McGill University — Desautels Faculty of Management

MGSC 670 - Revenue Management

Group Assignment 1: Markdown Strategy

261083570 Konstantin Volodin

261078897 Ying-Fang Liang

260658030 Emery Dittmer

261077392 Julie Chanzy

Instructor: Professor Maxime Cohen

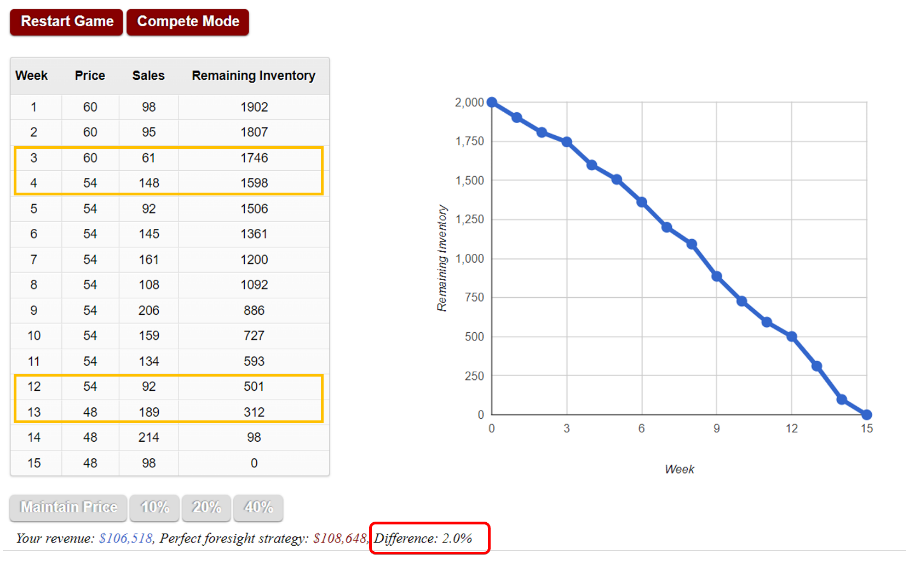
May 2023

Contents

1 Introduction

Any modern consumer recognizes words like “discount”, “clearance”, or “liquidation.” Behind the scenes of these words is a carefully orchestrated pricing and markdown strategy for each product from retailers. The relationship between price and demand is well documented and pricted by the field of economics. With perfect information and competition, as the price increases, the demand decreases, and vice-versa. Simplified price is, therefore, a control mechanism for the quantity of an item sold.

Our report will examine and propose a markdown strategy for a hypothetical retail fashion company that sells goods to customers (B2C). Within this framework, the organization holds 2,000 units of inventory with a starting price of $60. The inventory must be sold within 15 weeks. We have three discount options, hence a price that varies among: full price, -10% -20% and -40% off. Each discount is irreversible. Each pricing strategy adopted will generate a unique demand curve. This inventory-to-price scenario is illustrated in USC Marshall’s School of Business retail game [Randhawa, 2014]. The web application simulates market behaviour based on a user-generated markdown strategy.



Week	Price	Sales	Remaining Inventory
1	60	98	1902
2	60	95	1807
3	60	61	1746
4	54	148	1598
5	54	92	1506
6	54	145	1361
7	54	161	1200
8	54	108	1092
9	54	206	886
10	54	159	727
11	54	134	593
12	54	92	501
13	48	189	312
14	48	214	98
15	48	98	0

Your revenue: $106,518, Perfect foresight strategy: $108,648, Difference: 2.0%

Figure 1: Example of Markdown Strategy in the Retailer Game.[Randhawa, 2014]

As a retailer, your objective is to maximize revenue, being as close as possible to the *perfect foresight strategy* (i.e. with a *difference* of 0%). The optimized strategy or *perfect foresight strategy* maximizes the revenue generated over the 15 weeks by implementing the right markdown strategy at the right time. Since the salvage value is $0, the maximum revenue is mathematically equivalent to a linear sum of units sold at sale price for each sale price. The goal is to develop a data science approach to consistently minimize the difference between the expected revenue and perfect foresight strategy. We will use a sample of 15 "historical" sales, independent runs of the retail model and the web app to validate our strategies.

2 Background

The retail industry is increasingly focused on price markdowns to optimize profit, despite the challenges such as time constraints and sometimes the lack of historical data. Many revenue management models have tried to optimize the timing, duration and price discounts to influence consumer behaviour. Intuitively price drives demand. Markdowns of price increase reach and conversions to sales. Therefore, the optimal strategy balances maximizing revenue while minimizing the inventory left at the end of the sales. Any unsold inventory will likely contribute to environmental waste, incur disposal cost or increase inventory storage costs. In the last decades, innovations and new data management tools have allowed companies to move away from a manual pricing system and better manage their revenues.

