Andrew ElHabr
ISyE 6664
11/3/17

Homework 8

1. a) Yes, this MDP is unichain. There are only two possible stationary deterministic policies, i.e. $\pi_1 : d(s_1) = a_{1,1}, d(s_2) = a_{2,1}, d(s_3) = a_{3,1}$ and $\pi_2 : d(s_1) = a_{1,2}, d(s_2) = a_{2,1}, d(s_3) = a_{3,1}$, and their respective transition matrices are

$$P_{d_1} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \text{ and } P_{d_2} = \begin{bmatrix} 2/3 & 0 & 1/3 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

It is clear that for both of these policies there is a single recurrent class with one transient state ($\{s_1, s_2\}$ is recurrent and $\{s_3\}$ is transient for π_1 and $\{s_1, s_3\}$ is recurrent and $\{s_2\}$ is transient for π_2).

b) Using value iteration and setting $v^0 = (0, 0, 0)$ and $\epsilon = 0.001$, we find that the epsilon-optimal policy is $\pi^*_\epsilon : d(s_1) = a_{1,2}, d(s_2) = a_{2,1}, d(s_3) = a_{3,1}$ after 11 iterations. The algorithm was coded in R, and the output for v for selected iterations and the final output for d_ϵ are shown in the following:

V

	[, 1]	[, 2]	[, 3]
[1,]	0.000000	3.000000	4.000000
[2,]	1.500000	3.000000	4.000000
[3,]	2.333333	4.500000	5.500000
⋮			
[9,]	8.399691	10.400463	11.400463
[10,]	9.400077	11.399691	12.399691
[11,]	10.399949	12.400077	13.400077

d_epsilon

	[, 1]
[1,]	2
[2,]	1
[3,]	1

Note that this result is very dependent on ϵ. For a different value of ϵ, e.g. $\epsilon = 0.0005$, we find that the epsilon-optimal policy is $\pi^*_\epsilon : d(s_1) = a_{1,1}, d(s_2) = a_{2,1}, d(s_3) = a_{3,1}$.

c) Using policy iteration and setting d_0 to be $d_0(s_1) = a_{1,2}, d_0(s_2) = a_{2,1}, d_0(s_3) = a_{3,1}$. In the policy evaluation step, we add the constraint that $h_0(s_1) = 0$, since it is recurrent under d_0, to solve

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix} + \begin{bmatrix} g_0 \\ g_0 \\ g_0 \end{bmatrix} + \left(\begin{bmatrix} 2/3 & 0 & 1/3 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} 0 \\ h_0(s_2) \\ h_0(s_3) \end{bmatrix},$$

and get that $g_0 = 1, h_0(s_2) = 2$, and $h_0(s_3) = 3$. Then, in the policy iteration step, we find that

$$\begin{aligned}
d_1 =& \arg\max\left(\begin{bmatrix}0\\3\\4\end{bmatrix} + \begin{bmatrix}1/2 & 1/2 & 0\\1 & 0 & 0\\1 & 0 & 0\end{bmatrix}\begin{bmatrix}0\\2\\3\end{bmatrix}, \begin{bmatrix}0\\3\\4\end{bmatrix} + \begin{bmatrix}2/3 & 0 & 1/3\\1 & 0 & 0\\1 & 0 & 0\end{bmatrix}\begin{bmatrix}0\\2\\3\end{bmatrix}\right)\\
=& \arg\max\left(\begin{bmatrix}1\\3\\4\end{bmatrix}, \begin{bmatrix}1\\3\\4\end{bmatrix}\right)\\
=& d_0.
\end{aligned}$$

It is clear that both policies can be selected in the last step, so we set $d_1 = d_0$ since this allows us to stop the algorithm. Therefore, $d^* = d_0$, and the optimal policy is $\pi^* : d(s_1) = a_{1,2}, d(s_2) = a_{2,1}, d(s_3) = a_{3,1}$. Note that if we had set d_0 to be $d_0(s_1) = a_{1,1}, d_0(s_2) = a_{2,1}, d_0(s_3) = a_{3,1}$ instead, we would have found that policy to be optimal with the same algorithm and reasoning.

