POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

A Multi-Armed Bandit Approach to Dynamic Pricing

Master Thesis of **Gianmarco Genalti**
Master of Science in Mathematical Engineering

Artificial Intelligence and Robotic Laboratory
of Politecnico di Milano

Student ID: 952905
Advisor: Prof. Nicola Gatti
Co-advisors: Marco Mussi, M.Sc., Alessandro Nuara, Ph.D.
Academic Year: 2020-2021

Abstract

International reports quantify that more than 14 trillion USD per year, since 2030, will be unlocked thanks to the automation of industrial and business processes. Automation of pricing tasks, in particular, is estimated to unlock value for about 0.5 trillion of USD per year worldwide. In this work we investigate this class of problems relying on fully data-driven methodologies. One of the main challenges in these settings usually concerns how scarce data can be effectively adopted to optimize the processes. We present a novel *dynamic pricing* approach that can incorporate business logic and operate in scenarios where industry needs are hard to assess for most of the other works. We focus on a monopolistic pricing problem on e-commerce with volume discounts and a strong seasonality, where the objective function is a convex combination between margins and turnover and only transaction data are available. We design an *online learning* (bandit) algorithm, namely PSV-B, in which the relationship controlling sales is decomposed into two terms, the first depending on seasonality and the second depending on the price elasticity. Furthermore, to alleviate data scarcity issues, we assume this relationship to be monotonically decreasing in price and we design a novel Bayesian regression algorithm capable of capturing such behavior. We develop a new methodology to compute optimal volume discounts starting from the prices proposed by our algorithm. This approach is evaluated both in synthetic environments and through a real-world experimental campaign. Model's design choices are validated performing multiple simulations, monotonicity comes up as a crucial feature when dealing with uncertainty and nonstationarity of the market. To quantify the business value unlocked by this approach, we performed a real-world, 4-month-long, A/B testing experiment, where our algorithm PSV-B, —corresponding to A configuration—has been compared with human pricing specialists—corresponding to the B configuration. At the end of the experiment, PSV-B produced a total turnover of about 300 KEuros with a performance that is better than the performance of B configuration for about 55%.

Keywords: Dynamic Pricing, e-commerce, Volume Discounts, Multi-Armed Bandit, Bayesian Linear Regression, Thompson Sampling.

Abstract in lingua italiana

Report internazionali stimano che, a partire dal 2030, l'automazione di processi industriali ed aziendali genererà valore per più di 14 trilioni di dollari. L'automazione del *pricing* di prodotti, in particolare, genererà circa 0.5 trilioni di dollari a livello mondiale. In questo lavoro, studiamo questa classe di problemi basandoci esclusivamente su metodologie *data-driven*. Una delle sfide principali in questo contesto è quella di costruire una soluzione in grado di operare anche in scarsità di dati. Noi presentiamo un nuovo algoritmo di pricing dinamico che sia in grado di incorporare logiche di business e operare in scenari in cui i bisogni delle aziende sono difficili da fronteggiare per le altre soluzioni esistenti. Ci concentriamo su un problema di pricing monopolistico su un e-commerce, dove vige una forte stagionalità e una politica di scontistica per quantità. La funzione obiettivo sarà una combinazione convessa del margine netto operativo e del fatturato, e l'algoritmo avrà accesso solamente a dati transazionali. In questo lavoro progettiamo un algoritmo di *online learning* (bandit), chiamato PSV-B, dove i volumi di vendita sono decomposti in due componenti: la prima dipendente dalla stagionalità e dal trend di mercato, mentre la seconda dalla curva di domanda dei clienti. Inoltre, per mitigare la scarsità di dati, assumiamo che all'aumentare dei prezzi i volumi decrescano. Proponiamo un nuovo modello di regressione Bayesiana integrandovi la relazione monotona tra i volumi e la seconda componente. Sviluppiamo una nuova metodologia che, partendo dal prezzo proposto dal nostro algoritmo di pricing, calcoli una politica di scontistica per quantità ottimale. Abbiamo valutato questo approccio sia in ambienti simulati che con una campagna sperimentale reale. Le scelte di design del modello sono state validate attraverso molteplici simulazioni, la monotonicità si è rivelata una caratteristica determinante per gestire la rumorosità dei dati e la nonstazionarietà nella curva di domanda. Per quantificare il valore economico di questo approccio, abbiamo effettuato un A/B test lungo 4 mesi su un e-commerce italiano. Il nostro algoritmo, PSV-B, è stato valutato rispetto ad un pricing effettuato da specialisti umani. Alla fine dell'esperimento, PSV-B ha prodotto un fatturato totale per 300.000 Euro, con una performance migliore rispetto alla configurazione B di circa il 55%.