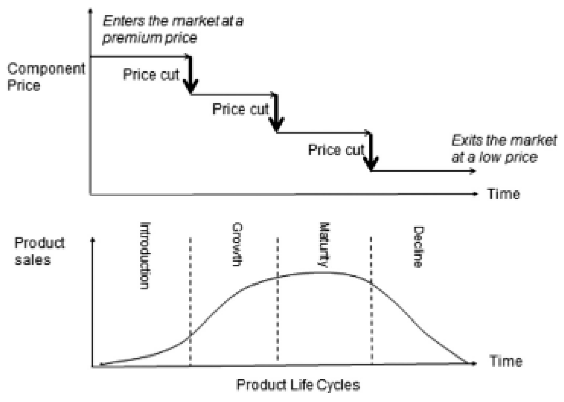


Figure 2: Demand & Price through Product Life Cycles.[Chung et al., 2015]

Airline carriers started managing discounted fares back in the 1970s according to [Talluri and Van Ryzin, 2004]. The key pain point was optimizing sales through relevant discounted fares management. Statistical forecasting techniques and mathematical optimization methods have allowed airlines to frame their strategies, before extending other services such as hospitals, car rentals, cruise lines, railways, energy and broadcasting. In their paper, Talluri and van Ryzin highlight the importance and difficulty of modelling consumer behavior in predicting revenues. Some research discussed the implementation of dynamic pricing models and brought the problem back to a set of binary variables based on whether the customer will buy the article [Bitran and Pedrosa, 1998, Feng and Gallego, 2000, Gallego and Van Ryzin, 1994]. Talluri and van Ryzin focused on a choice-based approach that simulates consumer behaviour with a choice model by including probabilities of purchasing depending on the company's offer panel, even without historical data.

On their side, fast-fashion retailers are looking at clearance price strategies before liquidation [Caro and Gallien, 2012]. In short, in retail, products are clustered by price categories; and each price category receives a clearance price. Therefore, the clearance price affects all the products from that price category. The approach then relies on dynamic programming. In 2008, the fast fashion multinational company Zara, opted to experiment with markdown strategies on multiple items. The model produced a 6% or $90M increase in sales revenue throughout the store and other cultural and business model improvements [Caro and Gallien, 2012]. The approach allows for a scalable, simple and reliable method to set pricing.

3 Proposed Strategies

The literature and background on this web app pricing markdown structure suggest that revenue management is more complex than a one-size-fits-all approach. Several strategies or pricing policies may be appropriate for benchmarking and discovering an optimal or compromising strategy. We will investigate several markdown or price change methods to understand their impact and potential application. We will evaluate the results using a simulation and web app validation. The simulation of the price and demand is done via a randomized sales distribution generated from mean price, standard deviation, and random number generation. Meanwhile the validation comes from a crosswalk code that interacts with the model and web app.

3.1 Random Policy

We will first investigate a random choice policy. This policy represents a likely worst-case scenario, where a company has no expertise and not logic to operate with. The policy will arbitrarily choose the pricing policy at each time t for future time periods.

Based on the random nature of the model and rules of the model, we expect a smaller distribution then the baseline mode (presented next). Since we cannot raise the price once lowered the cumulative probability of a 40% discount approaches 40% by the end of week 2 and 50% by the end of the 15 weeks, which is the highest of any choice. Empirically we have observed that if the 40% discount is offered early all units are sold within the 15-week period. The distribution of revenue should largely follow the distribution of the 40% off price distribution we have observed.

3.2 Baseline Model - Price Consistency

The second strategy we will investigate a naïve approach to markdown strategies: do not change the price. We consider this a baseline policy as this is the easiest to follow and requires no intervention on the price. While this policy ensures each item is sold at the highest price, it will likely leave unsold products. This naïve model is useful as a comparison point for the average revenue and standard deviation. In other words, the naïve "do nothing" model is a baseline to compare against and build off. The optimal price over the 15 weeks may be the original set price of $60 or maybe some combination of price reductions, this model helps to determine the best approach in most situations.

We expect a highly varied revenue with a high standard deviation from this model that could be more consistent and optimal than the *perfect foresight strategy*.

