Foundations of Inventory Management **Fall 2022, NUS**

Lecture 2: Markov Decision Process

Lecturer: Zhenyu Hu

2.1 Some General Concepts

A Markov process is defined by a pair $(\mathcal{X}, (p_{ij})_{i,j\in\mathcal{X}})$, where $\mathcal{X}$ is called the state space. Let $X_1, ..., X_T$ are $\mathcal{X}$-valued random variables. The matrix (p_{ij}) then specifies the transition probability, i.e.,

$$p_{ij} = \mathbb{P}(X_{t+1} = j | X_t = i).$$

Markov decision process (MDP) then allows the transition probability to influenced by some actions. Let $\mathcal{A}$ denote the action space, and $\mathcal{A}(x) \subseteq \mathcal{A}$ be the set of feasible actions at state x. The transition probability from state i to state j under action a is then specified by

$$p_{ij}(a) = \mathbb{P}(X_{t+1} = j | X_t = i, a_t = a).$$

In many cases, the transition of the state is represented by a dynamic system of the following form:

$$x_{t+1} = f(x_t, a_t, \xi_t),$$

where ξ_t is a random variable whose distribution possibly depends on (x_t, a_t). The two representations are equivalent. That is, given $p_{ij}(a)$, we can write the dynamic system equation

$$x_{t+1} = \xi_t,$$

where $\mathbb{P}(\xi_t = j | X_t = i, a_t = a) = p_{ij}(a)$. Conversely, given a dynamic system equation $x_{t+1} = f(x_t, a_t, \xi_t)$, by letting

$$\Xi_j(i, a) = \{\xi | f(i, a, \xi) = j\},$$

be all realizations of random noises that lead to state j under action a and state i, then we can define

$$p_{ij}(a) = \mathbb{P}(\xi \in \Xi_j(i, a) | x_t = i, a_t = a).$$

Every period, there is a cost $c(x_t, a_t)$ that depends on the current period's state x_t and the action a_t chosen in period t. We associate a terminal cost $c_T(x_T)$ at the end of the horizon.

An admissible policy π is a sequence of functions:

$$\pi = \{\mu_0(\cdot), ..., \mu_{N-1}(\cdot)\},$$

with $\mu_t(\cdot)$ being a function that maps the state x_t to an action $\mu_t(x_t) \in \mathcal{A}(x_t)$. The total expected cost under a policy π can then be computed as

$$J^\pi(x_0) = \mathbb{E}\left[\sum_{t=0}^{T-1} c(x_t, \mu_t(x_t)) + c_T(x_T)\right].$$

The optimization problem is then defined as

$$J^*(x_0) = \min_{\pi\in\Pi} J^\pi(x_0) = \min_{\pi\in\Pi} \mathbb{E}\left[\sum_{t=0}^{T-1} c(x_t, \mu_t(x_t)) + c_T(x_T)\right], \tag{2.1}$$

where Π is the set of all admissible policies.

Suppose $\mathcal{A}(x_t) = \mathcal{A}$. It would be useful to compare the formulation of (2.1) with the following formulation that is commonly used in stochastic programming:

$$\min_{a_t \in \mathcal{A}} \mathbb{E}\left[\sum_{t=0}^{T-1} c(x_t, a_t) + c_T(x_T)\right].$$

We sometimes refer to the solution to problem (2.1) as a close-loop solution while the one above as an open-loop solution. The difference in their values is referred to as the value of information.

The following result shows that the optimal solution to problem (2.1) can obtained recursively via dynamic programming.

Theorem 2.1 *Let $J_0, J_1, ..., J_T$ be functions defined on $\mathcal{X}$ recursively by*

$$J_t(x) = \min_{a \in \mathcal{A}(x)} \{c(x, a) + \mathbb{E}[J_{t+1}(f_t(x, a, \xi_t)]\},$$

for $t = 0, 1..., T-1$, and $J_T(x) = c_T(x)$, and $a_t^*(x)$ be the corresponding optimal solution to the above problem. Then,

$$J^*(x_0) = J^{\pi^*}(x_0) = J_0(x_0),$$

where $\pi^* = \{a_0^*(\cdot), ..., a_{T-1}^*(\cdot)\}$.

Proof: For the case when the state space $\mathcal{X}$ and the action space $\mathcal{A}$ are finite, see the proof of Theorem 5.1 in Porteus (2002) or Proposition 1.3.1 in Bertsekas (2012).

For general state and action spaces, additional measurability conditions are needed and one is referred to Chapter 3 in Hernández-Lerma and Lasserre (2012). ■

The recursive equation defined in Theorem 2.1 is referred to as the Bellman equation.

