Assignment 4

Miao-Chin Yen

March 7, 2022

Problem 1 (Manual Value Iteration)

1. Initialize the Value Function for each state to be it's max (over actions) reward, i.e., we initialize the Value Function to be $v_0(s_1) = 10.0, v_0(s_2) = 1.0, v_0(s_3) = 0.0$. Then manually calculate $q_k(\cdot,\cdot)$ and $v_k(\cdot)$ from $v_{k-1}(\cdot)$ using the Value Iteration update, and then calculate the greedy policy $\pi_k(\cdot)$ from $q_k(\cdot,\cdot)$ for $k = 1$ and $k = 2$ (hence, 2 iterations).

$$q_1(s_1, a_1) = \mathcal{R}(s_1, a_1) + \mathcal{P}(s_1, a_1, s_1) \cdot v_0(s_1) + \mathcal{P}(s_1, a_1, s_2) \cdot v_0(s_2) = 10.6$$

$$q_1(s_1, a_2) = \mathcal{R}(s_1, a_2) + \mathcal{P}(s_1, a_2, s_1) \cdot v_0(s_1) + \mathcal{P}(s_1, a_2, s_2) \cdot v_0(s_2) = 11.2$$

$$v_1(s_1) = max(q_1(s_1, a_1), q_1(s_1, a_2)) = 11.2 \Rightarrow \pi_1(s_1) = a_2$$

$$q_1(s_2, a_1) = \mathcal{R}(s_2, a_1) + \mathcal{P}(s_2, a_1, s_1) \cdot v_0(s_1) + \mathcal{P}(s_2, a_1, s_2) \cdot v_0(s_2) = 4.3$$

$$q_1(s_2, a_2) = \mathcal{R}(s_2, a_2) + \mathcal{P}(s_2, a_2, s_1) \cdot v_0(s_1) + \mathcal{P}(s_2, a_2, s_2) \cdot v_0(s_2) = 4.3$$

$$v_1(s_2) = max(q_1(s_2, a_1), q_1(s_2, a_2)) = 4.3 \Rightarrow \pi_1(s_2) = a_1$$

$$q_2(s_1, a_1) = \mathcal{R}(s_1, a_1) + \mathcal{P}(s_1, a_1, s_1) \cdot v_1(s_1) + \mathcal{P}(s_1, a_1, s_2) \cdot v_1(s_2) = 12.82$$

$$q_2(s_1, a_2) = \mathcal{R}(s_1, a_2) + \mathcal{P}(s_1, a_2, s_1) \cdot v_1(s_1) + \mathcal{P}(s_1, a_2, s_2) \cdot v_1(s_2) = 11.98$$

$$v_1(s_1) = max(q_2(s_1, a_1), q_2(s_1, a_2)) = 12.82 \Rightarrow \pi_2(s_1) = a_1$$

$$q_2(s_2, a_1) = \mathcal{R}(s_2, a_1) + \mathcal{P}(s_2, a_1, s_1) \cdot v_1(s_1) + \mathcal{P}(s_2, a_1, s_2) \cdot v_1(s_2) = 5.65$$

$$q_2(s_2, a_2) = \mathcal{R}(s_2, a_2) + \mathcal{P}(s_2, a_2, s_1) \cdot v_1(s_1) + \mathcal{P}(s_2, a_2, s_2) \cdot v_1(s_2) = 5.89$$

$$v_2(s_2) = max(q_2(s_2, a_1), q_2(s_2, a_2)) = 5.89 \Rightarrow \pi_2(s_2) = a_2$$

2. Now argue that $\pi_k(\cdot)$ for $k > 2$ will be the same as $\pi_2(\cdot)$. Hint: You can make the argument by examining the structure of how you get $q_k(\cdot,\cdot)$ from $v_{k-1}(\cdot)$. With this argument, there is no need to go beyond the two iterations you performed above, and so you can establish $\pi_2(\cdot)$ as an Optimal Deterministic Policy for this MDP.

