A Study on Machine Learning Algorithms for Dynamic Pricing

Project synopsis submitted in partial fulfillment of the requirements for the degree of

Master of Business Administration

By

HARIDEEPAK MARTHATI

20810029

MBA Batch 2020-2022

Under the guidance of

Dr Manu K. Gupta

Assistant Professor

DoMS, IIT Roorkee

Department of Management Studies

Indian Institute of Technology

30th April 2022

Acknowledgement

The completion of this project, which is a study on the topic of Machine Learning Algorithms for Dynamic Pricing, would not have been possible without the expertise and support of **Dr Manu K. Gupta**, my project guide. I would also like to extend my gratitude to the entire IT faculty at IIT Roorkee, for equipping me with the know-how to tackle the project with confidence.

HARIDEEPAK MARTHATI
DoMS IIT Roorkee

Table of Contents

Abstract

Many modern e-commerce businesses have a need to set item prices appropriately on a daily basis, in order to optimize their inventory management process and improve profitability. Large e-commerce retailers often carry over four hundred million items in their stock; and they have a need to understand the right price-point to set for a particular item, to maximize the revenue generated by it on a daily basis. It is easy in this connected world to study online customer behavior and identify as to which items are consistently purchased together by regular shoppers. Items which are purchased together can be classified as a 'Basket'. Hence a 'Basket' can consist of {Coffee, Bread, Butter, Jam and Toothpaste} – Each item in the basket has a specific price and an associated purchase quantity. It is clear that the basket needs to be priced appropriately to make it attractive for consumers – pricing it too high could cause it to remain unsold, while pricing it too low could reduce the revenue derived from it. This project showcases as to how bandit algorithms (leveraging Thompson Sampling), can be used to develop robust & effective dynamic pricing systems.

Keywords: Dynamic Pricing, Bandit Algorithms, Thompson Sampling, Pricing & Revenue Management

1. Introduction

Dynamic Pricing in an e-commerce environment, can be challenging due to the scale of operations and unpredictable consumer behavior. Machine learning can be effectively used to develop dynamic pricing algorithms. Thompson sampling can be used to develop an active dynamic pricing algorithm and we explore in detail bandit algorithm (Thompson Sampling) for dynamic pricing.

In a normal approach to dynamic pricing, the assumption is that it is of some parametric form. The issue with a simple demand modelling function is that it assumes that demand is stationary – this is rarely the case. The general demand function is of the form:

$$Demand_{i,t}(p_{i,t}) = Baseline\ Demand_{i,t} * (\frac{p_{i,t}}{p_{i,t-1}})^{price\ elasticity(i)}$$

Here $Baseline\ Demand_{i,t}$ is the demand forecast for an *item i* on *day t*, if the price of *'item i'* on *'day t-1'* is $p_{i,t-1}$, the price of *'item i'* on *'day t'* is $p_{i,t}$ & the price elasticity of *'item i'* is *price elasticity(i)*. Both the *Baseline Demand* and *price elasticity of all items* are determined on a daily basis. Observations about the demand equation shown above:

- Price Elasticity of any 'item i' in a basket, is always less than or equal to -1.
- Price on any particular day for a normal good, is slightly greater than (or) in some cases equal to the price on the previous day.
- Given that the price elasticity of any item is <= -1 and that price of an item on a particular day is greater than the price set for the item on the previous day; the second part of the RHS side of the equation becomes less than one; and hence baseline demand calculated on any particular day effectively becomes the upper limit for the demand on the same day – In the exceptional case when $p_{i,t}$ *is 'equal to'* $p_{i,t-1}$, the demand on a particular day becomes equal to the baseline forecast on that particular day (this exceptional case happens in the case of sparse data when multiple repetitions of the same price point can occur for an item on multiple consecutive days).

1.1 Major Objectives

The major objective of the project is to draw out a comparison between Passive and Active dynamic pricing algorithms. Major deliverables in the project are:

- Build a passive Max-Rev Optimization algorithm which uses historical forecasting techniques like ARIMA to forecast future elasticities and prices.
- Build an active Max-Rev Optimization algorithm which uses Thomson Sampling to update the elasticity vector and solve for the revenue maximizing price across the shopping basket.
- Perform graphical comparison of the two algorithms, to assess their performance vis-à-vis one-another.

2. Literature Review

The problem of solving Dynamic Pricing with an unknown demand model, was first explored in the paper titled “Optimal Learning by Experimentation (June 1991)” by Philippe Aghion, Christopher Harris, Patrick Bolton and Bruno Jullien. In this paper the dynamic pricing problem was converted into a decision-making problem with discounted rewards – Bayesian dynamic programming was used to solve the dynamic pricing problem and arrive at the optimal pricing scheme. While approaches like these may on some occasions lead to obtaining the optimal pricing scheme, measuring their effectiveness in practical scenarios can become intractable.

The problem of solving Dynamic Pricing (DP) under uncertain/unknown demand constraint was first tackled in the paper titled "A partially observed Markov decision process for dynamic pricing." by Aviv, Yossi, and Amit Pazgal. They assumed that the demand function is derived from a family of parametric models, in which the values of the parameters are unknown. A prior distribution is applied on the unknown parameters and posteriors are obtained through the application of Bayes theorem. The objective was discounted revenue maximization with an infinite horizon.