3.3 Naïve Adaptive Likelihood Policy

The next policy we will investigate is a naïve adaptive likelihood method. This policy uses the previously observed data to predict the likelihood that given the current price the inventory will sell out. If at time t the inventory is not likely to sell out, the model investigates the remaining price choices and selects the first discount that gives a high probability of inventory sell out. For instance, at week 5 the model determines that the sales are not likely to lead to sell out, the 10% , 20% and 40% are all likely to lead to inventory sell out, so the model adjusts the % off to 10% to stimulate sales.

We expect this model to have a high consistency of difference between revenue and the *perfect foresight strategy*, with higher revenue than the baseline model and potential room for improvement.

3.4 Adaptive Likelihood Price Policy

The fourth policy we will investigate is an adaptive likelihood price method. This method is similar to the naïve adaptive likelihood policy with a critical difference. Instead of using the estimated unit price increase this model aggregates the revenue from 10 separate forecasts for the remaining weeks. The 10 forecasts are run for each price point remaining and are based on randomly selected observed data from pervious runs. The model should help capture the original model's randomness through more robust forecasts.

We expect this model to produce a more consistent difference between revenue and the *perfect foresight strategy* with a higher average revenue than the naïve adaptive likelihood policy.

3.5 Likelihood Demand Trend Projection

The likelihood demand trend projection policy we will investigate is an adaptive likelihood method that uses multiple runs of 15-week data to assume the same trends hold for future data. The given data suggests that each price reduction has a compounding effect on the unit volume sold. For instance, 10% off increases unit sales by 30%, and 20% increases unit sales by 75% compared to previous periods. The model uses rules based on the sales increase per price category to estimate the unit sales for the remaining periods. To accomplish this the price distribution for the current model is added to the observed price distributions and the model evaluates the estimated likelihood of “out of stock condition”. In other words, if the estimated units sold by the end of the 15 weeks is less than the remaining units, the model increases the discount to stimulate the forecasted demand.

We expect this model to produce a more consistent difference between revenue and the *perfect foresight strategy* with a higher average revenue than the naïve adaptive likelihood policy.

3.6 Short-term Moving Average Naïve

Next, we will investigate a naive moving average approach. In this approach the previous 3-day demand is accumulated into a moving average to forecast demand for time t+1. The 3-day average helps smooth any randomness in the demand curve due to noise. The function considers the momentum of previous 3 week of sales to determine if the current trajectory of sales will result in no inventory at the end of the 15-week period. If the sales momentum from the previous 3 days is lower than the average number of units needed to sell out by the end of the period, the price is dropped. Let’s consider an example: at week 4, the previous 3 weeks sales average is 100 units per week. There are 1,700 units left. The average sales needed to sell out of inventory is 141 units (1,700 units / 12 weeks). Since the moving average sales (100 per week) is below the required 141 sales per week the model will lower the price. The model will then wait 3 weeks to allow the new policy to take shape before re-examining a price decrease. The same factors will help decide whether to lower the price once again or not. This simple model accounts for the randomness of demand and helps to ensure more unit sales by valuing unit sales momentum over optimal price strategy.

We expect this model to have a higher consistency than the baseline mode, lower the price quickly to the lowest amount and consistently sell out of all units. This model is likely better than the baseline but could be better as the price has a maximum decrease speed.

3.7 Long-term Adaptive Moving Average with Wait

The final policy we will investigate is an adaptive moving average with a policy. This policy is similar to the naïve moving average policy with added levels of complexity. The moving average is used to predict the total sales over the remaining time and compares it to the remaining amount. If the predicted sales are less than the remaining total remaining amount at time t then the price will decrease in an effort to increase unit sales. This method unlike the previous one helps smooth out the randomness in the data more so than the previous short term moving average policy.

We expect this model to have a lower difference consistency than the short-term naïve average, since it is likely that more of the units sold will be at different price points more quickly whereas the average model will have a slower price evolution. However, we also expect this model to have the highest average revenue of all models.

3.8 Reinforcement Learning

A proposed final future state policy is the reinforcement learning strategy. Artificial intelligence models rely on supervised reinforcement learning which learn as they go. The models are rewarded for certain actions and attempt to maximize the reward, relying on previous runs or iterations to build off of. These models are ideal for repeated tasks that have defined conditions. For example, Open Ai in 2019 demonstrated a reinforcement learning model that simulated a game of hide and seek [Baker et al., 2019]. We will apply reinforcement learning to this problem, where the reward function is the maximization of the revenue, and the inputs are the app itself. While this model offers little explanation ability since rl models are black box models, the reinforcement learning policy may perform better than previous policies.

We expect that the reinforcement learning model will have the highest revenue and the lowest revenue distribution of all models previously tested.

