Stochastic Models and Optimization: Problem Set 3

Roger Garriga Calleja, José Fernando Moreno Gutiérrez, David Rosenfeld, Katrina Walker

March 11, 2017

Q1. Inventory control with forecasts

(a): Single period

Primitives D: Demand, which is a random variable

$D = \begin{cases} P_L & \text{with probability q} \\ P_S & \text{with probability 1 - q} \end{cases}$

p: Sale price per unit

c: Ordering cost per unit

s: Salvage value per unit

$b = p - c$: Backorder cost per unit

$h = c - s$: Inventory holding cost per unit

Q: Inventory decision.

The profit for this problem can be written as:

$\pi(Q) = E[p.min\{D, Q\} - c.Q + s(Q - D)^+]$

$\pi(Q) = pE[D] - (p - c)E[(D - Q)^+] + (c - s)E[(Q - D)^+]$

$\pi(Q) = pE[D] - bE[(D - Q)^+] - hE[(Q - D)^+]$

We then expand the formula to account for the expectation of demand D which takes value P_L with probability q:

$\pi(Q) = pE[D] - b[q.E_{P_L}[(D-Q)^+] + (1-q).E_{P_S}[(D-Q)^+]] - h[qE_{P_L}[(Q-D)^+] + (1-q)E_{P_S}[(Q-D)^+]]$

We take the derivative of the profit with respect to quantity in order to minimise, and set it equal to zero (with P_L and P_S the probability distributions in the large and small demand cases respectively):

$$\pi'(Q^*) = -b[qP_L(D > Q^*) + (1 - q)P_S(D > Q^*)] + h[qP_L(Q^* > D) + (1 - q)P_S(Q^* > D) = 0$$

$$-bq(1 - P_L(D \leqslant Q^*) - b(1 - q)(1 - P_S(D \leqslant Q^*)) + hqP_L(D \leqslant Q^*) + h(1 - q)P_S(D \leqslant Q^*) = 0$$

We can then obtain that the optimal decision Q^* satisfies:

$$qP_L(D \leqslant Q^*) + (1 - q)P_S(D \leqslant Q^*) = \frac{b}{b + h}$$

(b) Multi-period problem

Primitives

x_k: Inventory level at period k

u_k: Decision variable over the number of units to order.
w_k: Demand at period k.
$y_{k+1} = \xi$ where ξ takes the value "large demand" with probability q:
$\xi_k = \begin{cases} L & \text{with probability q} \\ S & \text{with probability 1 - q} \end{cases}$
c: ordering cost per unit
p: Sale price per unit
s: Salvage value
$b = p - c$: Backorder cost per unit
$h = c - s$: Inventory holding cost per unit

Dynamics:
$x_{k+1} = x_k + u_k - w_k$
$y_{k+1} = \xi_k$
$g_N(x_N) = -s(x_N)$
$g_k(x_k, u_k, w_k) = cu_k - pw_k + bmax\{w_k - x_k - u_k, 0\} + hmax\{x_k + u_k - w_k\}$

DP algorithm: $J_N(x_N) = -s(x_N)$
$J_k(x_k) = \min_{u_k \geqslant 0} E_{w_k,\xi_k}[g_k(x_k, u_k, w_k) + J_{k+1}(x_k, u_k, w_k)]$

We can introduce the variable $z_k = x_k + u_k$, and rewrite our DP equation as follows:

$$J_k(x_k) = \min_{z_k \geqslant x_k} [G_k(z_k)] - cx_k$$

where $G_k(z_k) = cz_k + bE[max\{0, w_k - z_k\}] + hE[max\{0, z_k - w_k\}] + E[J_{k+1}(z_k - w_k)]$

Once again we can expand this expression to take into account the two different values of ξ for large and small demand:
$G_k(z_k) = cz_k + bqE[max\{0, w_k - z_k\}|\xi = L] + b(1-q)E[max\{0, w_k - z_k\}|\xi = S] + hqE[max\{0, z_k - w_k\}|\xi = L] + h(1-q)E[max\{0, z_k - w_k\}|\xi = S] + E[J_{k+1}(z_k - w_k)]$

$G_k(z_k)$ is a convex function for all k, and there will be a solution S_k which minimises $G(z)$ such that the optimal ordering policy becomes:
$u_k^* = \mu_k^* = S_k - x_k$ if $S_k > x_k$, and 0 otherwise.