The first attempt at creating a non-myopic policy for dynamic pricing, when the actual demand function is unknown was explored in the paper titled “Dynamic Optimization and Learning: How Should a manager set Prices when the Demand Function is Unknown? (January 2005)” by Alexandre Xavier Carvalho and Martin Puterman. They proposed a variation of the one-step lookahead policy which maximizes revenue for the next two steps, instead of the next step – this one-step lookahead policy variant was used in conjunction with a binomial demand distribution (which had a logit expectation). This semi-myopic one-step lookahead policy was compared with other myopic policies and proved the superiority of one-step lookahead policies in solving dynamic pricing optimization problems.

An innovative exploration-exploitation approach was first crafted by Besbes, Omar, and Assaf Zeevi in their paper "Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms." Many price-points were experimented with; and the demand function was estimated at these price points, in the exploration phase. In the subsequent exploitation phase an optimal price point is obtained by leveraging a ‘revenue maximization approach’.

A study on various parametric models for the demand function was conducted in the paper titled “Dynamic Pricing with an Unknown Demand Model: Asymptotically Optimal Semi-Myopic Policies (January 2014)” which was authored by Bora Keskin and Assaf Zeevi. They used maximum likelihood-based formulations to derive exploration-exploitation based policies. The paper titled “Online Network Revenue Management using Thompson Sampling (2016)”, authored by Kris Johnson Ferreira, David Simchi-Levi and He Wang – provided a Thompson-Sampling based solution to maximize revenue for multiple items in a finite horizon, given an unknown demand function and inventory constraints. The paper titled “Thompson Sampling for Dynamic Pricing (February 2018)”, authored by Ravi Ganti, Matyas Sustik, Quoc Tran & Brian Seaman from Walmart Labs – provided a showcase as to how to apply active learning algorithms to develop an efficient dynamic pricing model for a real-life e-commerce dataset.

3. Model Description

For designing a dynamic pricing (DP) system, we need to identify the below listed three components:

i. Modelling the demand component based on historical price and demand data.
ii. Based on the modelling in step one, we derive a demand function – This demand function is leveraged to calculate optimal prices on a daily basis – We can extend the forecasting to any number of time periods.
iii. Optimization Component which optimizes prices based on a pre-decided metric like revenue.

3.1 Demand Modelling Component

It is a known fact that an optimal price point exists for every item in an online marketplace. Increasing the price above the optimal price point often reduces the demand for the product, below the desired value; while reducing the price below the optimal price point may increase demand slightly, but adversely affects the overall profitability of the item. Hence the demand function belongs to some parametric family; and the underlying parameters can be estimated using historical data and statistical techniques. To create real world pricing systems, we need dynamic demand functions rather than static/stationary demand functions. One commonly used model to capture the demand of a particular 'item i' based on a certain baseline forecast on a reference day is listed as: $\boxed{demand_i(price_i) = (Baseline\ forecast_i) * ((\frac{price_i}{price_{0,i}})^{\gamma_{*,i}})}$

In the above function we know that for a $price_{0,i}$ on a given reference day, the demand is $Baseline\ forecast_i$. We compute the price elasticity of an item using historical data available for a particular item – then we maximize the $price_i$ for which $demand_i$ is maximized. However, it can be clearly seen that this model lacks the flexibility to dynamically model demand – Hence the above equation can be altered slightly to create a dynamic parametric model which leverages price elasticity to the optimal price to be set for a particular item on a given day.

$$\boxed{demand_{i,t}\left(price_{i,t}\right) = \left(Baseline\ forecast_{i,t}\right) * ((\frac{price_{i,t}}{price_{i,t-1}})^{\gamma_{*,i}})}$$

This new demand function is dynamic as the price set on any particular day is designed to maximize demand, given that the values of $\boxed{Baseline\ forecast_{i,t}}$ and the price elasticity of 'item i' ($\boxed{\gamma_{*,i}}$) can be computed using historical price and demand data for 'item i'.

3.2 Baseline Forecast Component

Online e-commerce stores and traditional brick-and-mortar e-commerce stores have access to huge volumes of past sales information which includes demand information for a variety of items at varying price points. Hence for a particular ‘item i’ we have access to a multitude of {price, demand} tuples:

$$(\{price_1, demand_1\}, \{price_2, demand_2\}, \ldots \{price_{t-1}, demand_{t-1}\})$$

Using the above historical demand information, the $Baseline\ forecast_{i,t}$ can be computed using some suitable parametric equation. Similarly, a forecast for $price_{i,t}$ can be generated using the historical price information – this price forecast however needs to be checked and corrected in necessary to ensure that it maximizes $demand_{i,t}(price_{i,t})$ or $Revenue_{i,t}(price_{i,t})$.

Ideally the ‘price set for an item i’ should maximize the revenue generated by the ‘item i’ at any time-period ‘t’.

One method for forecasting $Baseline\ forecast_{i,t}$ using historical demand data is -- by giving recent historical demand data more importance than older historical demand data – this can be modelled using the below equation:

$$Baseline\ Forecast_{i,t} = c_0 + \sum_{\tau=t-1}^{0} \beta^{t-\tau} * demand_{i,\tau} + \varepsilon_t$$

In the equation c_0 can be a small constant in the range {0.1, 0.2}, β can be a value which gives lesser weight to a historical demand point, based on how farther away it is from the current time period – It can be a value in the interval {0,1} like 0.5. ε_t is the probability density function value of a standard normal distribution at any specific point on the curve and acts as an *error term associated with a baseline forecast at time ‘t’*.