2.2 Examples

2.2.1 $0-1$ knapsack problem

Given n items, indexed by $i \in \{0, ..., n-1\}$. Item i has value v_i and weight w_i. The capacity of the knapsack is W. The problem can be formulated as an integer program as below:

$$\begin{aligned} \max \ & \sum_{i=0}^{n-1} v_i a_i \\ \text{s.t.} \ & \sum_{i=0}^{n-1} w_i a_i \leq W, \\ & a_i \in \{0, 1\}, i = 0, ..., n-1. \end{aligned}$$

One can also formulate the problem as a dynamic program. Consider a sequential decision making process in which we pick up the item one by one from 0 to $n-1$ and decide sequentially for each one of them whether we put it into the knapsack or not. Each of the element we discussed in the general MDP can be specified as below.

- State: We define the state x_i in "period i" as the remaining capacity in the knapsack right before we pick up the i-th item, which takes values in the state space $\mathcal{X} = \{0, 1, ..., W\}$.
- Action: The action space is simply $\mathcal{A} = \{0, 1\}$ and given the current state x, the set of feasible action is $\mathcal{A}_i(x) = \{a \in \{0, 1\} | w_i a \leq x\}$.
- State transition: $x_{i+1} = x_i - w_i a$.
- Per-period reward: $r_i(x_i, a_i) = v_i a_i$.

We can then write the Bellman equation of the problem as

$$\begin{aligned} J_i(x_i) &= \max_{a \in \mathcal{A}_i(x_i)} \{r_i(x_i, a_i) + J_{i+1}(x_i - w_i a_i)\} \\ &= \max_{\substack{w_i a \leq x_i \\ a \in \{0,1\}}} \{v_i a + J_{i+1}(x_i - w_i a_i)\}, \end{aligned}$$

with the terminal condition $J_n(x_n) = 0$. Note that for deterministic problem, the states are completely determined by the actions and hence can be viewed as decision variables. One would then arrive at the following equivalent formulation:

$$\begin{aligned} J^*(x_0) = \max_{a,x} \quad & \sum_{i=0}^{n-1} v_i a_i \\ \text{s.t. } & w_i a_i \leq x_i, i = 0, ..., n-1 \\ & x_{i+1} = x_i - w_i a_i, i = 0, ..., n-1 \\ & a_i \in \{0, 1\}, x_0 = W. \end{aligned}$$

2.2.2 Secretary problem

Consider n applicants to a job position. The employer is able to rank the candidates from the best to the worst if all applicants are seen at the same time. In the secretary problem, however, the employer is only able to interview the applicants sequentially, and has to make an offer or rejection decision on the spot. Rejected applicants cannot be recalled. Suppose that the applicants are interviewed in random order, indexed by $i = 1, ..., n$, with each order being equally likely and the employer is seeking to maximize the probability of choosing the best applicant.

The problem again can be modeled using the general MDP framework as below.

- State: Let $\mathcal{X} = \{-1, 0, 1\}$. We use $x_i = -1$ to denote the termination state, meaning that an offer has already been made prior to "interviewing" applicant i. We use $x_i = 1$ to encode the information that applicant i is the best among the candidates seen so far, and $x_i = 0$ otherwise.
- Action: The action space is $\mathcal{A} = \{0, 1\}$ with $a = 0$ meaning rejection and $a = 1$ meaning making an offer. Note that $\mathcal{A}(x) = \mathcal{A}$ if $x \in \{0, 1\}$ but $\mathcal{A}(x) = \{0\}$ for $x = -1$.
- State transition:

$$\begin{aligned} &\mathbb{P}(x_{i+1} = -1 | x_i = -1, a) = 1 \\ &\mathbb{P}(x_{i+1} = -1 | x_i, a = 1) = 1 \\ &\mathbb{P}(x_{i+1} = 1 | x_i, a = 0) = \frac{1}{i+1}, \ x_i \neq -1 \\ &\mathbb{P}(x_{i+1} = 0 | x_i, a = 0) = \frac{i}{i+1}, x_i \neq -1 \end{aligned}$$

.

- Per-period reward:

$$r_i(x,a) = \begin{cases} 0 & a = 0 \text{ or } x \in \{-1, 0\} \\ \frac{i}{n} & a = 1, x = 1. \end{cases}.$$

The Bellman equation then follows as

$$J_i(x_i) = \max_{a \in \mathcal{A}(x_i)} \{r_i(x_i, a) + \mathbb{E}[J_{i+1}(x_{i+1})]\}$$

with the terminal condition $J_{n+1}(x) = 0$. We can further explicitly write the Bellman equation by discussing each of the state separately:

$$\begin{aligned} &J_i(-1) = 0; \\ &J_i(0) = \max\{\underbrace{0}_{a=1}, \underbrace{\frac{1}{i+1} J_{i+1}(1) + \frac{i}{i+1} J_{i+1}(0)}_{a=0}\} = \frac{1}{i+1} J_{i+1}(1) + \frac{i}{i+1} J_{i+1}(0); \\ &J_i(1) = \max\{\underbrace{\frac{i}{n}}_{a=1}, \underbrace{\frac{1}{i+1} J_{i+1}(1) + \frac{i}{i+1} J_{i+1}(0)}_{a=0}\} = \max\{\frac{i}{n}, J_i(0)\}. \end{aligned}$$