4 Implementation

4.1 Data Treatment

While we have access to 15 runs of "historical" data, before using this data, we must apply data preprocessing.

First, we must ensure that the data does not contain errors or null values. We, therefore, removed null rows and appended the run number (1-15) to each observation within the data. After the data cleanup, we are left with a dataset containing the run number, the week number, the price of the item ($60, $54, $48, $36), the unit demand and the revenue generated from that week of sales.

Second, we need to transform the data into more meaningful formats. We grouped the data by price to measure unit sales' standard deviation and mean. This first grouping was helpful in the naïve price policy we applied. We also grouped the data by price to measure the mean and standard deviation of the unit sales compared to the previous week. In other words, what is the % increase (or decrease) in unit sales when a new price point is set. It may be that a 20% decrease in price increases unit sales by 70%, independent of the previous price point.

Based on some casual observations of the data, the distributions appear roughly normally distributed, we will use this as a basis for simulation and modelling.

4.2 Simulation

To forecast the impact of the pricing strategies we are investigating, we must test the policies against data. Since we cannot access the exact equation or similar random number generator used for the web app, we must first utilize simulations. The model built takes the policy, the average and the standard deviation of the observed price points and generates 100 simulations. For each simulation and each time t, the model generates unit sales based on a numpy random normal method using the mean and standard deviation of the current price point. The simulation approach best reflects a real-life scenario as a forecast function is generally unknown, resulting in a reliance on past data to predict future demand. The generated numbers reflect the data distribution found within the historical data.

- Pseudo Code of Simulation Approach and Validation

```
procedure SIMULATE(means per price, stdev per price)
    for week in length(means per price) do
        generate unit sales with mean, stdev
    end for
    return unit sales
end procedure

procedure TEST POLICY(policy, list of unit sales)
    for t = 1 to 15 do
        generate price at week t+1 from policy logic
    end for
    return price at week t+1
end procedure

for policy in policy candidate list do
    for t = 1 to 100 do
        list of unit sales = Simulate (price mean, price stdev)
        policy results = Test Policy (policy, list of unit sales)
    end for
end for
```

4.3 Validation

The simulation approach helps validate the models and quickly select the best policy with limited information. However, for this context, we need not rely solely on the simulation data, we can instead use the actual data from the web app. The assumptions and approach we have previously laid out may have unknown challenges or other pitfalls. In this scenario we are not limited to the simulation and approximation of "real world data". We will therefore validate the simulation results using the web application. To implement the validation, we will utilize a selenium-based python script which goes between the code and user interface on the web app. Each policy is tested in real-time against the app while recording the results to evaluate after the fact.

- Pseudo Code of Policy Validation using WebApp

```
procedure POLICY VALIDATE(policy)
    URL = "http://www.randhawa.us/games/retailer/nyu.html"
    expected return = with URL use policy for t+1 decision
    return expected return
end procedure

for policy in policy candidate list do
    for t = 1 to 100 do
        policy results = Policy Validate (policy)
    end for
end for
plot policy results
```

5 Discussion and Recommendations

5.1 Policy Comparison

In this report, we proposed eight various pricing strategies including heuristic, probability-based, traditional moving average, and reinforcement learning approaches. All except the reinforcement learning approach were simulated and validated on the web app. Based on the simulation results in Figure 3, Naïve Adaptive Likelihood Policy provides the highest average revenue followed by likelihood shared.

5.2 Recommended Policy

6 References

[Baker et al., 2019] Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., and Mordatch, I. (2019). Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*.

[Bitran and Pedrosa, 1998] Bitran, G. and Pedrosa, L. (1998). A structured product development perspective for service operations. *European Management Journal*, 16(2):169–189.

[Caro and Gallien, 2012] Caro, F. and Gallien, J. (2012). Clearance pricing optimization for a fast-fashion retailer. *Operations research*, 60(6):1404–1422.

[Chung et al., 2015] Chung, W., Talluri, S., and Narasimhan, R. (2015). Optimal pricing and inventory strategies with multiple price markdowns over time. *European Journal of Operational Research*, 243(1):130–141.

[Feng and Gallego, 2000] Feng, Y. and Gallego, G. (2000). Perishable asset revenue management with markovian time dependent demand intensities. *Management science*, 46(7):941–956.

[Gallego and Van Ryzin, 1994] Gallego, G. and Van Ryzin, G. (1994). Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management science*, 40(8):999–1020.

[Randhawa, 2014] Randhawa, R. S. (2014). Retailer game.

[Talluri and Van Ryzin, 2004] Talluri, K. and Van Ryzin, G. (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33.