3.3 Price Elasticity

Each ‘item i’ in a ‘basket B’ has an associated price elasticity. To determine the price elasticity of an ‘item i’ we need to create a linear model featuring the price, demand and baseline forecast information of ‘item i’. The equation

$$demand_{i,t}(price_{i,t}) = (Baseline\ forecast_{i,t}) * ((\frac{price_{i,t}}{price_{i,t-1}})^{\gamma_{*,i}})$$

can be approximated to

$$demand_{i,t}(price_{i,t}) = (Baseline\ forecast_{i,t}) + ((price_{i,t} - price_{i,t-1}) * (Baseline\ forecast_{i,t} * \gamma_{*,i}/price_{i,t-1}))$$

The above equation shows a regression relationship between historical demand and historical elasticity data. Either the Ordinary Least Squares (OLS) or Robust Least Squares (RLS) approach can be used to estimate elasticity of a particular item. However, one has to note that price of an 'item i' at 'time t' needs to be different from the price of an 'item i' at 'time t-1' for the regression relationship to hold good. In case historical information of price elasticities exists at distinct price points, then price elasticity estimation becomes easier. However, in reality for many e-commerce products, the number of distinct price points across a series of time-periods could be as low as five or even less in some extreme cases – Hence estimating price elasticity using historical data, is a bigger challenge compared to demand estimation using historical data – as the problem of data sparsity needs to be tackled in an effective manner.

3.4 Optimization Component

The most complicated aspect of the demand modelling system is the optimization component and this varies based on the underlying algorithm. A Passive/static algorithm would have a simpler optimization component compared to an active algorithm. The optimization component of the demand modelling system is its most critical part and one which often singlehandedly determines the extent to which the demand modelling component can set the right prices for multiple items in a basket, on a particular day, by predicting the revenue (or) demand maximizing price set for the basket.

Let $\boxed{price_{Basket} = [price_1 \, , \, price_2 \, , \ldots ., \, price_{|B|}]}$ be the optimal price basket.
The revenue maximizing equation (MAX-REV optimization problem) -- which we are supposed to solve is listed below (Equation 1):

$$\boxed{\begin{gathered} price_{t,Basket} = \left(\arg max_{price_{Basket}} \sum_{i \epsilon Basket} \frac{(price_{i,t}^2 * \; Baseline \; forecast_{i,t} \; * \; \gamma_{*,i})}{price_{i,t-1}} \right) - \\ (price_{i,t} * \; Baseline \; forecast_{i,t} \; * \; \gamma_{*,i}) + (price_{i,t} * \; Baseline \; forecast_{i,t}) \end{gathered}}$$

3.4.1 Passive Algorithm

Let us first look at the passive algorithm for demand modelling which uses ARIMA model to forecast prices and price elasticities based on historical data. The algorithm for the passive dynamic pricing engine is as listed below:

Algorithm 1: Passive Dynamic Pricing Engine (MAX_REV_PASSIVE)
Input: A basket B, and time period T over which we intend to maximize cumulative revenue

- For $\boxed{t \; \leftarrow 1 \; to \; T}$ do:
  - For each item $\boxed{i \; \epsilon \; Basket}$ calculate their demand forecasts using the demand forecaster.
  - For each item $\boxed{i \; \epsilon \; Basket}$ calculate their price elasticities $\boxed{\gamma_{*,i}}$ using ARIMA.
  - Solve the MAX-REV optimization problem, shown in Equation (1) to obtain the revenue $\boxed{Revenue_t}$.
- End

3.4.2 Active Algorithm

For going ahead with the Active Algorithm, we need to introduce a Gaussian Prior on the elasticity vector as shown below (Equation 2):

$$\pi_0(\gamma_*) = N(\mu_0, \in_0)$$

Here μ_0 is the mean vector of elasticities of the base dataset, using which we project revenues into the future; and $\in_0$ is the covariance matrix which can be defined as $\in_0 = cI$ where 'c' is an appropriate constant in the range {0,1} and 'I' is the identity matrix. 'c' should be set adequately to perform exploitation. For demonstration purposes the value of 'c' is set to 0.15.

We now define a vector θ_t as a vector comprising of all $\theta_{i,t}$ equal to (Equation 3):

$$\theta_{t,i} = \left(\frac{price_{i,t}^2 * \ Baseline\ forecast_{i,t}}{price_{i,t-1}}\right) - \left(price_{i,t} * Baseline\ forecast_{i,t}\right)$$

We define another matrix (Equation 4) as:

$$M_t^{-1} = \frac{\theta_t * \ \theta_t^T}{\sigma^2} + \lambda I$$

In Equation 4, λ is a small positive constant. For demonstration purposes the value of λ is set to 0.18.

We constantly need to update the mean vector and covariance matrix of the gaussian prior vector after each prediction as below (Equation 5 and Equation 6):

$$\mu_t = ((\in_{t-1}^{-1} + \ M_t^{-1})^{-1})\left((\in_{t-1}^{-1} * \ \mu_{t-1}) + \left(\left(\frac{Revenue_t - Revenue_t^{mean\ of\ basket}}{\sigma^2}\right) * (\theta_t)\right)\right)$$

$$\in_t = (\in_{t-1}^{-1} + \ M_t^{-1})^{-1}$$

After updating mean vector and covariance vector in the above fashion, we can recompute the new gaussian prior as listed below (Equation 7):

$$\pi_t(\gamma_*) = N(\mu_t, \in_t)$$

In every iteration we sample the price elasticities from the gaussian prior shown in Equation 7 – We only accept those price elasticities which are lesser than -1.