Now we analyze the Bellman equation to obtain insights on the optimal solution. First observe that $J_i(1) \geq J_i(0)$ for all i, which then implies

$$J_i(0) = \frac{1}{i+1} J_{i+1}(1) + \frac{i}{i+1} J_{i+1}(0) \geq J_{i+1}(0).$$

This conveys the intuitive message that the value of continuing interviewing is decreasing as there are fewer applicants left. Now, since i/n is increasing in i and $J_i(0)$ is decreasing in i, there must exist a threshold i^* such that

$$a_i^*(1) = \begin{cases} 1 & i \geq i^* \\ 0 & i < i^*. \end{cases}$$

To compute the optimal i^*, we first compute the employer's payoff under any threshold policy π_j with the threshold j such that

$$a_i(1) = \begin{cases} 1 & i \geq j \\ 0 & i < j. \end{cases}$$

In particular, we denote the payoff by $J^{\pi_j}(1)$ (note that $x_1 \equiv 1$) which can be computed as

$$\begin{aligned} J^{\pi_j}(1) &= \sum_{i=j}^{n} \mathbb{P}(\{i \text{ is made an offer}\} \cap \{i \text{ is the best applicant}\}) \\ &= \sum_{i=j}^{n} \mathbb{P}(\{i \text{ is made an offer}\} | \{i \text{ is the best applicant}\}) \cdot \mathbb{P}(\{i \text{ is the best applicant}\}) \\ &= \sum_{i=j}^{n} \frac{j-1}{i-1} \cdot \frac{1}{n} \\ &= \frac{j-1}{n} \sum_{i=j}^{n} \frac{1}{i-1}. \end{aligned}$$

Note that

$$\begin{aligned} J^{\pi_{j+1}}(1) - J^{\pi_j}(1) &= \frac{j}{n} \sum_{i=j+1}^{n} \frac{1}{i-1} - \frac{j-1}{n} \left(\frac{1}{j-1} + \sum_{i=j+1}^{n} \frac{1}{i-1} \right) \\ &= \frac{1}{n} \sum_{i=j+1}^{n} \frac{1}{i-1} - \frac{1}{n}, \end{aligned}$$

which is decreasing in j. Hence,

$$i^* = \min \left\{ j : \sum_{i=j+1}^{n} \frac{1}{i-1} - 1 \leq 0 \right\}.$$

We can further analyze the asymptotic behavior of the optimal policy as $n \to \infty$. To this end, we let $z = (j-1)/n$ be the fraction of rejected applicants, and

$$\begin{aligned} f_n(z) &= z \cdot \sum_{i=zn+1}^{n} \frac{1}{i-1} \\ &= \frac{z}{n} \sum_{i=zn+1}^{n} \frac{1}{(i-1)/n} \\ &= \frac{z}{n} \left(\frac{1}{z} + \frac{1}{z+1/n} + \frac{1}{z+2/n} + ... + \frac{1}{1-1/n} \right). \end{aligned}$$

Therefore,

$$\lim_{n \to \infty} f_n(z) = z \int_z^1 \frac{1}{x} dx = -z \log(z).$$

The optimal fraction of rejected applications $z^* = 1/e$ and the employer's asymptotic payoff is $-z^* \log(z^*) = 1/e$.

Now, compare this with the open-loop solution where one solves

$$\max_{a_1, ..., a_n} \mathbb{E} \left[\sum_{i=1}^{n} r_i(x_i, a_i) \right].$$

Clearly with $a_j = 1$ and $a_i = 0, j \neq i$, we can compute the objective to be

$$\mathbb{E}\left[r_j(x_j, 1) \right] = \frac{1}{j} \cdot \frac{j}{n} = \frac{1}{n}.$$

Note that asymptotically, the expected payoff under the open-loop policy goes to zero while that under the closed-loop policy is $1/e$, which indicates significant value in information.

References

Bertsekas, D. (2012). *Dynamic programming and optimal control: Volume I*, Volume 1. Athena scientific.

Hernández-Lerma, O. and J. B. Lasserre (2012). *Discrete-time Markov control processes: basic optimality criteria*, Volume 30. Springer Science & Business Media.

Porteus, E. L. (2002). *Foundations of stochastic inventory theory*. Stanford University Press.