Q2. Inventory Pooling

Primitives
D = demands
Q = quantity ordered
P = price
h = inventory costs = c-s
b = backholding costs = p-c

First we will show that $Q_p^* = \sqrt{n}Q^* + \mu(n - \sqrt{n})$

$$\begin{aligned} G(Q) &= hE[(Q-D)]^{+} bE[(D-Q)]^{+} \\ G'(Q^*) &= hP(D \leqslant Q^*) - b(1 - P[(D \leqslant Q^*)]) = 0 \\ P(D \leqslant Q^*) &= \frac{b}{h+b} \end{aligned}$$

Now, for the pooling we would obtain the same:

$$P(\sum_{L=1}^{n} D_i \leqslant Q_p^*) = \frac{b}{b+R}$$

Since this $\sum_{i=1}^{n} D_i = \sqrt{n}D_i + \mu(n - \sqrt{n})$,

$$P(\sqrt{n}D_1 + \mu(n-\sqrt{n}) \leqslant Q_p^*) = \frac{b}{b+R} \Longleftrightarrow P(D_1 \leqslant \frac{1}{\sqrt{n}}(Q_p^* - \mu(n - \sqrt{n}))) = \frac{b}{b+h}$$

which implies that $\frac{1}{\sqrt{n}}(Q_p^* - \mu(n-\sqrt{n}))) = Q^*$. Thus, $Q_P^* = \sqrt{n}Q^* + \mu(n - \sqrt{n})$. Next, we will apply the hint to prove the desired result:

$$\begin{aligned} nG(Q^*) &= n[hE[(Q^* - D)^{+}] + bE[(D - Q^*)^{+}]] \\ G(Q_p^*) &= [hE[(Q_p^* - \sum_{i=1}^{n} D_i)^{+}] + bE[(\sum_{i=1}^{n} D_i - Q_p^*)^{+}]] \end{aligned}$$

Since $\sum_{i=1}^{n} D_i = \sqrt{n}D_i + \mu(n - \sqrt{n})$ and $Q_P^* = \sqrt{n}Q^* + \mu(n - \sqrt{n})$, we can show that:

$$\begin{aligned} G(Q_P^*) &= [hE[(\sqrt{n}Q^* + \mu(n-\sqrt{n}) - \sqrt{n}D_i - \mu(n-\sqrt{n}))^{+}] \\ &\quad + bE[\sqrt{n}D_i + \mu(n-\sqrt{n}) - \sqrt{n}Q^* + \mu(n-\sqrt{n})^{+}]] \\ &= \sqrt{n}hE[(Q^* - D_i)^{+}] + \sqrt{n}\; bE[(D_i - Q^*)^{+}] = \frac{nG(Q^*)}{\sqrt{n}} \end{aligned}$$

Q.E.D.

Q3. Asset Selling with Maintenance Cost

Primitives:

w_k: Offer received in the period k.

x_k: Highest offer received up to period k.

$$u_k = \begin{cases} 1 & \text{if accept the offer } x_k \\ 0 & \text{otherwise} \end{cases}$$

c: Maintenance cost.

Dynamics:

$$x_{k+1} = \begin{cases} \max\{w_k, x_k\} & \text{if } u_k = 0 \\ T & \text{if } x_k = T \text{ or } u_k = 1 \end{cases}$$

Reward:

$$G_N(x_N) = \begin{cases} 0 & \text{if } x_N = T \\ x_N & \text{otherwise} \end{cases}$$

$$G_k(x_k) = \begin{cases} -c & \text{if } u_k = 0 \text{ and } x_k \neq T \\ x_k & \text{if } u_k = 1 \\ 0 & \text{if } x_k = T \end{cases}$$

DP-Algorithm:

$$J_N(x_N) = \begin{cases} 0 & \text{if } x_N = T \\ x_N & \text{otherwise} \end{cases}$$

$$\begin{aligned} J_k(x_k) &= \begin{cases} \max\limits_{u_k \in \mu_k} \{x_k, \mathbb{E}_{w_k}[J_{k+1}(x_{k+1})] - c\} \quad \text{if } x_k \neq T \\ 0 \text{ otherwise} \end{cases} \\ &= \begin{cases} \max\limits_{u_k \in \mu_k} \{x_k, \mathbb{E}_{w_k}[J_{k+1}(\max\{w_k, x_k\})] - c\} \quad \text{if } x_k \neq T \\ 0 \text{ otherwise} \end{cases} \end{aligned}$$

Where $\mathbb{E}_{w_k}[J_{k+1}(\max\{w_k, x_k\})] = \sum\limits_{j=1}^{n} p_j \max(x, w_j)$

Thus the optimal policy is given by

$$\mu^*(x) = \begin{cases} 1 & \text{if } x \geqslant \sum\limits_{j=1}^{n} p_j \max(x, w_j) - c \\ 0 & \text{otherwise} \end{cases}$$

This optimal policy implies that the actual reward should be greater or equal to the expected reward if we don't accept the maximum offer up to k. Thus we can define the optimal policy as follows:

$$\mu^*(x) = \begin{cases} 1 & \text{if } c \geqslant \sum\limits_{j=1}^{n} p_j \max(x, w_j) - x \\ 0 & \text{otherwise} \end{cases}$$

where $h(x) = \sum\limits_{j=1}^{n} p_j \max(x, w_j) - x$ is a nonincreasing function. Therefore, we obtain that if the cost is greater than the difference between the expected value of future offers and the maximum offer we have received, we decide to sell. Thus, we will have less incentives to sell as time goes by. Each period it is more probable that our maintenance cost will will be equal to our reward improvement.

