INSE6300 Jia Yuan Yu

3: Inventory management

Concordia February 9, 2016

Supply chain management is about making a sequence of decisions over a sequence of time steps, after making observations at each of these time steps. We illustrate this with the problem of managing an inventory of nonperishable goods when demand is stochastic.

We first need to introduce the notion of Markov chains (e.g., weather model, coupon collector, etc.).

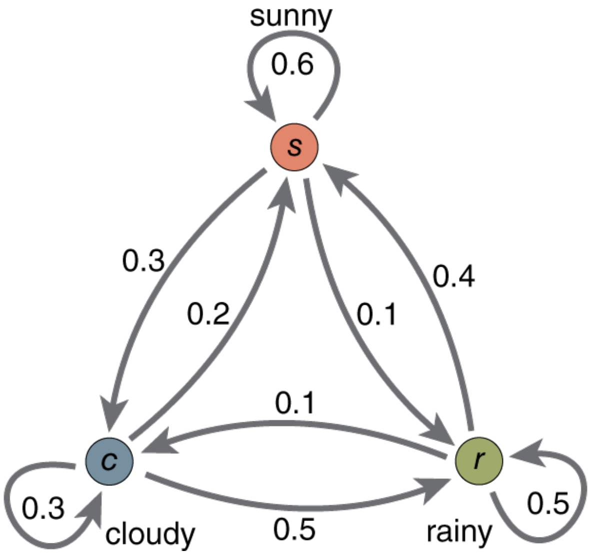

Figure 1: From `http://bit-player.org/wp-content/extras/markov/art/weather-model.png`

1 Numerical example

Consider the last horse-drawn carriage dealer in the world, which just took a lease of $N = 4$ months on a showroom. At time $t = 1, 2, 3, 4$, the starting inventory level is s_t, the order size is a_t, and the random demand is d_t. The random variables $\{d_t\}$ are i.i.d. with the following distribution:

$$d_t = \begin{cases} 0 & \text{w.p. } 0.1, \\ 1 & \text{w.p. } 0.7, \\ 2 & \text{w.p. } 0.2. \end{cases}$$

Figure 2: From `exportersindia.com`

The inventory level at the next time step $t+1$ evolves according to the following Markov process:

$$s_{t+1} = \begin{cases} 0 & \text{if } s_t + a_t - d_t < 0, \\ s_t + a_t - d_t & \text{if } 0 \leq s_t + a_t - d_t \leq 2, \\ 2 & \text{if } s_t + a_t - d_t > 2. \end{cases} \tag{1}$$

Suppose that the ordering cost of each unit of inventory is 1, and that both the holding and backorder costs are quadratic, the overall cost is

$$c(s_t, a_t, d_t) = a_t + (s_t + a_t - d_t)^2.$$

Unsold inventory has no salvage value.

Backward induction starts at $t = N = 4$, and assigns a value to each inventory state. Since unsold inventory has no salvage value, we have

$$V_4(0) = V_4(1) = V_4(2) = 0.$$

For $t = 1, 2, 3$, we assign the following values to the states:

$$\begin{aligned} V_t(s) &= \min_a \mathbb{E}\{c(s, a, d_t) + V_{t+1}(s_{t+1})\} \\ &= \min_a \mathbb{E}\{c(s, a, d_t) + V_{t+1}([s + a - d_t]_0^2)\}, \end{aligned}$$

where we used the definition of s_{t+1} in (1) and the notation $[\cdot]_0^2$ to clamp to state to the allowed range. Namely, for $t = 3$, using the distribution of d_t above, we obtain

$$\begin{aligned} V_3(s) &= \min_a \mathbb{E}\{c(s, a, d_3) + V_4(s_4)\} \\ &= \min_a \mathbb{E}\{a + (s + a - d_t)^2 + 0\} \\ &= \min_a \Big(a + 0.1(s + a - 0)^2 + 0.7(s + a - 1)^2 + 0.2(s + a - 2)^2\Big), \end{aligned}$$

and

$$V_3(s) = \begin{cases} 1.3 & \text{if } s = 0, \\ 0.3 & \text{if } s = 1, \\ 1.1 & \text{if } s = 2. \end{cases}$$

The optimal order sizes are

$$a_3^*(s) = \begin{cases} 1 & \text{if } s = 0, \\ 0 & \text{if } s = 1, \\ 0 & \text{if } s = 2. \end{cases}$$

Repeat for $t = 2$ and $t = 1$:

$$V_2(s) = \begin{cases} 2.5 & \text{if } s = 0, \\ 1.5 & \text{if } s = 1, \\ 1.68 & \text{if } s = 2. \end{cases}$$

$$a_2^*(s) = \begin{cases} 1 & \text{if } s = 0, \\ 0 & \text{if } s = 1, \\ 0 & \text{if } s = 2. \end{cases}$$

$$V_1(s) = \begin{cases} 3.7 & \text{if } s = 0, \\ 2.7 & \text{if } s = 1, \\ 2.818 & \text{if } s = 2. \end{cases}$$

$$a_1^*(s) = \begin{cases} 1 & \text{if } s = 0, \\ 0 & \text{if } s = 1, \\ 0 & \text{if } s = 2. \end{cases}$$

Observe that the optimal order size is 1 if the current inventory is 0, and 0 otherwise.

