INSE6290 Jia Yuan Yu

4: Stochastic Inventory, Queues

Concordia October 31, 2016

First, we review how to compute the least-squares forecast in the linear regression setting. Next, we consider two models of supply chain problems that involve Markov chains.

1 Stochastic inventory management

Figure 1: From `shutterstock.com`

The EOQ model has deterministic demand, the newboy model does not allow inventory carry-over from one time period to the next (perishable inventory). These problem are essentially one-decision problems. In this section, we model stochastic demand for inventory that is not perishable, where solving the problem requires a sequence of decisions. This model of stochastic inventory management is closer in spirit to the beer game.

Consider a single product, 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.

1.1 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 $\sigma_t(s_t)$.

1.2 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.

To illustrate how it works, consider first the game of tic-tac-toe and solving it by backward induction. Take the point of view of one player (e.g., the X-player), consider every possible board configuration with only one last move remaining for the X-player[1], record in a look-up table the outcome. Next, consider every possible board configuration with two moves remaining, find the best next move using the last look-up table, record the outcome corresponding to the best move in a new look-up table. Repeat until we are at the first move for the X-player. As another example, consider how to solve the shortest path problem by backward induction[2].

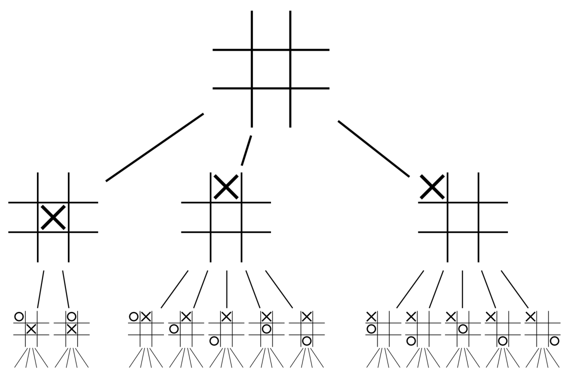

Figure 2: From `https://en.wikipedia.org/wiki/Tic-tac-toe`

The backward induction algorith 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

$$V_j(s) = \max_{a \in A} \left\{ r_j(s, a) + \sum_{s' \in S} P(s' \mid s, a) V_{j+1}(s) \right\};$$

[1] These are all board configurations where neither player has three squares in a row, and each player has made four moves.

[2] Cf. Chapter 11 of *Applied Mathematical Programming* by Bradley, Hax, and Magnanti (Addison-Wesley, 1977), `http://web.mit.edu/15.053/www/AMP-Chapter-11.pdf`.

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 $\sigma_1, \dots, \sigma_N$ are optimal (cf. Puterman, Section 4.3).

2 Queues

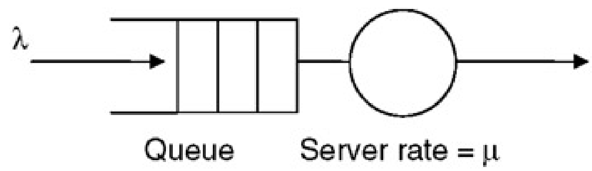

Figure 3: From Flylib

Queues appear supply chain design: communication networks, supermarkets, assembly lines, airports. Whenever you have customers or items arriving at one rate and departing at another rate, you have a queue. For instance, queues arise when stock arrive in a warehouse at a different rate than the demand—inventory is an example of queue. Queues also arise when customers arrive at random time instants and take a nonzero amount of time to serve and depart, which is not capture in the Bass model. Queueing models do capture the interaction between the arrival times and the service times of the supply chain. The arrivals can model jobs, phone calls, inventory items, etc. Sevice can model demand, sales, etc.

2.1 Queues as Markov chains

Many queueing models can be analyzed as Markov chains. For instance, in the $M/M/1$ queue model[3], customers arrive at random time instants, where the time interval between consecutive arrivals is exponential with parameter λ. The customers are served on a first-come first-served basis. The service times are exponential with parameter $1/\mu$. There is no limit on the number of customers in the queue.