With all the notations in place now let us look at the algorithm for the active dynamic pricing engine:

Algorithm 2: Active Dynamic Pricing Engine (MAX_REV_ACTIVE)
Input: A basket B, and time period T over which we intend to maximize cumulative revenue

- Choose an initial price elasticity vector γ_* randomly where all values of γ_* are lesser than -1.
- Initialize the gaussian prior on the above elasticity vector $\pi_0(\gamma_*)$ as described in Equation 2, above.
- For $t \leftarrow 1\ to\ T$ do:

- Keep sampling from $\boxed{\gamma_t \rightarrow \pi_{t-1}}$ till all components of $\boxed{\gamma_t}$ are lesser than -1.
- For each item $\boxed{i \epsilon Basket}$ calculate their demand forecasts using the demand forecaster.
- Solve the MAX-REV optimization problem, shown in Equation (1) -- with $\boxed{Baseline\ forecast_{i,t}}$, $\boxed{price_{i,t-1}}$ and $\boxed{\gamma_t}$, obtained from solving the MAX-REV optimization equation, to obtain the revenue $\boxed{Revenue_t}$.

- End

3.4.3 Synthetic Data Generation Algorithm

We can run the **MAX_REV_PASSIVE** and **MAX_REV_ACTIVE** algorithms on both synthetically generated and real-life datasets. To generate synthetic data another algorithm is listed below:

Algorithm 3: Synthetic Data Generation algorithm
Input: A basket B, containing B = 'n' items, a small constant $\boxed{c_0}$ > 0, time period T and an algorithm A (such as MAX-REV-ACTIVE or MAX-REV-PASSIVE)

- Choose an initial price elasticity vector $\boxed{\gamma_*}$ randomly where all values of $\boxed{\gamma_*}$ are lesser than -1 but greater than -3.
- Set β = 0.5, $\boxed{price_{i,0}}$ appropriately based on the kind of shopping basket you want to create – set all values of $\boxed{demand_{i,0}}$ randomly based on some limits and also set $\boxed{Baseline\ forecast_{i,0}}$ equal to $\boxed{demand_{i,0}}$. *(The logic behind this is Baseline forecast can only be derived when prior demand exists and there is no prior demand before period 0)*
- Initialize ***Baseline forecast array*** with $\boxed{Baseline\ forecast_{i,0}}$, ***demand array*** with $\boxed{demand_{i,0}}$ and ***price array*** with $\boxed{price_{i,0}}$.
- For $\boxed{t \leftarrow 1\ to\ T}$ do:
  - Calculate the Baseline forecast for each item 'i' using the formula (Equation 8):
    $$\boxed{Baseline\ forecast_{i,t} = (c_0) + \left(\sum_{\tau=t-1}^{0}(\beta^{t-\tau} * demand_{i,\tau})\right) + (error_t)}$$
  - Use $\boxed{Baseline\ forecast_{i,t}}$ and $\boxed{\gamma_{*,i}}$ & $\boxed{price_{i,t}}$ generated by algorithm A to generate the demand vector for time period 't' using the formula (Equation 9):
    $$\boxed{demand_{i,t} = Max((Baseline\ forecast_{i,t} * \left((\frac{price_{i,t}}{price_{i,t-1}})^{\gamma_{*,i}}\right),\ 0)}$$
    (In some exceptional case where the first part of the RHS term in equation 9, become less than zero – the formula takes care of the situation)
  - Add $\boxed{Baseline\ forecast_{i,t}}$ to the ***Baseline forecast array***.
  - Add $\boxed{demand_{i,t}}$ to the ***demand array***.
  - Add $\boxed{price_{i,t}}$ to the ***price array***.
  - Observe the revenue $\boxed{Revenue_t}$ generated by algorithm A and store it for comparison.
- End

4. Results

Let us look at some results derived for some use-case datasets, which were used to compare the performance of the 'Passive Max-Rev' and 'Active Max-Rev TS' algorithms, which were described in the previous section.

4.1 Experiment 1

Initial Dataset was generated using the Synthetic Data Generation Algorithm described in the previous section.

- **Number of items in a Basket:** 11
- **Composition of price of items in basket:** All items are in price range (10-20 price units)
- **Number of periods for which base dataset was created:** 10 days
- **Number of periods for which data was forecasted from the base dataset:** 70 days
- **Number of iterations:** 10

Initial Price Dataset (Base)

price vector

15.11	15.19	15.26	15.32	15.37	15.42	15.44	15.52	15.56	15.57
19.49	19.55	19.59	19.66	19.7	19.75	19.8	19.89	19.92	19.93
14.49	14.54	14.56	14.6	14.62	14.72	14.74	14.77	14.86	14.95
10.33	10.41	10.48	10.52	10.53	10.57	10.6	10.7	10.75	10.78
12.68	12.73	12.75	12.79	12.85	12.91	12.99	13.01	13.02	13.12
14.19	14.25	14.32	14.41	14.42	14.46	14.51	14.54	14.62	14.67
14.24	14.26	14.32	14.38	14.4	14.48	14.57	14.62	14.71	14.74
17.34	17.37	17.4	17.44	17.53	17.55	17.58	17.62	17.65	17.73
15.5	15.58	15.67	15.76	15.81	15.9	15.99	16.01	16.09	16.18
13.99	14.04	14.09	14.15	14.22	14.26	14.28	14.37	14.4	14.42
13.97	14.02	14.05	14.07	14.15	14.23	14.24	14.27	14.29	14.39

Figure 1

Initial Demand Dataset (Base)

demand vector

13	7	7	7	7	7	7	7	7	7
12	6	6	6	6	6	6	6	6	6
12	6	6	6	6	6	6	6	6	6
8	4	4	4	4	4	4	4	4	4
5	3	3	3	3	3	3	3	3	3
10	5	5	5	5	5	5	5	5	5
14	7	7	7	7	7	7	7	7	7
9	5	5	5	5	5	5	5	5	5
15	8	8	8	8	8	8	8	8	8
9	5	5	5	5	5	5	5	5	5
8	4	4	4	4	4	4	4	4	4