$$q_k(s_1, a_1) - q_k(s_1, a_2)$$

$$= \mathcal{R}(s_1, a_1) - \mathcal{R}(s_1, a_2) + (\mathcal{P}(s_1, a_1, s_1) - \mathcal{P}(s_1, a_2, s_1)) \cdot v_{k-1}(s_1) + (\mathcal{P}(s_1, a_1, s_2) - \mathcal{P}(s_1, a_2, s_2)) \cdot v_{k-1}(s_2)$$

$$= -2.0 + 0.1 \cdot v_{k-1}(s_1) + 0.4 \cdot v_{k-1}(s_2),$$

$$q_k(s_2, a_2) - q_k(s_2, a_1)$$

$$= \mathcal{R}(s_2, a_2) - \mathcal{R}(s_2, a_1) + (\mathcal{P}(s_2, a_2, s_1) - \mathcal{P}(s_2, a_1, s_1)) \cdot v_{k-1}(s_1) + (\mathcal{P}(s_2, a_2, s_2) - \mathcal{P}(s_2, a_1, s_2)) \cdot v_{k-1}(s_2)$$

$$= -2.0 + 0.2 \cdot v_{k-1}(s_1)$$

Because $v_{k-1}(s_1) \geq v_2(s_1)$ and $v_{k-1}(s_2) \geq v_2(s_2) \forall k \geq 3$,

$$q_k(s_1, a_1) - q_k(s_1, a_2) \geq -2.0 + 0.1 \cdot v_2(s_1) + 0.4 \cdot v_2(s_2) > 0 \ \forall k \geq 3$$

$$q_k(s_2, a_2) - q_k(s_2, a_1) \geq -2.0 + 0.2 \cdot v_2(s_1) > 0 \ \forall k \geq 3$$

Hence $q_k(s_1, a_1) > q_k(s_1, a_2)$ and $q_k(s_2, a_2) > q_k(s_2, a_1) \ \forall k \geq 3 \Rightarrow \pi_k(s_1) = a_1$ and $\pi_k(s_2) = a_2 \ \forall k \geq 3$

Problem 4 (Two-Stores Inventory Control)

We model this as a finite markov decision process. Notation are as follows:
$\alpha_A :=$ on-hand inventory for store A, $\beta_A :=$ on-order inventory for store A
$\alpha_B :=$ on-hand inventory for store B, $\beta_B :=$ on-order inventory for store B
$h_A :=$ holding cost for store A, $h_B :=$ holding cost for store B (per unit of overnight inventory)
$p_A :=$ stockout cost for store A, $p_B :=$ holding cost for store B (per unit of missed demand)
$C_A :=$ shelf capacity for store A, $C_B :=$ shelf capacity for store B
$\lambda_A :=$ poisson distribution parameter of deamd of store A
$\lambda_B :=$ poission distribution parameter of demand of store B
$K_1 :=$ transportation cost from supplier to stores per order, $K_2 :=$ transportation cost between two stores
$\theta_A :=$ order quantity for store A, $\theta_B :=$ order quantity for store B
$\theta_E :=$ transported quantity between two stores. (If $\theta_E \leq 0$, which means we transport from store A to store B; otherwise, store B to store A)
$f(\cdot) :=$ PMF of demand, $F(\cdot) :=$ CMF of demand

Let $\mathcal{S} = \{(\alpha_A, \beta_A, \alpha_B, \beta_B) : 0 \leq \alpha_A + \beta_A \leq C_A, 0 \leq \alpha_B + \beta_B \leq C_B\}$ characterize the state space.

Let $\mathcal{A}((\alpha_A, \beta_A, \alpha_B, \beta_B)) = \{(\theta_A, \theta_B, \theta_E) : max\{-\alpha_A, -(C_B - (\alpha_B + \beta_B)\} \leq \theta_E \leq \min\{\alpha_B, C_A - (\alpha_A + \beta_A)\}, 0 \leq \theta_A \leq C_A - (\alpha_A + \beta_A + \theta_E), 0 \leq \theta_B \leq C_B - (\alpha_B + \beta_B - \theta_E)\}$ characterize the action space.
Note in the action space, we need to consider the amount of inventories we can move between two stores which is related to the capacity of the stores, on-hand inventory and on-order inventory. Also, we need to notice that the inventories transported between stores would arrive faster (overnight). We suppose we decide the inventories to be transported between two stores first and then decide the inventories to purchase from supplier to the two stores.