Parole chiave: Pricing Dinamico, e-commerce, Scontistica per Quantità, Multi-Armed Bandit, Regressione Lineare Bayesiana, Thompson Sampling.

Contents

List of Figures

List of Tables

1 | Introduction

Dynamic Pricing refers to a family of techniques used to learn the optimal price of a product or service in a real-time fashion. Due to the strict connection with the economic sphere and the technical possibilities that are evolving in neighboring sectors, dynamic pricing is getting a lot of attention both from the industrial world and the scientific community.

1.1. Goal

Our goal is the design of a framework which, on the one side, is scientifically sound and, on the other side, is ready-to-use for the companies' business units. While the *Artificial Intelligence* scientific community has been developing new data-driven approaches year after year, thus pushing the state-of-the-art forward, companies have been attempting to transfer those results to their business needs or to build novel tailored solutions. Nowadays, a crucial task is to close the gap between the scientific state-of-the-art and industrial needs. In our work, we move toward this direction, proposing a novel solution and validating it through an experimental evaluation performed in a real-world scenario. Furthermore, our solution adds a new layer to the data-driven dynamic pricing which was not sufficiently explored previously. This layer concerns the data-driven volume discount policy. We remark, that most of the scientific works in dynamic pricing propose algorithmic results without a concrete experimental evaluation in real-world settings. This is a critical issue, as all the known works are based on assumptions whose satisfaction in practice may be questionable. Instead, in our work, we present a framework that has been validated with simulations and a real-world experiment lasted more than 4 months and involving products for a total turnover of 160 KEuros. Our solution accounts many crucial issues characterizing real-world scenarios (*i.e.* seasonality, volume discounts) and campaign's results validate its goodness and practical applicability.

1.2. AI for Pricing in Industry

Most of the international economic forecasts agree that almost 50% of the value per year unlocked by the adoption of artificial intelligence (AI) from 2030 on will be in marketing&sales [16] and this value amounts about 6 trillion USD. Attracting and acquiring new customers, suggesting and recommending products, and optimizing customers' retention and loyalty are examples of activities in which AI tools will play a role of paramount importance, allowing their automation and thus dramatically increasing their effectiveness. In the specific case of pricing problems—which constitute the class of problems we investigate in this work—, the estimated value unlocked will be about 0.5 trillion USD per year. The primary challenge in these settings usually concerns how scarce data can be effectively adopted to optimize the processes by combining data-driven and shape-constrained approaches.

In this work, we focus on a monopolistic pricing problem on e-commerce with volume discounts and seasonality in which the objective function is a convex combination between margins and turnover and only transaction data are available.

In particular, companies may have different objective functions, depending on the specific market segment, business constraints and time horizon, so as to change gradually from the maximization of margins to the maximization of turnover. Controlling this trade-off is crucial for the e-commerce management since it allows to implicitly balance the instantaneous revenue generated by new customers and a longer-term revenue generated by returning customers. Interestingly, when optimizing this process, volume discounts can play a crucial role.

The primary challenge in these settings usually concerns how scarce data can be effectively adopted in online fashion to optimize the processes by combining model-based and model-free approaches. In particular, most of the works available in the literature [7, 32, 48] do not address a number of tasks that are central in practical applications, preventing their successful adoption. For instance, no algorithm works in online fashion, thus not being capable of adapting quickly to the customers' behavior. Furthermore, no learning algorithm deals with seasonality and volume discount.

