,

$$q_k(s, a_1) - q_k(s, a_2) > 0.$$

First, s_1:

$$\begin{aligned}
q_k(s_1, a_1) - q_k(s_1, a_2) &= R(s_1, a_1) - R(s_1, a_2) + \sum_{s' \in N} v_k(s')(p(s_1, a_1, s') - p(s_1, a_2, s')) \\
&= -2 + v_k(s_1)(0.25 - 0.1) + v_k(s_2)(0.65 - 0.4) \\
&= -2 + v_k(s_1)(0.15) + v_k(s_2)(0.25)
\end{aligned}$$

For $k \geq 3$, due to the monotonicity of each entry of v_k, we have that $v_k(s_1) \geq v_3(s_1) = 13.88$ and $v_k(s_2) \geq v_3(s_2) = 5.46$. Substituting these, we get

$$\begin{aligned}
q_k(s_1, a_1) - q_k(s_1, a_2) &\geq -2 + 14.48 \cdot (0.15) + 5.60 \cdot (0.25) \\
&= 1.572 \\
&> 0.
\end{aligned}$$

A similar argument works for s_2, where

$$\begin{aligned}
q_k(s_2, a_1) - q_k(s_2, a_2) &= 2 + v_k(s_1)(0.3 - 0.25) + v_k(s_2)(0.15 - 0.55) \\
&= 2 + v_k(s_1)(0.05) + v_k(s_2)(-0.4) \\
&\geq 2 + 14.48 \cdot (0.05) + 5.60 \cdot (-0.4) \\
&= 0.484 \\
&> 0.
\end{aligned}$$

Which is exactly what we want.

Part C

As noted when calculating the valuation functions in part A, because we take action a_1 no matter which state we are in, we have that $R^\pi(s) = R(s, a_1)$, for each $s \in N$. Also, $p^\pi(s, s') = P(s, a_1, s')$. Let $v_i = v^{\pi^2}(s_i)$. Then

$$\begin{aligned}
v_1 &= R^{\pi^2}(s) + \sum_{s \in N} p^\pi(s, s') v_s \\
&= R(s_1, a_1) + p(s_1, a_1, a_1) v_1 + p(s_1, a_1, s_2) v_2
\end{aligned}$$

and similarly

$$v_2 = R(s_2, a_1) + p(s_2, a_1, s_1) v_1 + p(s_2, a_1, s_2) v_2.$$

Substituting in for the values, we get a system of two equations:

$$-8 = v_1(-0.75) + v_2(0.65) \tag{1}$$
$$-1 = v_1(0.3) + v_2(-0.85). \tag{2}$$

Solved, this yields $\boxed{v^{\pi^2}(s_1) = 16.84 \text{ and } v^{\pi^2}(s_2) = 7.12}$.

Part D:

We begin by calculating what the initial greedy policy is:

$$\begin{aligned}
q_0(s_1, a_1) &= 8 + 0.25 \cdot 10 + 0.65 \cdot 1 \\
&= 11.15 \\
q_0(s_1, a_2) &= 11 + 0.1 \cdot 10 + 0.4 \cdot 1 \\
&= 12.4.
\end{aligned}$$

We pick then $\boxed{\pi_D^0(s_1) = a_2 \text{ and } v_1(s_1) = 12.4.}$ Similarly,

$$\begin{aligned}
q_0(s_2, a_1) &= 1 + 0.3 \cdot 10 + 0.15 \cdot 1 \\
&= 4.15 \\
q_0(s_2, a_2) &= -1 + 0.25 \cdot 10 + 0.55 \cdot 1 \\
&= 2.05
\end{aligned}$$

giving us $\boxed{\pi_D(s_2) = a_1 \text{ and } v_1(s_2) = 4.15}$. Let's calculate the next greedy policy π_D^1. First,

$$q_1(s_1, a_1) = 8 + 12.4 \cdot (0.25) + 4.15(0.65)$$
$$= 13.8$$
$$q_1(s_1, a_2) = 11 + 12.4 \cdot (0.1) + 4.15 \cdot (0.4)$$
$$= 13.9$$

So we pick $\boxed{\pi_D^1(s_1) = a_2 \text{ and } v_2(s_1) = 13.9.}$ Similarly

$$\begin{aligned}
q_1(s_2, a_1) &= 1 + 12.4 \cdot (0.3) + 4.15 \cdot (0.15) \\
&= 5.34 \\
q_1(s_2, a_2) &= -1 + 12.4 \cdot 0.25 + 4.15 \cdot 0.55 \\
&= 4.38
\end{aligned}$$

giving us $\boxed{\pi_D^1(s_2) = a_1 \text{ and } v_2(s_2) = 5.34.}$

Showing stability: We claim that this policy is stable and thus optimal. We'll show this in the same way as in part B. Namely, for $k \geq 2$

$$\begin{aligned}
q_k(s_1, a_1) - q_k(s_1, a_2) &= R(s_1, a_1) - R(s_1, a_2) + v_k(s_1)(p(s_1, a_1, a_1) - p(s_1, a_2, s_1)) \\
&\quad + v_k(s_2)(p(s_1, a_1, s_2) - p(s_1, a_2, s_2)) \\
&= -3 + v_k(s_1) \cdot (0.15) + v_k(s_2) \cdot (0.25) \\
&\geq -3 + 13.9 \cdot (0.15) + 5.34 \cdot (0.25) \\
&= 0.42 \\
&> 0
\end{aligned}$$

which shows that $\pi_D^k(s_1) = a_1$ for all $k \geq 2$. Similarly,

$$\begin{aligned}
q_k(s_2, a_1) - q_k(s_2, a_2) &= 2 + v_k(s_1) \cdot (0.05) + v_k(s_2) \cdot (-0.4) \\
&\geq 2 + 13.9 \cdot (0.05) + 5.34 \cdot (-0.4) \\
&= 0.559 \\
&> 0
\end{aligned}$$

meaning that $\pi_D^k(s_2) = a_1$ for all $k \geq 2$. In other words, the policy stabilizes to the same as it would have been in if we hadn't changed the reward.