d) First, we find that

$$P^*_{d_1} = \begin{bmatrix}2/3 & 1/3 & 0\\2/3 & 1/3 & 0\\2/3 & 1/3 & 0\end{bmatrix} \text{ and } P^*_{d_2} = \begin{bmatrix}3/4 & 0 & 1/4\\3/4 & 0 & 1/4\\3/4 & 0 & 1/4\end{bmatrix}.$$

Thus,

$$\begin{aligned}
g^{\pi_1} =& P^*_{d_1} r_{d_1} = \begin{bmatrix}2/3 & 1/3 & 0\\2/3 & 1/3 & 0\\2/3 & 1/3 & 0\end{bmatrix}\begin{bmatrix}0\\3\\4\end{bmatrix} = \begin{bmatrix}1\\1\\1\end{bmatrix} \quad \text{and}\\
g^{\pi_2} =& P^*_{d_2} r_{d_2} = \begin{bmatrix}3/4 & 0 & 1/4\\3/4 & 0 & 1/4\\3/4 & 0 & 1/4\end{bmatrix}\begin{bmatrix}0\\3\\4\end{bmatrix} = \begin{bmatrix}1\\1\\1\end{bmatrix}\\
&\implies g^{\pi_1} = \begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}1\\1\\1\end{bmatrix} = g^{\pi_2}.
\end{aligned}$$

In addition,

$$\begin{aligned}
h^{\pi_1} =& (I - P_{d_1} + P^*_{d_1})^{-1}(I - P^*_{d_1}) r_{d_1}\\
=& \left(\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} - \begin{bmatrix}1/2 & 1/2 & 0\\1 & 0 & 0\\1 & 0 & 0\end{bmatrix} + \begin{bmatrix}2/3 & 1/3 & 0\\2/3 & 1/3 & 0\\2/3 & 1/3 & 0\end{bmatrix}\right)^{-1}\left(\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} - \begin{bmatrix}2/3 & 1/3 & 0\\2/3 & 1/3 & 0\\2/3 & 1/3 & 0\end{bmatrix}\right)\begin{bmatrix}0\\3\\4\end{bmatrix}\\
=& \begin{bmatrix}-2/3\\4/3\\7/3\end{bmatrix} \quad \text{and}\\
h^{\pi_2} =& (I - P_{d_2} + P^*_{d_2})^{-1}(I - P^*_{d_2}) r_{d_2}\\
=& \left(\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} - \begin{bmatrix}2/3 & 0 & 1/3\\1 & 0 & 0\\1 & 0 & 0\end{bmatrix} + \begin{bmatrix}3/4 & 0 & 1/4\\3/4 & 0 & 1/4\\3/4 & 0 & 1/4\end{bmatrix}\right)^{-1}\left(\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} - \begin{bmatrix}3/4 & 0 & 1/4\\3/4 & 0 & 1/4\\3/4 & 0 & 1/4\end{bmatrix}\right)\begin{bmatrix}0\\3\\4\end{bmatrix}\\
=& \begin{bmatrix}-3/4\\5/4\\9/4\end{bmatrix}\\
&\implies h^{\pi_1} = \begin{bmatrix}-2/3\\4/3\\7/3\end{bmatrix} > \begin{bmatrix}-3/4\\5/4\\9/4\end{bmatrix} = h^{\pi_2}.
\end{aligned}$$

Thus, the bias optimal policy is clearly $\pi_1 : d(s_1) = a_{1,1}, d(s_2) = a_{2,1}, d(s_3) = a_{3,1}$ since it is the maximal gain policy that has a greater bias than $\pi_2 : d(s_1) = a_{1,2}, d(s_2) = a_{2,1}, d(s_3) = a_{3,1}$. This shows that the policy that policy iteration declared to be optimal, i.e. $\pi^* = \pi_2$, is not necessarily a bias optimal policy.