1.3. AI for Pricing in Scientific Literature

Due to the widespread of e-commerce and online marketplaces over the last years, data collection is now easier than ever and data-driven approaches to dynamic pricing are receiving increasing attention from the scientific community due to the possibility of easily performing experimental evaluations. This task has been addressed from many different points of view: from classical economics to reinforcement learning, involving optimization techniques, Statistical Learning methodologies, and Machine Learning. Each scientific community has focused on a different perspective. Many works coming from operational research and management science journals focus, for example, on the joint pricing-restocking problem, with a particular interest for perishable goods and limited inventory (See [21], [30], [15]). Literature of Machine Learning and Statistical Learning is divided between two main strands: Bayesian modeling and non-parametric approaches. While in the former an attempt is made to incorporate business and market knowledge inside demand function modeling, the latter tries to be as general as possible to cover the majority of possible business scenarios. One of the major tradeoffs is interpretability, where Bayesian approaches usually outperform non-parametric ones (See [6], [17], [8]). Recently, Online Learning and Reinforcement Learning communities joined this research trend and started proposing pricing policies that can handle the *exploration-exploitation* dilemma (See [46], [48], [34]).

1.4. Original Contributions

In our work, we decompose the demand curve into two terms. The first term depends on the seasonality and market trend, where seasonality captures the yearly periodic customers' behavior, while the trend captures the market contraction or expansion of a specific year. The second term depends on the price elasticity. This decomposition allows an efficient use of the data available in practice. Motivated by goods different from luxury, Veblen, and Giffen, we assume that the demand curve is monotonically decreasing in the price and we force such an assumption in the learning algorithm. This is captured by designing a novel Bayesian regression algorithm that forces a subset of features to be monotonic. Furthermore, we address the exploitation-exploration dilemma by using a Thompson-sampling-like approach, which randomly draws samples according to the degree of uncertainty of the estimates. This approach allows the algorithm to explore with more probability the prices providing a better optimistic confidence bound on the reward. Our algorithm also computes volume discounts, adapting such discounts to users need, given the buyback probability. We name our algorithm as PSV-B (Pricing with Seasonality and Volume discounts Bandit algorithm). We performed a real-world 4-month-long A/B testing experiment, in which our algorithm PSV-B—corresponding to A configuration—has been compared with human pricing specialists—corresponding to B configuration. The total number of different products involved in the experiment is more than 320. At the beginning of the test, the algorithm received information related to the orders received in the previous 2 years. The total turnover of A configuration is more than 300 KEuro. At the end of the experiment, our algorithm PSV-B provided a performance that is better than the performance of B configuration for about 55%.

1.5. Work Structure

The remaining of this work is organized as follows.

- *Chapter 2*: An introduction to the methodologies and techniques which act as a backbone to the proposed solutions: we span from Linear Basis Function Regression to Multi-Armed Bandits. We also provide an overview of the related literature in dynamic pricing and volume discounts.
- *Chapter 3*: The problem formulation: formal definition of the variables characterizing the problem and definition of the objective of the algorithm.
- *Chapter 4*: Proposed solution to the problem, formalization of the PSV-B algorithm and its framework.

- *Chapter 5*: Validation of the proposed solution, first using a simulated setting to validate assumptions and design choices, then in a real-world scenarios to assess the goodness of the solution for business purposes.
- *Chapter 6*: The conclusions to be drawn from this work, along with new settings to which potentially apply the proposed solution of this work or possible extensions of it.

2 | Preliminaries and Related Works

In this chapter, we introduce the mathematical groundings needed in our work. In particular, we introduce the mathematical formulation of Linear Basis Function Models and their extension to a Bayesian framework with Bayesian Linear Regression [10]. Subsequently, we introduce Multi-Armed Bandit problems [44], focusing on Bayesian approaches to deal with the exploration-exploitation dilemma such as, *e.g.*, Thompson Sampling. We are then ready to introduce the dynamic pricing problem from a Multi-Armed Bandit point of view, highlighting prominent works using this approach addressing this problem. Finally, related works on dynamic pricing using different methodologies are presented, with a section dedicated to volume discounts.