Q4 An Investment Problem

Primitives:
x_k: wealth at the beginning of period k.
u_k: amount invested.
w_k: outcome (return).
$w_k = \begin{cases} 2u_k & \text{w.p. } p \\ 0 & \text{w.p. } 1-p \end{cases}$
where $\frac{1}{2} < p < 1$.

Constrains:
$u_k \geqslant 0$,
$u_k \leqslant x_k$.

Dynamics:
$x_{k+1} = x_k - u_k + w_k$,

Cost:
$g_N(x_N) = x_N$
$g_k(x_k, u_k, w_k) = 0$.

DP algorithm:
To simplify th $J_N(x_N) = \log(x_N)$.
$J_k(x_k) = \max_{u_k \in \mathcal{U}_k} \mathbb{E}_{w_k} \left[J_{k+1}(x_k - u_k + w_k)\right]$.
We will first prove that $J_k(x_k) = A_k + \log(x_k)$ by induction (where log is the logarithm in base e).

$$J_{N-1}(x_{N-1}) = \max_{u_{n-1}} \mathbb{E}_{w_{N-1}} \left[\log(x_{N-1} - u_{N-1} + w_{N-1})\right] = \tag{1}$$
$$= \max_{u_{N-1}} \mathbb{E}_{w_{N-1}}[\log(x_{N-1} - u_{N-1} + w_{N-1})] = \tag{2}$$
$$= \max_{u_{N-1}} \left[p \log(x_{N-1} + u_{N-1}) + (1-p)\log(x_{N-1} - u_{N-1})\right]. \tag{3}$$

Now, we derive and apply FOC to find the maximum,

$$\frac{p}{x_{N-1} + u_{N-1}} - \frac{1-p}{x_{N-1} - u_{N-1}} = 0 \tag{4}$$

$$p x_{N-1} - p u^*_{N-1} = x_{N-1} + u^*_{N-1} - p x_{N-1} - p u^*_{N-1}, \tag{5}$$

solving we obtain $u^*_{N-1} = (2p-1)x_{N-1}$, which makes sense, because $2p-1$ is between 0 and 1 for $\frac{1}{2} < p < 1$. Plugging into the expression of $J_{N-1}(x_{N-1})$

$$J_{N-1}(x_{N-1}) = p\log(x_{N-1} + (2p-1)x_{N-1}) + (1-p)\log(x_{N-1} - (2p-1)x_{N-1}) = \tag{6}$$
$$= p\log(2p x_{N-1}) + (1-p)\log(2(1-p)x_{N-1}) = \tag{7}$$
$$= p\log(2p) + p\log(x_{N-1}) + (1-p)\log(2(1-p)) + (1-p)\log(x_{N-1} = \tag{8}$$
$$= p\log(2p) + (1-p)\log(2(1-p)) + \log(x_{N-1}). \tag{9}$$

So, for $N-1$ we have $A_{N-1} = p\log(2p) + (1-p)\log(2(1-p))$. Now, suppose $J_{k+1}(x_{k1}) = A_{k+1} + \log(x_{k+1})$ let's prove it for k:

$$\begin{aligned} J_k(x_k) &= \max_{u_k} \mathbb{E}_{w_k}\left[A_{k+1} + \log(x_{k+1})\right] = && (10)\\ &= \max_{u_k} \mathbb{E}_{w_k}\left[A_{k+1} + \log(x_k - u_k + w_k)\right] = && (11)\\ &= \max_{u_k} \left[A_{k+1} + p\log(x_k + u_k) + (1-p)\log(x_k - u_k)\right]. && (12)\end{aligned}$$

Applying first order conditions we would obtain the same equation as before (A_{k+1} does not depend on u_k), so $u_k = (2p-1)x_k$. Plugging in the $J_k(x_k)$ we obtain the same way as before

$$J_k(x_k) = A_{k+1} + p\log(2p) + (1-p)\log(2(1-p)) + \log(x_k). \tag{13}$$

So, $J_k(x_k) = A_k + \log(x_k)$, where $A_k = A_{k+1} + p\log(2p) + (1-p)\log(2(1-p))$. So, the policy will be to invest $(2p-1)x_k$ of wealth each period k.

