ORIE 5530: Modeling under Uncertainty Final Project

Queenie Liu
ql299
ql299@cornell.edu

December 2019

Abstract

The paper aims for developing different models to solve an inventory-type problem for a dealer shop. To narrow down the model, I choose a specific car model, Benz GLA 250 to construct my models. Finally, I get the most optimizing policy for my business.

1 Introduction

I am working on managing a dealer shop to sell brand-new cars. To address the optimizing strategy for the sales, I intend to solve an inventory-type problem of choosing a specific car model, Benz GLA250 to determine the best-selling price and appropriate quantity in stock. The value of the car will be depreciated with the increase of time it has been invented and will also be influenced by the demand of the market.

2 Initial Model

First, I chose to use discrete-time Markov Chain for analyze the supply chain of Benz GLA 250. To begin with, the initial inventory of the car in stock is X where t is the number of weeks, so X_t represents the number of GLA in stock at the end of week t. The price c is independent on each client. Suppose that each week will only have one client who has the intention to buy the car. The client eventually buys the GLA c with probability $\lambda(c)$ and the probability that they won't buy the GLA is $1-\lambda(c)$. Also, set a random variable B_i, i is the ith client. If the ith client buys the GLA when it is offered at price c, $B_i = 1$. Whenever $B_i = 1$ the inventory of GLA in stock is reduced by 1 and the reward achieved by the store will be increased by c.

Inventory Level	X	X-1	X-2	X-3	...	2	1	0
X	1 - $\lambda(c)$	$\lambda(c)$	0	0	...	0	0	0
X-1	0	1 - $\lambda(c)$	$\lambda(c)$	0	...	0	0	0
X-2	0	0	1 - $\lambda(c)$	$\lambda(c)$	...	0	0	0
X-3	0	0	0	1 - $\lambda(c)$	...	0	0	0
X-4	0	0	0	0	...	0	0	0
...	...	...	...	...	...	...	...	...
2	0	0	0	0	0	1 - $\lambda(c)$	$\lambda(c)$	0
1	0	0	0	0	0	0	1 - $\lambda(c)$	$\lambda(c)$
0	0	0	0	0	0	0	0	1

3 Discrete-time Markov Chain Model

To achieve the initial best solution, I start with initializing an appropriate range of the inventory X and the price c. Suppose that the initial possible range of the inventory X is between 0 to 20, and the possible range of the price c is between 30 thousand and 50 thousand. To make the model more straightforward, I assume that when $c = 50$, the probability of the client will buy the GLA is $\lambda(c) = 0$ and when $c = 30$ the probability of the client will buy the GLA is $\lambda(c) = 1$.

Then I built a discrete-time Markov Chain of this problem and solved the DTMC to get the long-run average reward $E_x[\sum_{i=1}^{t} \beta c B_i(c)]$. Assume the discount factor is 0.98 and my initial cost per car is 15k. Through this model, I can solve for the most optimizing values of X and c in order to achieve the most reward. Sensitivity analysis is performed with respect to changes in the values of X and c within a local region of their most optimizing values to gain insights into how the rewards change over this region.

$$\begin{bmatrix} V(20) \\ V(10) \\ V(18) \\ V(17) \\ \vdots \\ V(2) \\ V(1) \\ V(0) \end{bmatrix} = \begin{bmatrix} R(20) \\ R(19) \\ R(18) \\ R(17) \\ \vdots \\ R(2) \\ R(1) \\ R(0) \end{bmatrix} + \begin{bmatrix} 1-\lambda(c) & \lambda(c) & 0 & ... & 0 & 0 \\ 0 & 1-\lambda(c) & \lambda(c) & ... & 0 & 0 \\ 0 & 0 & 1-\lambda(c) & ... & 0 & 0 \\ 0 & 0 & 0 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & \lambda(c) & 0 \\ 0 & 0 & 0 & ... & 1-\lambda(c) & \lambda(c) \\ 0 & 0 & 0 & ... & 0 & 1 \end{bmatrix} \beta \begin{bmatrix} V(20) \\ V(10) \\ V(18) \\ V(17) \\ \vdots \\ V(2) \\ V(1) \\ V(0) \end{bmatrix}$$