Figure 2

Initial Forecast Dataset (Base)

forecast vector

13	6.78	7.03	7.16	7.22	7.25	7.26	7.27	7.28	7.28
12	6.28	6.28	6.28	6.28	6.28	6.28	6.28	6.28	6.28
12	6.28	6.28	6.28	6.28	6.28	6.28	6.28	6.28	6.28
8	4.28	4.28	4.28	4.28	4.28	4.28	4.28	4.28	4.28
5	2.78	3.03	3.16	3.22	3.25	3.26	3.27	3.28	3.28
10	5.28	5.28	5.28	5.28	5.28	5.28	5.28	5.28	5.28
14	7.28	7.28	7.28	7.28	7.28	7.28	7.28	7.28	7.28
9	4.78	5.03	5.16	5.22	5.25	5.26	5.27	5.28	5.28
15	7.78	8.03	8.16	8.22	8.25	8.26	8.27	8.28	8.28
9	4.78	5.03	5.16	5.22	5.25	5.26	5.27	5.28	5.28
8	4.28	4.28	4.28	4.28	4.28	4.28	4.28	4.28	4.28

Figure 3

Active Algorithm vs Passive Algorithm graph

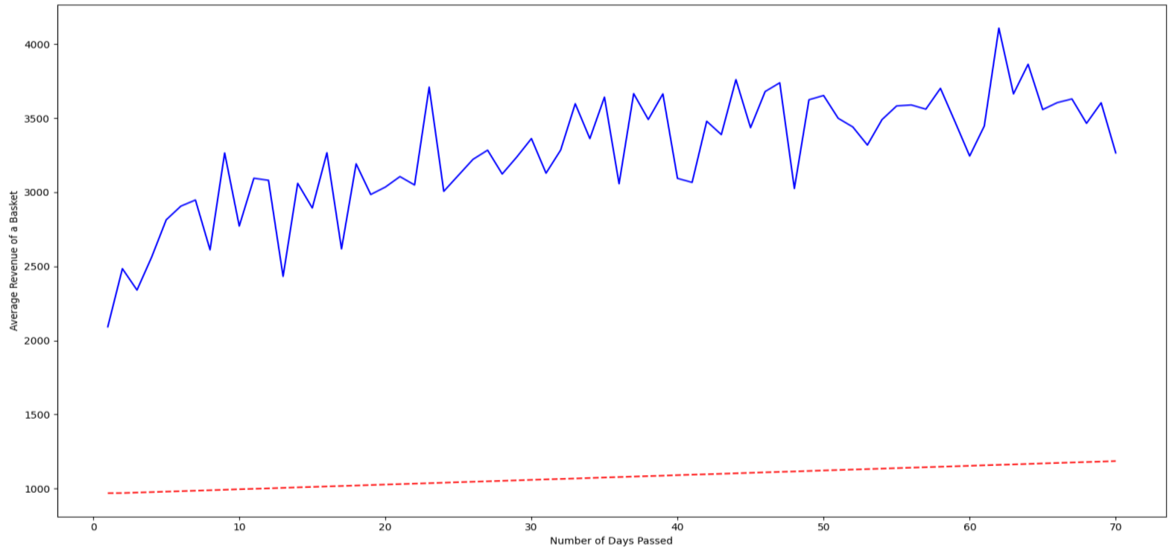

Figure 4 - dotted line (Passive) and continuous line (Active)

In this case the active algorithm outperforms the passive algorithm, even though the price points of all items in the basket are uniformly distributed in the range {10 to 20}.

4.2 Experiment 2

Initial Dataset was generated using the Synthetic Data Generation Algorithm described in the previous section and was different from the one used in 'Experiment 2'.

- **Number of items in a Basket:** 12
- **Composition of price of items in basket:** First item is in price range (5 – 15 price units) | Items two to six are in price range (16 – 26 price units) | Items five to eight are in price range (30-45 price units) | Items nine to 12 are in price range (48-70 price units)
- **Number of periods for which base dataset was created:** 10 days
- **Number of periods for which data was forecasted from the base dataset:** 70 days
- **Number of iterations:** 20

Initial Price Dataset (Base)

price vector

14.33	14.41	14.42	14.49	14.59	14.62	14.7	14.74	14.83	14.89
14.94	14.98	15.07	15.1	15.13	15.23	15.33	15.38	15.42	15.5
18.39	18.45	18.51	18.53	18.57	18.66	18.67	18.76	18.82	18.83
21.6	21.62	21.71	21.74	21.78	21.79	21.84	21.94	21.99	22.03
21.16	21.25	21.29	21.38	21.46	21.55	21.6	21.68	21.75	21.77
24.23	24.32	24.4	24.48	24.53	24.62	24.65	24.74	24.79	24.81
40.33	40.43	40.49	40.58	40.6	40.64	40.69	40.76	40.81	40.87
36.28	36.35	36.37	36.42	36.47	36.5	36.59	36.68	36.72	36.79
42.9	42.97	43.07	43.14	43.15	43.24	43.28	43.34	43.41	43.45
50.62	50.67	50.73	50.78	50.82	50.92	50.98	51	51.03	51.05
67.36	67.44	67.46	67.48	67.5	67.51	67.56	67.64	67.73	67.77
67.07	67.11	67.2	67.25	67.3	67.34	67.37	67.47	67.52	67.6

