ISyE 4232 Spring 2013

H. Ayhan

Solutions to Homework 5

1. Define the state space $\mathcal{S} = \{1, 2\}$, where $1 = high$ and $2 = low$. Define the action space $\mathcal{A} = \{1, 2\}$, where $1 = Do\ Nothing$, $2 = Advertise$. First calculate the expected immediate rewards

$$\begin{aligned}
r(1,1) &= 10 \times 0.5 + 4 \times 0.5 = 7 \\
r(2,1) &= 7 \times 0.2 - 2 \times 0.8 = -0.2 \\
r(1,2) &= 7 \times 0.8 + 6 \times 0.2 = 6.8 \\
r(2,2) &= 3 \times 0.4 - 5 \times 0.6 = -1.8
\end{aligned}$$

Let us begin with the policy iteration. We will start the algorithm at some arbitrary policy, say $d_0(1) = 2, d_0(2) = 1$. Now for the policy evaluation step we need to solve the following linear system,

$$\begin{aligned}
V^1_{0.6}(1) &= 6.8 + 0.6 \times \big(0.5 * V^1_{0.6}(1) + 0.5 * V^1_{0.6}(2)\big) \\
V^1_{0.6}(2) &= -0.2 + 0.6 \times \big(0.2 * V^1_{0.6}(1) + 0.8 * V^1_{0.6}(2)\big)
\end{aligned}$$

Solving this system we get $V^1_{0.6}(1) = 13.72$ and $V^1_{0.6}(1) = 2.78$. Now for policy improvement:

$$\begin{aligned}
d_1(1) &= \arg\max_{a \in A_1}\{7 + 0.6 \times (0.5 * 13.72 + 0.5 * 2.78)\ ,\ 6.8 + 0.6 \times (0.8 * 13.72 + 0.2 * 2.78)\} \\
&= \arg\max_{a \in A_1}\{11.97\ ,\ 13.72\} \\
&= 2 \\
d_1(2) &= \arg\max_{a \in A_2}\{-0.2 + 0.6 \times (0.2 * 13.72 + 0.8 * 2.78)\ ,\ -1.8 + 0.6 \times (0.4 * 13.72 + 0.6 * 2.78)\} \\
&= \arg\max_{a \in A_2}\{2.78\ ,\ 2.49\} \\
&= 1
\end{aligned}$$

So we have $d_0(1) = d_1(1) = 2, d_0(2) = d_1(2) = 1$, therefore this is the optimal policy.

Now we do the value iteration. Let us arbitrarily take $V_0(1) = V_0(2) = 0$. And for good measure let's take $\epsilon = 10^{-5}$, this may be unnecessarily small, but I'd rather err in the side of caution.

$$\begin{aligned}
V_1(1) &= \max\{7\ ,\ 6.8\} = 7 \\
V_1(2) &= \max\{-0.2\ ,\ -1.8\} = -0.2
\end{aligned}$$

We calculate the stopping condition and get

$$\max\{|7 - 0|, |-0.2 - 0|\} = 7 \not< 3.33 \times 10^-6 = 10^{-5}\frac{1 - 0.6}{2 \times 0.6}$$

so we continue, set $n = 1$ and

$$
\begin{aligned}
V_2(1) &= \max\{7 + 0.6 \times (0.5 * 7 + 0.5 * (-0.2))\ ,\ 6.8 + 0.6 \times (0.8 * 7 + 0.2 * (-0.2))\} \\
&= \max\{9.04\ ,\ 10.14\} \\
&= 10.14 \\
V_2(2) &= \max\{-0.2 + 0.6 \times (0.2 * 7 + 0.8 * (-0.2))\ ,\ -1.8 + 0.6 \times (0.4 * 7 + 0.6 * (-0.2))\} \\
&= \max\{0.544\ ,\ -0.192\} \\
&= 0.544
\end{aligned}
$$

We calculate the stopping condition and get

$$\max\{|10.14 - 7|, |0.544 - (-0.2)|\} = 3.14 \not< 3.33 \times 10^-6$$

So we need to continue. In order to do this faster I coded this in Java, the results are as follows:

```
      Solver set to Value Iter. Solver  (Disc)
  2 states found.

      Max difference from previous value = 7.0
      Max difference from previous value = 3.2368000000000006
      Max difference from previous value = 1.6371379200000007
      Max difference from previous value = 0.8494684692480003
      Max difference from previous value = 0.45052549966725053
      Max difference from previous value = 0.2432745115610313
      Max difference from previous value = 0.1332456089417331
      Max difference from previous value = 0.07378407388916486
      Max difference from previous value = 0.04119541329742482
      Max difference from previous value = 0.023140960092947083
      Max difference from previous value = 0.013057128290842712
      Max difference from previous value = 0.007391188601182819
      Max difference from previous value = 0.004193611814780951
      Max difference from previous value = 0.0023833254986094232
      Max difference from previous value = 0.0013561042037188997
      Max difference from previous value = 7.722697412262391E-4
      Max difference from previous value = 4.400532273116653E-4
      Max difference from previous value = 2.5085691631687723E-4
      Max difference from previous value = 1.430467156851023E-4
      Max difference from previous value = 8.158728624252376E-5
      Max difference from previous value = 4.6540688423135634E-5
      Max difference from previous value = 2.655153604891325E-5
Value Iter. Solver  (Disc)
********* Best Policy *********

In every stage do:
STATE      ------> ACTION
(1)        ------> (2)
(2)        ------> (1)
Value Function:
(1)          :     -13.72
(2)          :      -2.78
22 iterations
```

As expected the optimal policy is the same for both methods, that is $d_0(1) = 2, d_0(2) = 1$.