Which is the same as:

$$
\begin{bmatrix} V(20) \\ V(10) \\ V(18) \\ V(17) \\ \vdots \\ V(2) \\ V(1) \\ V(0) \end{bmatrix} = (I_{21} \text{ - } \beta \begin{bmatrix} 1-\lambda(c) & \lambda(c) & 0 & ... & 0 & 0 \\ 0 & 1-\lambda(c) & \lambda(c) & ... & 0 & 0 \\ 0 & 0 & 1-\lambda(c) & ... & 0 & 0 \\ 0 & 0 & 0 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & \lambda(c) & 0 \\ 0 & 0 & 0 & ... & 1-\lambda(c) & \lambda(c) \\ 0 & 0 & 0 & ... & 0 & 1 \end{bmatrix})^{-1} \begin{bmatrix} R(20) \\ R(19) \\ R(18) \\ R(17) \\ \vdots \\ R(2) \\ R(1) \\ R(0) \end{bmatrix}
$$

Through coding my DTMC model in python, the best decision for me is [42,7] and the result is 37, which means that the most optimizing value is $X = 7$ and $c = 42$. Thus, the best policy for me is to set the price at 42k and the amount of inventory is 7. The long-run average reward is 37.07297843480234 K.

```
(x[0]+30,N-1-y[0])
```

```
(42, 7)
```

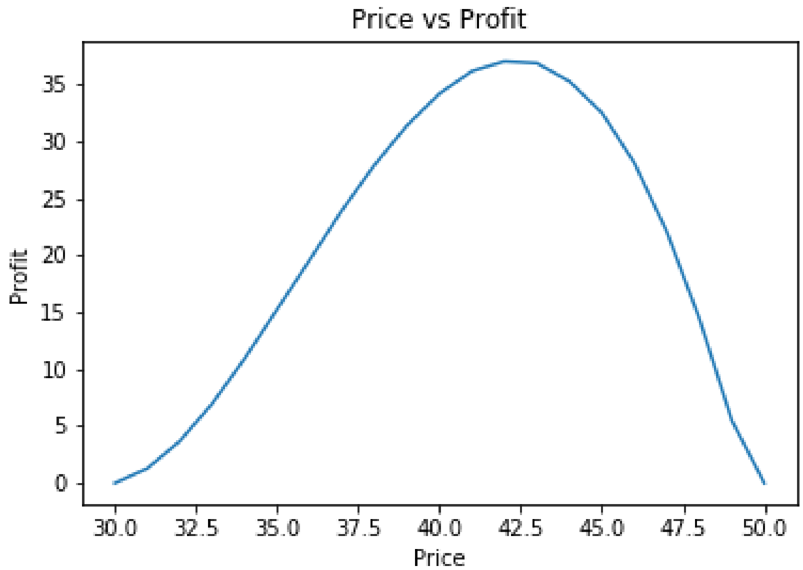
```
np.amax(np.array(result))
```

```
37.07297843480234
```

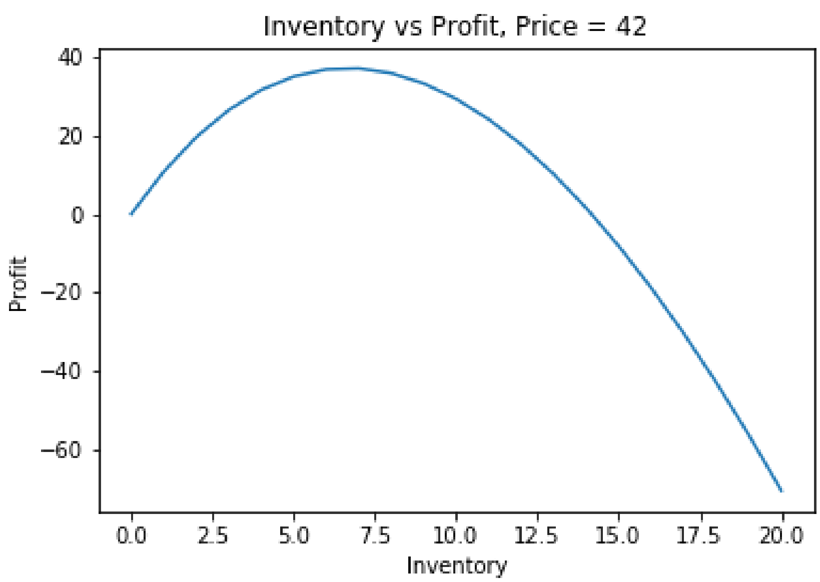
```
result
```

```
[array([-101.41195763,  -91.84893636,  -82.70299629,  -73.98264927,
         -65.69658089,  -57.85365397,  -50.46291221,  -43.53358389,
         -37.0750856 ,  -31.09702612,  -25.60921033,  -20.6216432 ,
         -16.14453387,  -12.18829987,   -8.7635713 ,   -5.8811952 ,
          -3.55224   ,   -1.788     ,   -0.6       ,    0.        ,
           0.        ]),
 array([-9.00838798e+01, -8.07623734e+01, -7.18850882e+01, -6.34615670e+01,
        -5.55015577e+01, -4.80150176e+01, -4.10121179e+01, -3.45032482e+01,
        -2.84990215e+01, -2.30102787e+01, -1.80480935e+01, -1.36237776e+01,
        -9.74888557e+00, -6.43522039e+00, -3.69483844e+00, -1.54005516e+00,
         1.65494515e-02,  9.62125165e-01,  1.28354568e+00,  9.67402734e-01,
         0.00000000e+00]),
 array([-79.609489  , -70.46231188, -61.78798789, -53.59723931,
        -45.90103159, -38.71057879, -32.03734928, -25.89307149,
        -20.28973978, -15.2396205 , -10.75525815,  -6.84948169,
         -3.53541098,  -0.82646338,   1.26363949,   2.72086487,
          3.53086181,   3.67895391,   3.15013201,   1.92904656,
          0.        ]),
 array([-70.1141975 , -61.06531869, -52.519468  , -44.48872293,
```