Figure 5

Initial Demand Dataset (Base)

```
demand vector
```

13	7	7	7	7	7	7	7	7	7
7	4	4	4	4	4	4	4	4	4
8	4	4	4	4	4	4	4	4	4
7	4	4	4	4	4	4	4	4	4
12	6	6	6	6	6	6	6	6	6
11	6	6	6	6	6	6	6	6	6
7	4	4	4	4	4	4	4	4	4
8	4	4	4	4	4	4	4	4	4
7	4	4	4	4	4	4	4	4	4
7	4	4	4	4	4	4	4	4	4
7	4	4	4	4	4	4	4	4	4
10	5	5	5	5	5	5	5	5	5

Figure 6

Initial Forecast Dataset (Base)

forecast vector

13	6.78	7.03	7.16	7.22	7.25	7.26	7.27	7.28	7.28
7	3.78	4.03	4.16	4.22	4.25	4.26	4.27	4.28	4.28
8	4.28	4.28	4.28	4.28	4.28	4.28	4.28	4.28	4.28
7	3.78	4.03	4.16	4.22	4.25	4.26	4.27	4.28	4.28
12	6.28	6.28	6.28	6.28	6.28	6.28	6.28	6.28	6.28
11	5.78	6.03	6.16	6.22	6.25	6.26	6.27	6.28	6.28
7	3.78	4.03	4.16	4.22	4.25	4.26	4.27	4.28	4.28
8	4.28	4.28	4.28	4.28	4.28	4.28	4.28	4.28	4.28
7	3.78	4.03	4.16	4.22	4.25	4.26	4.27	4.28	4.28
7	3.78	4.03	4.16	4.22	4.25	4.26	4.27	4.28	4.28
7	3.78	4.03	4.16	4.22	4.25	4.26	4.27	4.28	4.28
10	5.28	5.28	5.28	5.28	5.28	5.28	5.28	5.28	5.28

Figure 7

Active Algorithm vs Passive Algorithm graph

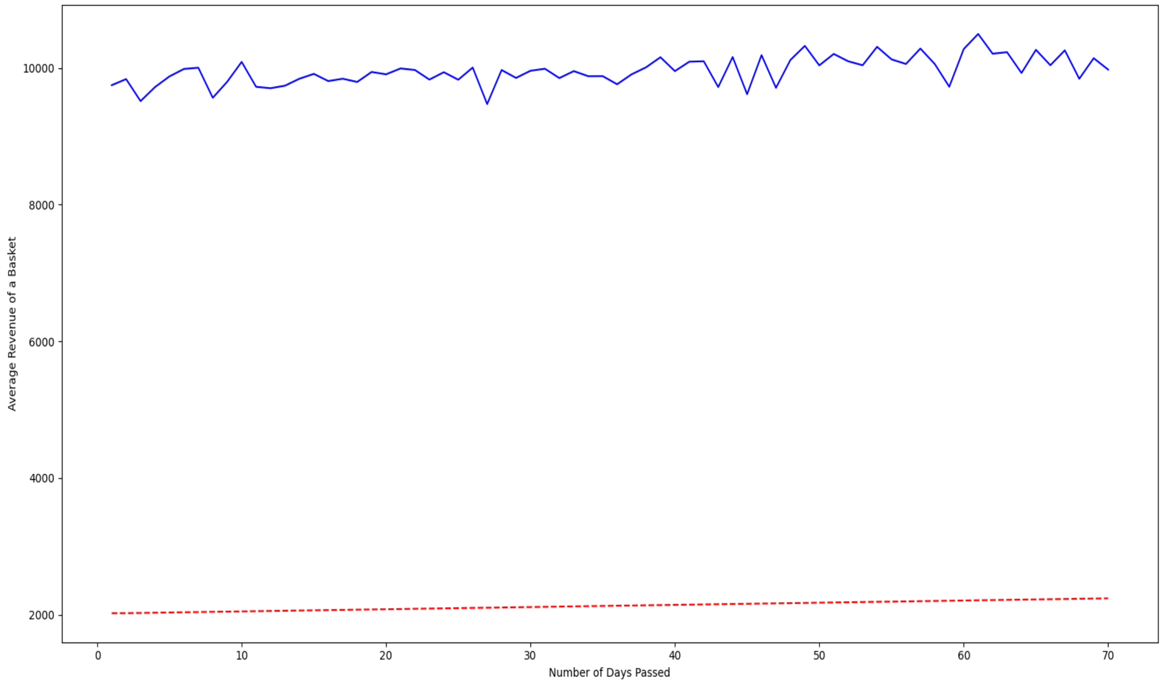

Figure 8 - dotted line (Passive) and continuous line (Active)

In this case it can be observed that the active algorithm consistently generates almost five times the revenue generated by the passive algorithm, over a prediction period of seventy days – Hence, active algorithm works effectively even when the price points of items in a basket vary significantly.

4.3 Experiment 3

Ten price and demand points were taken from an E-commerce dataset (link: https://www.kaggle.com/datasets/benroshan/ecommerce-data).