2 Stochastic inventory management

Consider a single product (e.g., cars), and discrete time steps (e.g., months 1, 2, etc.). Every time step (e.g., every month), the decision maker oberserves the current inventory level, and decides how much inventory to order from the supplier. There are costs for holding inventory. The demand is random, but we know the distribution of the random variable. The goal is maximize the expected value of the profit (revenue minus costs) over a number N of months.

Assumptions:

- Delivery is instantaneous (no lead-time);
- The demand take integer values;

- The demand is i.i.d. with given distribution $p_j = \mathbb{P}(D_t = j)$ for $j = 0, 1, \ldots$;
- Inventory has a capacity M.

For time steps $t = 1, 2, \ldots$, let s_t denote the inventory level, a_t the order size, and D_t the demand at time t—these are all integer-valued. The inventory level from one time step to the next follows this dynamics:

$$s_{t+1} = \max\{s_t + a_t - D_t, 0\}.$$

The reward or profit at time t is

$$\begin{aligned} r_t(s_t, a_t) &= \underbrace{F(s_t + a_t)}_{\text{present value of inventory}} - \underbrace{O(a_t)}_{\text{order}} - \underbrace{h(s_t + a_t)}_{\text{holding}}, \quad \text{for } t = 1, \ldots, N-1, \\ r_N(s_N, a_N) &= \underbrace{g(s_N, a_N)}_{\text{salvage value}} . \end{aligned}$$

where the expected present value of inventory is

$$F(z) = \sum_{j=0}^{z-1} \underbrace{f(j)}_{\text{revenue from } j \text{ sales}} p_j + \sum_{j \geq z} \underbrace{f(z)}_{\text{revenue capped to } z \text{ sales}} p_k, \quad \text{for } z = 0, 1, \ldots$$

The order and holding cost function can be arbitrary; for instance, $O(z) = [K + c(z)]1_{[z>0]}$.

Remark 1. Backorder costs (missed sales) are implicitly accounted for in the profit.

2.1 MDP

We can describe the stochastic inventory management problem as an MDP. The inputs are:

- Holding cost function h, order cost O, sales revenue f, salvage revenue g;
- Probabilities $p_0, p_1, \ldots$;
- Time horizon: $\{1, 2, \ldots, N\}$;
- State space: $S = \{0, 1, \ldots, M\}$;
- Action space: $A = \{0, 1, \ldots, M\}$;
- Expected reward: $r_1, r_2, \ldots, r_N$;
- State transition probabilities:

$$P(s' \mid s, a) = \begin{cases} 0 & \text{if } s' \in (s+a, M], \\ p_{s+a-s'} & \text{if } s' \in (0, s+a] \text{ and } s+a \leq M, \\ \sum_{k>s+a} p_k & \text{if } s' = 0 \text{ and } s+a \leq M. \end{cases}$$

This is the probability of having an inventory level s' at the next time step when the inventory level at the current time step is s and we order a units of inventory.

The output is a optimal sequence of policies $\sigma_1, \sigma_2, \ldots$, where $\sigma_j : S \to A$. These policies are used to pick the optimal action to take at each time step: suppose that at time $t = 1, 2, \ldots, N$, we observe the state s_t (a random variable), then the optimal action is $\sigma_t(s_t)$.

Remark 2 (Reward vs cost). We can define the MDP in terms of costs by replacing the expected reward by expected cost, as in the numerical example above, and by replacing the max by min.

2.2 Solving finite-horizon MDP by backward induction

How do we compute the optimal policies $\sigma_1, \sigma_2, \ldots$? We propose a method of dynamic programming called backward induction.

The backward induction algorithm for MDPs proceeds as follows.

1. Set $j = N$, and $V_N(s) = \max_{a \in A} r_N(s, a) = g(s)$ for all $s \in S$;
2. For $j = N - 1, N - 2, \ldots, 1$:
    (a) For $s \in S$:
        i. Compute
$$V_j(s) = \max_{a \in A} \left\{ r_j(s, a) + \sum_{s' \in S} P(s' \mid s, a) V_{j+1}(s) \right\};$$
        ii. Output $\sigma_j(s) \in \arg\max_{a \in A} \left\{ r_j(s, a) + \sum_{s' \in S} P(s' \mid s, a) V_{j+1}(s) \right\}$.

The output policies $\sigma_1, \ldots, \sigma_N$ are optimal (cf. Puterman, Section 4.3).

3 References

- Markov Decision Processes, M. Puterman, Chapter 1.