ORIE 5530: Modeling Under Uncertainty
Final Project

Instructor: Professor Mark S. Squillante

Jingxuan SUN (js3422)

2019.12.7

1. Problem Context

I would like to model the process of arrangement of films at a movie theater as I am a cinema lover. Although it is not very intuitive, I fit the scenario into an inventory type of problem, in the sense that how we can optimize the movie theater's profit by choosing an optimal policy on number of screens to show at each period of time to address the uncertainty of demand.

2. Model Setup

To start with, I developed a discrete-time Markov chain for the "screens". That is, I treated one screening of certain film as one unit of goods. The time horizon is divided by days. There are 10 theaters at the cinema, so the maximum number of screenings each day would be 10, and there are in total 11 states in the DTMC representing the number of screenings left not shown (out of 10) at the end of each day.

The transition of state depends on the demand of customers. For default setup, the number of customers for certain film each day follows a Poisson distribution with rate 100. There appears to be two different "unit of analysis", the number of screenings and the number of customers. To bring these two concepts together, I setup each screening or theater to have a capacity of 20 seats.

The optimization process aims to find an optimal policy for each film in order to maximize the infinite horizon profit. The policies are closely related to the chains' transition probabilities. Although it might be a bit artificial, I define the following set of policies: increase X screenings for tomorrow when the number of screenings I showed today didn't fulfil the demand; decrease X screenings for tomorrow when the demand today didn't fill up the number of screenings I showed today. Note that the X in each policy is fixed, and for this set of 11 policies, X ranges from 0 to 10 and X values keep number of screenings non-negative). For example, if my current policy is to increase/decrease $X = 2$ screenings, today I showed 5 screenings but there were only 21 people came to see the film, tomorrow I will show 3 screenings, although according to the demand today ideally I should show 2 screenings tomorrow. This explains how I define the policies (the counterpart of these policies is when I try to vary the X, this will be described in detail in dynamic programming section). To put it simply, I try to pick the optimal X to maximize the infinite horizon reward for different movies.

The reward function is essentially the profit per day. Revenue depends on number of tickets sold (ie. the demand), while the variable cost is calculated on a per screening basis. The cost component also includes a per day fix cost. In this way, I manage to create a convex reward function and make it possible and meaningful to “maximize”. As infinite horizon discounted reward is also related to the initial state, I also want to find the optimal number of screenings for the initial day. This makes sense in the way that box office success is highly correlated with the first day/weekend performance.

3. Mathematical Representations

The components mentioned above finish the setup of the discrete-time Markov chain. To put is mathematically, there will be 11 chains representing 11 policies for each film. For the policy $X = 2$, the chain can be represented as the following diagram:

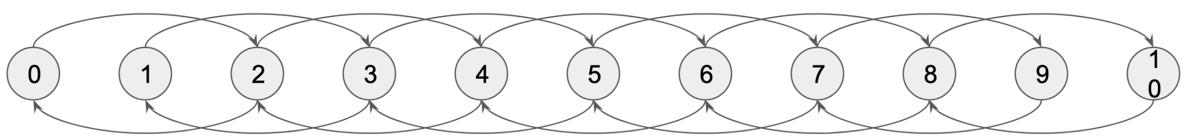

Figure 1: Diagram for chain with X = 2

The transition probability matrix is calculated based on demand distribution. Still using the example mentioned above, when the policy $X = 2$, $P_{53} = P(D \leq 80)$ (because if there were 81 people I will be showing 5 screenings as the capacity is 20), $P_{57} = P(D > 100) = 1 – P(D \leq 100)$, and $P_{55} = P(80 < D \leq 100)$ where $D \sim Poisson(100)$. Therefore, in each row of the transition probability matrix $\mathbb{P}$, there are 3 entries summing up to 1, which preserves the characteristics of the matrix.

I used `numpy.poisson.cdf` to generate these matrices. Full matrix for $X = 2$ is as follows:

$$\begin{bmatrix}
3.72e-44 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\
3.72e-44 & 1.90e-22 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\
1.90e-22 & 0 & 7.51e-12 & 0 & 0 & 0 & 0 & 0.99 & 0 & 0 & 0 \\
7.51e-12 & 0 & 0 & 1.08e-05 & 0 & 0 & 0 & 0 & 0.99 & 0 & 0 \\
1.08e-05 & 0 & 0 & 0 & 0.02 & 0 & 0 & 0 & 0 & 0.98 & 0 \\
0.02 & 0 & 0 & 0 & 0 & 0.50 & 0 & 0 & 0 & 0 & 0.47 \\
0 & 0.53 & 0 & 0 & 0 & 0 & 0.45 & 0 & 0 & 0 & 0.02 \\
0 & 0 & 0.98 & 0 & 0 & 0 & 0 & 0.02 & 0 & 0 & 6.4e-05 \\
0 & 0 & 0 & 0.99 & 0 & 0 & 0 & 0 & 6.40e-05 & 0 & 1.26e-08 \\
0 & 0 & 0 & 0 & 0.99 & 0 & 0 & 0 & 0 & 1.26e-08 & 2.25e-13 \\
0 & 0 & 0 & 0 & 0 & 0.99 & 0 & 0 & 0 & 0 & 2.25e-13
\end{bmatrix}$$

The reward function for number of showing i is:

$$R = \sum_{d=max(0,Cap*(i-1)+1)}^{Cap*i+1} t * d * \sqrt{P(d)} - C_s * (10 - i) - C_f$$

where $Cap = 20$, t = per ticket revenue, d = demand, C_s = per screening cost, C_f = per day fixed cost. It is a bit artificial to calculate the revenue by summing up all possibilities, but the aim is to manipulate the reward function to be "maximizable".