The reward transition $\mathcal{R}_T(s, a, s')$ is as follows:

$$\mathcal{R}_T((\alpha_A, \beta_A, \alpha_B, \beta_B), (\theta_A, \theta_B, \theta_E), (\alpha_A + \beta_A + \theta_E - i_A, \theta_A, \alpha_B + \beta_B - \theta_E - i_B, \theta_B)) =$$

$$-h_A\alpha_A - h_B\alpha_B - K_2\mathbb{1}_{\{\theta_E \neq 0\}} - K_1\mathbb{1}_{\{\theta_A > 0\}} - K_1\mathbb{1}_{\{\theta_B > 0\}} - h_A\theta_E\mathbb{1}_{\{\theta_E < 0\}} + h_B\theta_E\mathbb{1}_{\{\theta_E > 0\}}$$

for $0 \leq i_A \leq \alpha_A + \beta_A + \theta_E - 1$, $0 \leq i_B \leq \alpha_B + \beta_B - \theta_E - 1$.
Let $R_c = -h_A\alpha_A - h_B\alpha_B - K_2\mathbb{1}_{\{\theta_E \neq 0\}} - K_1\mathbb{1}_{\{\theta_A > 0\}} - K_1\mathbb{1}_{\{\theta_B > 0\}} - h_A\theta_E\mathbb{1}_{\{\theta_E < 0\}} + h_B\theta_E\mathbb{1}_{\{\theta_E > 0\}}$ be the base reward. The base is about the holding cost and transportation cost.
Note that when we transport from one store to another store, this would happen during midnight. At the same time, the transported inventories would not have holding cost to two stores.

The other reward transition function are stated as follows:

$$\mathcal{R}_T((\alpha_A, \beta_A, \alpha_B, \beta_B), (\theta_A, \theta_B, \theta_E), (0, \theta_A, \alpha_B + \beta_B - \theta_E - i_B, \theta_B)) =$$

$$R_C - p_A\Big(\sum_{j=\alpha_A+\beta_A+\theta_E+1}^{\infty} f_{\lambda_A}(j) \cdot (j - (\alpha_A + \beta_A + \theta_E))\Big)$$

for $0 \leq i_B \leq \alpha_B + \beta_B - \theta_E - 1$;

$$\mathcal{R}_T((\alpha_A, \beta_A, \alpha_B, \beta_B), (\theta_A, \theta_B, \theta_E), (\alpha_A + \beta_A + \theta_E - i_A, \theta_A, 0, \theta_B)) =$$

$$R_C - p_B\Big(\sum_{j=\alpha_B+\beta_B-\theta_E+1}^{\infty} f_{\lambda_B}(j) \cdot (j - (\alpha_B + \beta_B - \theta_E))\Big)$$

for $0 \leq i_A \leq \alpha_A + \beta_A + \theta_E - 1$

$$\mathcal{R}_T((\alpha_A, \beta_A, \alpha_B, \beta_B), (\theta_A, \theta_B, \theta_E), (0, \theta_A, 0, \theta_B)) =$$

$$-R_C - p_A\Big(\sum_{j=\alpha_A+\beta_A+\theta_E+1}^{\infty} f_{\lambda_A}(j)\cdot(j-(\alpha_A+\beta_A+\theta_E)) - p_B\Big(\sum_{j=\alpha_B+\beta_B-\theta_E+1}^{\infty} f_{\lambda_B}(j)\cdot(j-(\alpha_B+\beta_B-\theta_E))\Big)$$

Reference simple_inventory_mdp_cap_two _stores.py.
We found that if the transportation cost between two stores are high, we would tend to order from supplies, and vice versa. Also, if the holding cost for a store is high, we would avoid having too many inventories to be on-hand. Furthermore, if capacity is high enough, we would like to order to prevent stockout which in reality would occur high cost.
Maybe my problem formulation can be improved. For the transition probability in the coding part, maybe I can write a more readable one.