Assignment 4

Monte Fischer

March 2, 2022

Problem 1

We initialize $V_0(s_1) = 10, V_0(s_2) = 1, V_0(s_3) = 0$. Then we compute V_1 according to the Bellman Optimality Operator B^*:

$$
\begin{aligned}
V_1(s_1) = B^*(V_0)(s_1) = \max_a q_0(s_1, a) &= \max_a \{R(s_1, a) + \gamma \sum_{s' \in \mathcal{S}} P(s_1, a, s')V_0(s')\} \\
&= \max\{8 + 0.2 \cdot 10 + 0.6 \cdot 1 + 0.2 \cdot 0, 10 + 0.1 \cdot 10 + 0.2 \cdot 1 + 0.7 \cdot 0\} \\
&= \max\{10.6, 11.2\} = 11.2 \\
V_1(s_2) = B^*(V_0)(s_2) = \max_a q_0(s_2, a) &= \max_a \{R(s_2, a) + \gamma \sum_{s' \in \mathcal{S}} P(s_2, a, s')V_0(s')\} \\
&= \max\{1 + 0.3 \cdot 10 + 0.3 \cdot 1 + 0.4 \cdot 0, -1 + 0.5 \cdot 10 + 0.3 \cdot 1 + 0.2 \cdot 0\} \\
&= \max\{4.3, 4.3\} = 4.3
\end{aligned}
$$

Similarly, it is easy to see that $V_1(s_3) = 0$. To compute V_2 we apply B^* to V_1.

$$
\begin{aligned}
V_2(s_1) = B^*(V_1)(s_1) = \max_a q_1(s_1, a) &= \max_a \{R(s_1, a) + \gamma \sum_{s' \in \mathcal{S}} P(s_1, a, s')V_1(s')\} \\
&= \max\{8 + 0.2 \cdot 11.2 + 0.6 \cdot 4.3 + .2 \cdot 0, 10 + 0.1 \cdot 11.2 + 0.2 \cdot 4.3 + 0.7 \cdot 0\} \\
&= \max\{12.82, 11.98\} = 12.82 \\
V_2(s_2) = B^*(V_1)(s_2) = \max_a q_1(s_2, a) &= \max_a \{R(s_2, a) + \gamma \sum_{s' \in \mathcal{S}} P(s_2, a, s')V_1(s')\} \\
&= \max\{1 + 0.3 \cdot 11.2 + 0.3 \cdot 4.3 + 0.4 \cdot 0, -1 + 0.5 \cdot 11.2 + 0.3 \cdot 4.3 + 0.2 \cdot 0\} \\
&= \max\{5.65, 5.89\} = 5.89
\end{aligned}
$$

By paying attention to the index of the largest argument in the $\max_a$ above, we can immediately recover that $\pi_1(s_1) = G(V_1)(s_1) = \arg\max_a\{R(s_1, a) + \sum_{a'} P(s, a, s')V_1(s')\} = a_1$ and similarly that $\pi_1(s_2) = G(V_1)(s_2) = a_2$. We determine $\pi_2(s_1) = G(V_2)(s_1) = \arg\max_a\{q_2(s_1, a)\}$ by computing $\max\{8+0.2\cdot 12.82 + 0.6 \cdot 5.89 + 0.2 \cdot 0, 10 + 0.1 \cdot 12.82 + 0.2 \cdot 4.3 + 0.7 \cdot 0\} = \max\{14.098, 12.142\} = 14.098$ hence $\pi_2(s_1) = a_1$. Likewise $\pi_2(s_2) = G(V_2)(s_2) = \arg\max_a\{q_2(s_2, a)\} = a_2$ since $\max\{1 + 0.3 \cdot 12.82 + 0.3 \cdot 5.89 + 0.4 \cdot 0, -1 + 0.5 \cdot 12.82 + 0.3 \cdot 4.3 + 0.2 \cdot 0\} = \max\{6.613, 6.7\} = 6.7$.

Looking at the linear combination that is used to calculate $q_i(s, a)$, we infer that the term associated with a_1 will grow faster than the term associated with a_2 for s_1, and vice versa for s_2. So the optimal policy is indeed $\pi^*(s_1) = a_1, \pi^*(s_2) = a_2$

Problem 4

See the file `simple_two_store_inventory_mdp_cap.py` in directory `assignment4` for the implementation.

In my code, I represent state as a 4-tuple (H_1, O_1, H_2, O_2) of on-hand (H_i) and on-order (O_i) inventory for stores 1 and 2 $(i = 1, 2)$. Each action takes the form (A_1, A_2, T) where A_i represents the inventory ordered to store i and T denotes the net inventory transferred from store 2 to store 1. Letting C_i denote the capacity for store i, we have the following constraints:

$$H_1 + O_1 + A_1 + T \leq C_1 \tag{1}$$
$$H_2 + O_2 + A_2 - T \leq C_2 \tag{2}$$

which accounts for the logic of the `get_action_transition_reward_map` method of the implemented class `SimpleTwoStoreInventoryMDPCap`.

Something I noticed and am unsure about is the calculation on page 85 of the class book. In general I do not see why we have the equality

$$\sum_j = k^\infty j \cdot f(j) = \lambda(1 - F(j-2))$$

where f, F are the pmf and cdf of a $Pois(\lambda)$ distribution respectively, since

$$1 - F(x) = \sum_{j=x+1}^{\infty} f(j) \neq \sum_{j=x+1}^{\infty} jf(j)$$

in general. This would mean that the calculation of the rewards are wrong. Have I misunderstood, or is this criticism legitimate?