The following plot show the relationship between price and profit.



The following plot show the relationship between inventory and profit.



4 Simulation

As times goes on, the value of the car is actually have chance to decrease during our time horizon. The probability of selling the product at price c will decrease one percent as day passed. To simulate the market more accurately, the popularity of the product more go up and down each day. From above, I obtained the most optimizing value of X and c, which is $X = 7$ and $c = 42$. Then I used simulation to solve the above initial model. The probability that each client buys the GLA in each week will follows a uniform distribution over the time horizons from $\max\{\lambda(c) - \epsilon, 0\}$ to $\max\{\lambda(c) + \epsilon, 0\}$, for some relatively small value of $\epsilon > 0$. This simulation solution provides the long-run average reward for a given pair of parameters X and c sensitivity analysis is performed with respect to changes in the values of X, c and ϵ within a local region of their most optimizing values to gain insights into how the rewards change over this region. Using simulation to search around the optimal pair(7, 42), I got that the optimal pair is (6, 43): 6 inventory and 43 price with profit 39.3298120391 K. After the simulation, I found that the simulation results are a little bit greater than the result I got without simulation but they are very similar.

5 Dynamic Programming Model

For the dynamic programming model, I constructed an infinite time intervals discounted dynamic programming model of the GLA inventory problem with the goal of determining the optimal initial inventory X and the optimal prices c_i for every period $i = 1, ..., t$. Through the infinite horizon discounted dynamic programming model, I solved the corresponding dynamic optimization problem to obtain the optimal initial inventory X and the optimal prices c_i for every period $i = 1, ..., t$.

The initial inventory is X units of cars. One client arrives per period. The client eventually buys the GLA at price c with probability $\lambda(c)$ and the probability that they won't buy the GLA is $1 - \lambda(c)$. If I only had one car to sell I would choose the price that maximizes the expected immediate reward = price * probability of sale = $c * \lambda(c)$. If the arriving client buys a unit of the product at the offered price then the inventory goes down by 1. Also, set a random variable B_i, i is the ith client. If the ith client buys the GLA when it is offered at price c, $B_i = 1$. That is, $\mathcal{P}B_i(c) = 1 = \lambda(c)$. Assume that clients are statistically identical, which means that they all have the same probability of buying under a price c. Then, I seek to choose a price strategy to maximize

$$V(x) = E_x[\sum_{i=0}^{\infty} \beta^i c_i B_i(c_i)]$$

.

Since I am trying to constructing an infinite horizon setting, I do not have a dependence of V on the period i. This means I lose the terminal condition we had which said that $V_{n+1}(X) = 0$.

$$V(X) = \max_{c \geq 0}\{c\lambda(c) + (1 - \lambda(c))V(X) + \lambda(c)(V(X - 1))\}$$

Using $\lambda(c) = e^{-c}$ instead of $\lambda = 1 - c$, then

$$V(X) = \max_{c \geq 0}\{ce^{-c} + \beta(1 - e^{-c})V(X) + \beta e^{-c}(V(X-1))\}$$

$$= \beta V(X) + \max_{c \geq 0}\{ce^{-c} + \beta e^{-c}(V(X-1) - V(X))\}$$

Then I take a derivative and compare to 0. I get that

$$c^*(X) = 1 - \beta(V(X-1) - V(X)) = 1 + \beta(V(X) - V(X-1))$$

This optimal price can be achieved once I know the value.

$$V(X) = \beta V(X) + e^{-c^*(X)}$$

. The final result is obtained through value iteration. Maximum reward converges at 32.08928371093901 K which is at comparable level as the long-run average reward. One of the reason that the result of dynamic programming is that the sample amount in this model is still not enough to represent an average optimal reward.