Q5 A Scheduling Problem

(a)

Let i and j be the kth and $(k+1)$st questions in an optimal ordered list:

$$L = (i_0, \cdots, i_{k-1}, i, j, i_{k+2}, \cdots, i_{N-1})$$

Also consider the list with i and j interchanged:

$$L' = (i_0, \cdots, i_{k-1}, j, i, i_{k+2}, \cdots, i_{N-1})$$

Then define R as the reward. Thus

$$\begin{aligned}\mathbb{E}(R(L)) =& \mathbb{E}\left[R(i_0, \cdots, i_{k-1})\right] + p_{i_0}\cdots p_{i_{k-1}}\left(p_iR_i - (1-p_i)F_i + p_ip_jR_j - p_i(1-p_j)F_j\right)\\ &+ p_{i_0}\cdots p_{i_{k-1}}p_ip_j\mathbb{E}\left[R(i_{k+2}, \cdots, i_{N-1})\right]\\ \mathbb{E}(R(L')) =& \mathbb{E}\left[R(i_0, \cdots, i_{k-1})\right] + p_{i_0}\cdots p_{i_{k-1}}\left(p_jR_j - (1-p_j)F_j + p_jp_iR_i - p_j(1-p_i)F_i\right)\\ &+ p_{i_0}\cdots p_{i_{k-1}}p_jp_i\mathbb{E}\left[R(i_{k+2}, \cdots, i_{N-1})\right]\end{aligned}$$

L optimal implies

$$\begin{aligned}\mathbb{E}(R(L)) &\geqslant \mathbb{E}(R(L'))\\ p_iR_i - (1-p_i)F_i + p_ip_jR_j - p_i(1-p_j)F_j &\geqslant p_jR_j - (1-p_j)F_j + p_jp_iR_i - p_j(1-p_i)F_i\\ p_iR_i - (1-p_i)F_i - p_jp_iR_i + p_j(1-p_i)F_i &\geqslant p_jR_j - (1-p_j)F_j - p_ip_jR_j + p_i(1-p_j)F_j\\ R_i(p_i - p_jp_i) - F_i(1 - p_i - p_j(1-p_i)) &\geqslant R_j(p_j - p_ip_j) - F_j(1 - p_j - p_i(1-p_j))\\ R_ip_i(1-p_j) - F_i(1-p_i)(1-p_j) &\geqslant R_jp_j(1-p_i) - F_j(1-p_j)(1-p_i)\\ (1-p_j)(R_ip_i - F_i(1-p_i)) &\geqslant (1-p_i)(R_jp_j - F_j(1-p_j))\\ \frac{R_ip_i - F_i(1-p_i)}{1-p_i} &\geqslant \frac{R_jp_j - F_j(1-p_j)}{1-p_j} \qquad \text{Q.E.D}\end{aligned}$$

(b)

Primitives:

x_k: Reward up to period k.

$$u_k = \begin{cases} 1 & \text{to answer} \\ 0 & \text{otherwise} \end{cases}$$

$$w_k = \begin{cases} Rk & \text{w.p } p_k \\ -F_k & \text{w.p } (1-p_k) \end{cases}$$

Dynamics:

$$x_{k+1} = \begin{cases} x_k + u_k w_k & \text{if } x_k \neq R \\ T & \text{if } x_k = T \text{ or } u_k = 0 \end{cases}$$

Reward:

$G_N(x_N) = 0$

$$G_k(x_k) = \begin{cases} w_k & \text{if } u_k = 1 \text{ and } x_k \neq T \\ 0 & \text{otherwise} \end{cases}$$

DP-Algorithm:

$$J_N(x_N) = 0$$

$$\begin{aligned} J_k(x_k) &= \max_{u_k} \mathbb{E}_{w_k}\left[g(x_k, u_k, w_k) + J_{k+1}(x_{k+1})\right] \\ &= \max\left\{w_k + J_{k+1}(x_k + w_k), 0\right\} \end{aligned}$$

So we have that

$$\begin{aligned} J_{N-1}(x_{N-1}) &= \max\left\{\mathbb{E}_{w_{N-1}}\left[w_{N-1}\right], 0\right\} \\ &= \max\left\{R_{N-1}p_{N-1} - F_{N-1}(1-p_{N-1}), 0\right\} \end{aligned}$$

Then win and loose do not depend on x_{N-1}. They will depend on the optimal list L. Thus

$$J_k(x_k) = \max\left\{R_k p_k - F_k(1-p_k) + J_{k+1}(L), 0\right\}$$

Therefore, $R_k p_k - F_k(1-p_k) + J_{k+1}(L)$ does not depend on x_k. It depend in fact on u_k which is the decision of continuing playing or not. So the optimal policy since the beginning is to look if $R_k p_k - F_k(1-p_k) + J_{k+1}(L) > 0$.