- **Number of items in a Basket:** 10
- **Composition of price of items in basket:** Prices of items in the basket vary significantly as they are taken from an e-commerce dataset.
- **Number of periods for which base dataset was created:** 10 days
- **Number of periods for which data was forecasted from the base dataset:** 60 days
- **Number of iterations:** 15

Initial Price Dataset (Base)

price vector

200	201	202	204	202	203	201	202	202	205
44	45	49	48	45	47	46	46	49	45
7	8	11	12	9	11	10	9	9	8
11	12	16	16	16	13	14	15	14	12
16	20	21	17	18	19	17	18	20	21
172	174	176	177	173	177	173	176	175	177
49	50	51	51	52	52	51	51	52	54
823	824	824	825	828	824	826	827	827	826
23	26	24	25	24	25	25	26	25	24
457	462	459	462	459	461	461	461	461	459

Figure 9

Initial Demand Dataset (Base)

demand vector

4	2	2	2	2	2	2	2	2	2
3	2	2	2	3	3	3	3	3	4
1	1	1	1	2	1	2	2	2	3
2	1	1	1	1	3	2	2	3	4
2	1	1	2	2	2	3	3	3	3
3	2	2	2	2	2	3	3	3	3
1	1	1	1	1	1	1	1	1	1
7	4	4	4	4	4	4	4	4	4
1	1	1	1	1	1	1	1	1	2
4	2	2	2	2	2	2	2	2	2

Figure 10

Initial Forecast Dataset (Base)

forecast vector

4	2.28	2.28	2.28	2.28	2.28	2.28	2.28	2.28	2.28
3	1.78	2.03	2.15	2.22	2.75	3.01	3.15	3.21	3.25
1	0.78	1.03	1.15	1.22	1.75	1.51	1.9	2.09	2.18
2	1.28	1.28	1.28	1.28	1.28	2.28	2.28	2.28	2.78
2	1.28	1.28	1.28	1.78	2.03	2.15	2.72	3	3.14
3	1.78	2.03	2.15	2.22	2.25	2.26	2.77	3.03	3.15
1	0.78	1.03	1.15	1.22	1.25	1.26	1.27	1.28	1.28
7	3.78	4.03	4.16	4.22	4.25	4.26	4.27	4.28	4.28
1	0.78	1.03	1.15	1.22	1.25	1.26	1.27	1.28	1.28
4	2.28	2.28	2.28	2.28	2.28	2.28	2.28	2.28	2.28

Figure 11

Active Algorithm vs Passive Algorithm graph

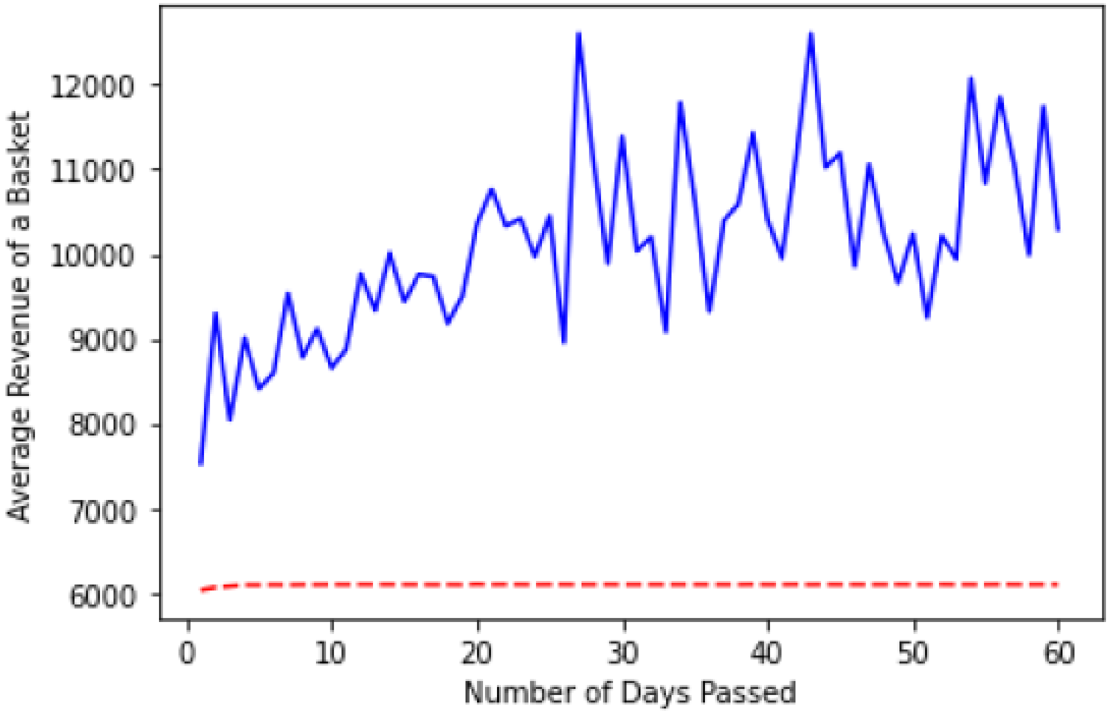

Figure 12 - dotted line (Passive) and continuous line (Active)

It can be seen that for an e-commerce dataset, the active algorithm greatly outperforms the passive algorithm (which is generating almost unchanged revenues even after 60 days). Hence using the bandit algorithm (Thompson Sampling) can be recommended for actual e-commerce datasets.

5. Conclusions and Future Scope

From the experiments conducted in Section 4 it is clear that the active bandit algorithm based on Thompson Sampling greatly outperforms the passive demand generation algorithm whether:

- The price basket is comprised on items with price points in a defined range (or)
- The price basket is comprised of items with varying price points (or)
- The price basket is taken directly from an e-commerce firm's historical demand and price data.

The main reason for this is that in case of the passive demand generation algorithm, elasticities are simply forecasted using historical values using some time-series model like ARMA/ARIMA; whereas in the case of the active bandit algorithm which leverages Thompson Sampling, elasticities are computed on a continuous basis by updating the Gaussian prior (using latest forecasted optimal revenue). This clearly showcases the power of the active algorithm.

