MATH3202 - Dynamic Programming

Maxwell Bo Chantel Morris

May 25, 2018

Sets

T	Set of days
L	Set of demand levels, { H, R }, where H is high and L is low

Data

d_{tl}	Demand of bottles on day $t \in T$ with $l \in L$ level of demand
m	Maximum number of bottles that can be ordered
b	Base delivery cost ($)
e	Bottle delivery cost ($)
r	Bottle retail price ($)
i	Number of bottles in the fridge on day $first(T)$
c	Number of bottles fridge can hold
h	Chance of having higher demand than usual, where "high demand" is H $\in L$
z	Discounted price ($)
p	Chance of a "high demand" day post discount, where "high demand" is H $\in L$

State

$S_{t\text{bottles}}$	Bottles in fridge at the start of day $t \in T$

Action

$a_{t\text{ordered}}$	Bottles to order in preparation for day $t \in T$
$a_{t\text{discount}}$	Decision to apply discount of retail price z on day $t \in T$, where 1 decides to apply the discount

Constraints

$$S_{t\text{bottles}} \in [0, c] \qquad \forall t \in T \tag{1}$$
$$a_{t\text{ordered}} \in [0, m] \qquad \forall t \in T \tag{2}$$
$$a_{t\text{discount}} \in \{0, 1\} \qquad \forall t \in T \tag{3}$$

- Constraint (1) ensures that the bottles stored must be between 0 and the number of bottles that the fridge can hold
- Constraint (2) ensures Jenny's Juices can only order up to the maximum that can be ordered. In addition no juice can be returned
- Constraint (3) ensures that the discount to retail cost can either apply or not apply.

Functions

$$\text{delivery\_cost}(a_t) = \begin{cases} b + a_{t\text{ordered}} \cdot e, & \text{if } a_{t\text{ordered}} > 0 \\ 0, & \text{otherwise.} \end{cases}$$

$$\text{decide\_price}(a_t) = \begin{cases} z, & \text{if } a_{t\text{discount}} = 1 \\ r, & \text{otherwise.} \end{cases}$$

$$\text{decide\_sold}(S_t, a_t, l) = \min(d_{tl}, S_{t\text{bottles}} + a_{t\text{ordered}})$$

$$\text{clamp}_l^h(x) = \max(l, \min(h, x))$$

$$S^M(S_t, a_t, l) = \text{clamp}_0^c(s_{t\text{bottles}} + a_{t\text{ordered}} - \text{decide\_sold}(S_t, a_t, l))$$

$$C_t(S_t, a_t, l) = (\text{decide\_price}(a_t) \times \text{decide\_sold}(S_t, a_t, l)) - \text{delivery\_cost}(a_t) + V_{t+1}(S^M(S_t, a_t, l))$$

- delivery_cost details the cost of delivery. If some bottles of juice are ordered, the delivery cost is equal to the wholesale price per bottle for each bottle plus the delivery fee. If no bottle of juice are ordered, there are no delivery fees.
- decide_price details the price that the juice is sold for. Bottles of juice are either sold at retail cost or discounted cost, if a discount is applied.
- decide_sold evaluates how much juice is sold on a given day. If the demand is less than the bottles stored added to the bottles order, then Jenny's juices will sell up to the demand of that day. Else, they will sell how much is stored added to how much is ordered for that day.
- The contribution function evaluates the total profit. It is calculated by multiplying the sell price of the juice by the amount of juice sold, plus the value function. The delivery cost of that order is then subtracted.
- The value function for Communication 9 maximises the contribution function in order to evaluate the optimal solution over the period of time.
- The value function for Communication 10 This function maximises the contribution function in order to evaluate the optimal solution over the period of time. Crucially, this function evaluates the contribution function where there is either High or Regular demand. The probability of each scenario is used, where Regular demand is the complementary probability of High demand.
- The value function for Communication 11 function maximises the contribution function in order to evaluate the optimal solution over the period of time. Crucially, this function evaluates the contribution function where there is either High or Regular demand. The probability of each scenario is used, where Regular demand is the complementary probability of High demand. Furthermore, this function optimises the solution in light of the action of whether or not to use a discount.

Value Functions

$$V_{succ(last(t))}(S_t) = 0$$

Communication 9

$$V_t(S_t) = \max_{a_t}\{C_t(S_t, a_t, R)\}$$

where

$$a_t \in \{\langle \text{ordered} \rangle \,|\, \text{ordered} \in [0, m]\}$$

Communication 10

$$\text{has\_higher\_demand}(a_t) = \begin{cases} p, & \text{if } a_{t\,\text{discount}} = 1 \\ h, & \text{otherwise.} \end{cases}$$

$$V_t(S_t) = \max_{a_t}\{((1 - \text{has\_higher\_demand}(a_t)) \times C_t(S_t, a_t, R)) + (\text{has\_higher\_demand}(a_t) \times C_t(S_t, a_t, H)\}$$

where

$$a_t \in \{\langle \text{ordered}, \text{discount} \rangle \,|\, \text{ordered} \in [0, m], \text{discount} \in \{0\}\}$$

Communication 11

$$V_t(S_t) = \max_{a_t}\{((1 - \text{has\_higher\_demand}(a_t)) \times C_t(S_t, a_t, R)) + (\text{has\_higher\_demand}(a_t) \times C_t(S_t, a_t, H)\}$$

where

$$a_t \in \{\langle \text{ordered}, \text{discount} \rangle \,|\, \text{ordered} \in [0, m], \text{discount} \in \{0, 1\}\}$$

Results

Communication 9 - Profit is $156.0

Day	Bottles	Order
1	0	15
2	8	0
3	0	15
4	4	15
5	8	0
6	4	11
7	10	0

Communication 10 - Profit is $180.47

To find the optimal action, index the row with the number of bottles you currently have, and the column with the day.

	1	2	3	4	5	6	7
0	15	15	15	15	14	15	15
1		15	15	15	13	14	15
2		15	15	15	12	13	15
3		15	15	15	11	12	15
4		14	15	15	10	11	14
5		13	15	15	9	0	13
6		12	15	15	0	0	12
7		11	14	14	0	0	11
8		0	13	13	0	0	10
9			12	12	0	0	9
10			11	11	0	0	0

Communication 11 - Profit is $189.91

In this instance, if you have 0 bottles on day 1, you should order 15 bottles, and apply a discount

	1	2	3	4	5	6	7
0	D 15	15	15	15	D 14	D 15	D 15
1		15	15	15	D 15	D 14	D 15
2		15	15	15	D 15	D 13	D 15
3		15	15	15	D 15	D 12	D 15
4		D 14	15	15	D 15	D 11	D 14
5		13	15	15	D 14	D 10	D 13
6		D 12	15	15	D 13	D 9	D 12
7		D 11	14	15	D 12	0	D 11
8		0	13	14	D 0	0	D 10
9			12	13	D 0	0	D 9
10			11	12	D 0	0	D 8