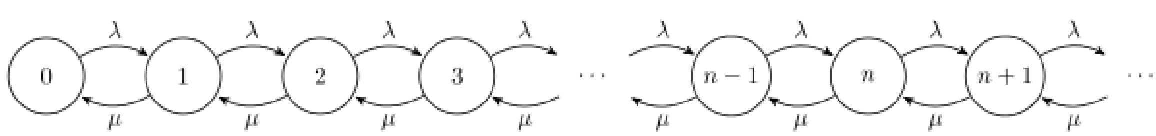

Figure 4: From `https://en.wikipedia.org/wiki/M/M/1_queue`

Example 2.1 ($M/D/1$ (D for deterministic service))**.** Consider deterministic service times of length τ. Let $X_0 = 0$ and let X_k denote the number of customers waiting

[3]M stands for Markovian: the arrival and the service processes are modeled as Poisson processes, which are Markovian.

in the queue when the k-th customer enters service. Let ξ_k denote the number of customers who arrived during the k-th customer's service time. Since the service times are fixed, and the time between arrivals are i.i.d. random variables, $\xi_1, \xi_2, \ldots$ are i.i.d. The queue length evolves as a Markov chain:

$$X_{k+1} = \max\{X_k + \xi_k - 1, 0\}.$$

Remark 2. Queues can also be controlled with actions and analyzed as MDPs (cf. Puterman, Section 3.7).

2.2 Quality of service, waiting time

When queues model customers, an important way of measuring quality of service is through the waiting times for customers. The waiting times are random variables. For instance, in the $M/D/1$ model, the arrival times are $t_1, t_2, \ldots$, where

$$\begin{aligned} t_1 &= 0, \\ t_i &= \sum_{j=1}^{i-1} e_j, \quad \text{for } i = 2, 3, \ldots, \end{aligned}$$

and $e_1, e_2, \ldots$ denote i.i.d. exponential random variables with parameter λ for the times between consecutive arrivals. Observe that the arrival times form a Markov process.

Suppose that it takes τ time units to serve each customer. Let $t_1 + d_1 + \tau, t_2 + d_2 + \tau, \ldots$ denote the departure times of customers, so that $d_1, d_2, \ldots$ are the durations of time that customers spend in the queue—i.e., their waiting times. Waiting times are also described by a Markov process[4]:

$$\begin{aligned} d_1 &= 0, \\ d_i &= \max\{d_{i-1} - \underbrace{(t_i - t_{i-1})}_{\text{difference in arrival times}} + \underbrace{\tau}_{\text{service time}}, 0\}, \quad i = 2, 3, \ldots. \end{aligned}$$

The probability distribution of the random variables d_i can be derived from first principles or estimated by simulation (by generating many samples of each d_i and counting empirical frequencies, as in Figure 2.2). The decision-maker can control this probability distribution by controling the parameters τ and λ, *e.g.*, by hiring more workers to reduce service time, by limiting the number of customers arriving in the queue through an invitation system, etc.

3 Reading material

- Chapters 1, 2, and 3 of Markov Decision Processes (Puterman).
- Chapter 11 of *Applied Mathematical Programming* by Bradley, Hax, and Magnanti (Addison-Wesley, 1977), `http://web.mit.edu/15.053/www/AMP-Chapter-11.pdf`.

[4] We say Markov chain for processes that take a finite number of values.

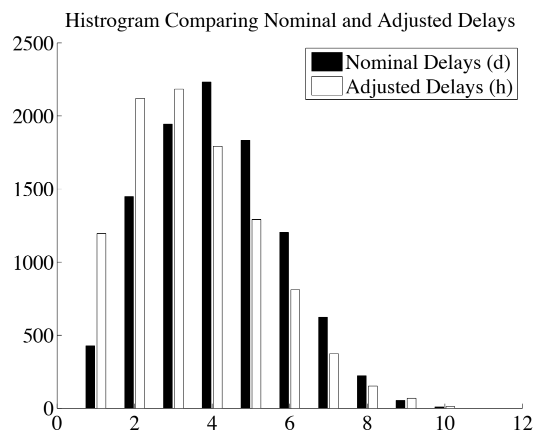

Figure 5: Simulated empirical frequency histogram for d_i given a fixed d_{i-i}.