Future investigations can include:

- Finding out the use-cases where the active bandit algorithm (Thompson Sampling), underperforms compared to other datasets showcased above.

- Instead of using revenue maximization as objective -- if 'volume of sales' is instead used as an objective function then what would be the performance of active vs passive dynamic pricing algorithms.

- Testing the performance of other 'Bandit Algorithms' compared to the 'Passive Maximum Revenue Optimization Algorithm'.

- Developing a framework for Regret Analysis, in an endeavor to better deal with modelling demand under uncertainty.

6. References

1. Aghion, Philippe, Bolton, Patrick, Harris, Christopher, and Jullien, Bruno. Optimal learning by experimentation. The review of economic studies, 58(4):621–654, 1991.

2. Araman, Victor F and Caldentey, Ren´e. Dynamic pricing for nonperishable products with demand learning. Operations Research, 57(5):1169–1188, 2009.

3. Auer, Peter, Cesa-Bianchi, Nicolo, and Fischer, Paul. Finite-time analysis of the multiarmed bandit problem. Machine learning, 47(2-3):235–256, 2002a.

4. Auer, Peter, Cesa-Bianchi, Nicolo, Freund, Yoav, and Schapire, Robert E. The nonstochastic multiarmed bandit problem. SIAM journal on computing, 32(1):48–77, 2002b.

5. Aviv, Yossi and Pazgal, Amit. A partially observed markov decision process for dynamic pricing. Management Science, 51(9):1400–1416, 2005.

6. Bertsekas, Dimitri P and Scientific, Athena. Convex optimization algorithms. Athena Scientific Belmont, 2015.

7. Besbes, Omar and Zeevi, Assaf. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. Operations Research, 57(6):1407–1420, 2009.

8. Broder, Josef and Rusmevichientong, Paat. Dynamic pricing under a general parametric choice model. Operations Research, 60(4):965–980, 2012.

9. Bubeck, S´ebastien, Cesa-Bianchi, Nicolo, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends R in Machine Learning, 5(1):1–122, 2012.

10. Caruana, Rich. Multitask learning. In Learning to learn, pp. 95–133. Springer, 1998.

11. Carvalho, Alexandre X and Puterman, Martin L. Learning and pricing in an internet environment with binomial demands. Journal of Revenue and Pricing Management, 3(4):320–336, 2005a.

12. Alexandre X. Carvalho and Martin L. Puterman, (2005), Dynamic Optimization and Learning: How Should a Manager set Prices when the Demand Function is Unknown ?, No 1117, Discussion Papers, Instituto de Pesquisa Econômica Aplicada

- IPEA.

13. Den Boer, Arnoud V. Dynamic pricing and learning: historical origins, current research, and new directions. Surveys in operations research and management science, 20(1): 1–18, 2015.

14. Den Boer, Arnoud V and Zwart, Bert. Simultaneously learning and optimizing using controlled variance pricing. Management science, 60(3):770–783, 2013.

15. Farias, Vivek F and Van Roy, Benjamin. Dynamic pricing with a prior on market response. Operations Research, 58(1):16–29, 2010.

16. Ferreira, Kris Johnson, Simchi-Levi, David, and Wang, He. Online network revenue management using thompson sampling. Harvard Business School Technology & Operations Mgt. Unit Working Paper. Available at SSRN: https://ssrn.com/abstract=2588730, 2016.

17. Gopalan, Aditya, Mannor, Shie, and Mansour, Yishay. Thompson sampling for complex online problems. In International Conference on Machine Learning, pp. 100–108, 2014.

18. Harrison, J Michael, Keskin, N Bora, and Zeevi, Assaf. Bayesian dynamic pricing policies: Learning and earning under a binary prior distribution. Management Science, 58(3):570–586, 2012.

19. Keskin, N Bora and Zeevi, Assaf. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. Operations Research, 62(5):1142–1167, 2014.

20. Phillips, Robert Lewis. Pricing and revenue optimization. Stanford University Press, 2005.

21. Rothschild, Michael. A two-armed bandit theory of market pricing. Journal of Economic Theory, 9(2):185–202, 1974.

22. Russo, Daniel, Van Roy, Benjamin, Kazerouni, Abbas, and Osband, Ian. A tutorial on thompson sampling. arXiv preprint arXiv:1707.02038, 2017.

23. Shah, Parikshit, Yang, Ming, Alle, Sachidanand, Ratnaparkhi, Adwait, Shahshahani, Ben, and Chandra, Rohit. A practical exploration system for search advertising. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1625–1631. ACM, 2017.

24. Talluri, Kalyan T and Van Ryzin, Garrett J. The theory and practice of revenue management, volume 68. Springer Science & Business Media, 2006.

25. Thompson,William R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika, 25(3/4):285–294, 1933.

26. Wasserman, Larry. All of statistics: a concise course in statistical inference. Springer Science & Business Media, 2013.

27. Ganti, Ravi, et al. "Thompson sampling for dynamic pricing." *arXiv preprint arXiv:1802.03050* (2018).

7. Appendix

The GitHub links for the code developed are listed below:

1. Revenue Optimization Passive vs Active for Synthetic Data: https://github.com/HarideepakM/MAX_REVENUE_OPT_Passive_vs_Active/blob/main/Max_Rev_Passive_vs_Active_main_Synthetic_Data.ipynb
2. Revenue Optimization Passive vs Active for Real Data: https://github.com/HarideepakM/MAX_REVENUE_OPT_Passive_vs_Active/blob/main/Max_Rev_Passive_vs_Active_main_Real_Data.ipynb