I used the following formula to calculate the infinite horizon discounted reward:

$$V = (I - \beta P)^{-1} R$$

where $\beta = 0.98$.

I did the same calculations for all 11 policies, the best result is as follows. For this particular film with $Demand \sim Poisson(100)$, the best policy is to increase/decrease 1 screenings according to the daily demand, and the best infinite horizon discounted reward 351609.74 (dollar) is achieved at showing 5 screenings on the first day. This result makes intuitive sense, as the average demand per day would be 100 customers which is approximately equivalent to 5 screenings, and increasing/decreasing 1 screenings is what people would normally imagine to do.

I first used simulation to verify the processed described above. For each policy and each starting number of screening, I ran 10 simulations and took the max and average "infinite" horizon discounted reward over 1000 days. The result complies with the previous calculations.

4. Dynamic Programming

In contrast with the fix number of increasing/decreasing of screenings, in the dynamic programming model, the degree of adjustment can be varied and would fit more closely to the change of demand. For example, if I showed 5 screenings for today, but only 21 people came, I'd show 2 screenings for tomorrow; if 110 people came, I'd show 6 screenings for tomorrow. The number of decreased screenings equals to 4 in the first case while the number of increased screenings equals to 1 in the second case. The Bellman equation is as follows:

$$V(b) = R * (10 - b) + \beta * \sum_{k=1-b}^{10-b+1} P_{11-(b+k)} * V(b+k)$$

where b is the "budget", that is the number of screenings "left unshown" for the day; R is the reward function used before; P is an array of 11 probabilities, each representing $P(Cap * (i-1) < D \leq Cap * i)$; k is the possible number of increasing screenings (negative values represent decreasing screenings).

The final result is obtained through value iteration. Maximum reward converges at 324861.32 which is at comparable level as the optimal policy reward. The potential reason that the dynamic programming result is smaller than that of the optimal policy is the number of sample path generated in the policy simulations is still small to represent an average optimal reward.

5. Simulation on Seasonality

To be closer to reality, different movies might have different viewing and box office dynamics, depending on their genres, releasing year, run time, etc. There are also general weekly patterns, for example, more people tend to go to cinema on weekend than weekdays. I generated 4 movies that display very different traits and did the afore mentioned simulation process. There's really nothing mind-blowing shown in the simulation without seasonality, as the policy of increasing/decreasing 1 screening is the optimal policy typically. However, after doing this step, there are more interesting results.

The 4 film entities are as follows. There are different numbers of high and low seasons for each film. Periods are in number of days. X-men has a weekly cycle, IT2 has a yearly cycle. Green Book is a generally popular Oscar nominated film, and Bohemian Rhapsody is niche for music fans. All of the four display very different patterns.

Name	Genre	Demand(Low)	Period(Low)	Demand(High)	Period(High)
X-men	Superhero	100	Weekday (5)	500	Weekend (2)
IT2	Horror	100	The rest	300	June-August (90), Halloween (30)
Green Book	Oscar, Drama	NA	NA	200	All time
Bohemian Rhapsody	Music, Biopic	50	All time	NA	NA

Table 1: Demand and Seasons for 4 film entities

Figure 2 shows that it is probably the best to increase/decrease 5 screenings for films display similar demand and seasonal trends as X-men; similarly, 1 for IT2, 6 for Green Book, and 2 for Bohemian Rhapsody. Some implication of the project is that for the cinema, it is really essential to allocate the showings differently depending on the characteristics of each film, thus it is important to run simulations, perhaps taking more relevant features into consideration, on a case by case basis.

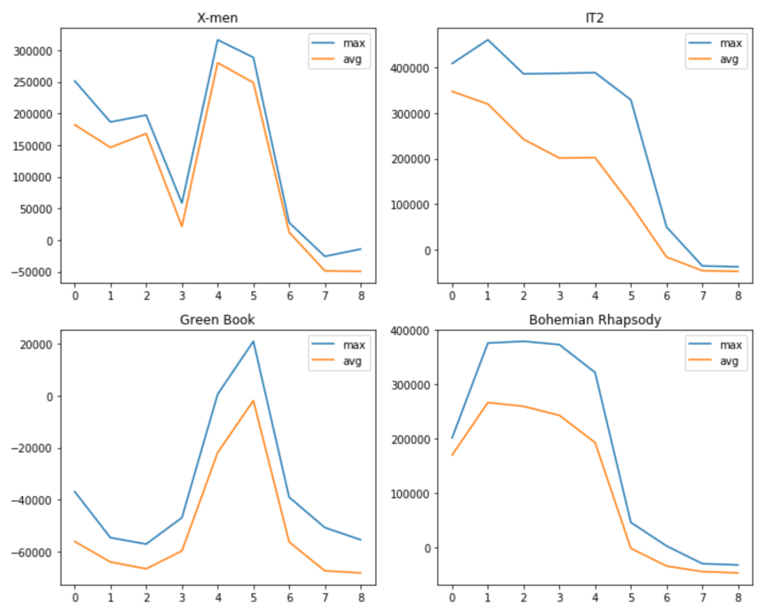

Figure 2: Max and avg reward at different policies