2.1. Linear Models

2.1.1. Linear Basis Function Models

We focus on the scenario in which a sample is of the form $\mathbf{x} = (x_1, \dots, x_D) \in \mathbb{R}^D$. The simplest linear model for regression is one that involves a linear combination of the input variables:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_D x_D. \tag{2.1}$$

The model is described by a linear function of the parameters (or *weights*) $\mathbf{w} = (w_0, \dots, w_D)$, in this particular case it is also a linear function of the input variables $\mathbf{x}$. We want to extend this formulation to grasp nonlinear relationships between input variables and the response function y, to do so we introduce *basis functions*. A basis function Φ_j maps an input sample to a value in $\mathbb{R}$. This results in a new formulation of Eq. 2.1:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \Phi_j(\mathbf{x}). \tag{2.2}$$

The model now has M parameters, where w_0 is the *bias* parameter and, defining $\Phi_0(\mathbf{x}) = 1 \; \forall \mathbf{x} \in \mathbb{R}^D$, we can define Eq. 2.2 compactly as:

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \Phi(\mathbf{x}), \tag{2.3}$$

where $\Phi(\mathbf{x}) = (\Phi_0(\mathbf{x}), \dots, \Phi_{M-1}(\mathbf{x}))$.

2.1.2. Maximum Likelihood Estimation

We assume that the target variable t is the superposition of the response function of the model and a *Gaussian noise*:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, \frac{1}{\beta})$. Parameter β is customarily called the *precision* of the noise. The previous relationship implies a distribution over the target variable:

$$p(t|\mathbf{x}, \mathbf{w}, \beta) \sim \mathcal{N}\left(y(\mathbf{x}, \mathbf{w}), \frac{1}{\beta}\right), \tag{2.4}$$

where $\mathbb{E}[t|\mathbf{x}] = y(\mathbf{x}, \mathbf{w})$.

Given a dataset of inputs $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and target values $\mathbf{t} = (t_1, \dots, t_N)$, we can obtain the likelihood function of $\mathbf{t}$ as:

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{w}^T \Phi(\mathbf{x}_n), \frac{1}{\beta}). \tag{2.5}$$

The goal is to obtain the parameters' values maximizing the likelihood of the target variable. In particular, to obtain $\mathbf{w}_{ML}$ we can minimize the gradient:

$$\nabla \ln p(\mathbf{t}|\mathbf{w}, \beta) = \sum_{n=1}^{N} (t_n - \mathbf{w}^T \Phi(\mathbf{x}_n)) \Phi(\mathbf{x}_n)^T. \tag{2.6}$$

By defining the $N \times M$ *design matrix* as

$$\mathbf{\Phi} = \begin{pmatrix} \Phi_0(\mathbf{x}_1) & \Phi_1(\mathbf{x}_1) & \dots & \Phi_M(\mathbf{x}_1) \\ \Phi_0(\mathbf{x}_2) & \Phi_1(\mathbf{x}_2) & \dots & \Phi_M(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_0(\mathbf{x}_N) & \Phi_1(\mathbf{x}_N) & \dots & \Phi_M(\mathbf{x}_N) \end{pmatrix},$$

we get that

$$\mathbf{w}_{ML} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi} \mathbf{t}. \tag{2.7}$$

Furthermore, the maximization of the log likelihood function w.r.t. to β results in

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} (t_n - \mathbf{w}_{ML}^T \Phi(x_n))^2. \tag{2.8}$$

2.1.3. Bayesian Linear Regression

To follow the Bayesian perspective of linear models, we introduce a prior over the parameters of the model. In particular, we have:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{m}_0, \mathbf{S}_0), \tag{2.9}$$

where $\mathbf{m_0}$ and $\mathbf{S_0}$ are prior values over mean and variance of $\mathbf{w}$.

The posterior distribution can be obtained as a product of the prior and the likelihood. The choice of a Gaussian prior allows us to get a Gaussian posterior in the form of:

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{m}_N, \mathbf{S}_N), \tag{2.10}$$

where

$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta \mathbf{\Phi}^T \mathbf{t}), \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta \mathbf{\Phi}^T \mathbf{\Phi}. \end{aligned} \tag{2.11}$$

Note that the mode of the posterior distribution coincides with the mean as it is Gaussian. Thus, the maximum posterior weight vector is simply given by $\mathbf{w}_{MAP} = \mathbf{m}_N$.

Bayesian Linear Regression and Online Learning

If data points arrive sequentially, then the posterior distribution at any stage acts as the prior distribution for the subsequent data point, such that the new posterior distribution is again described by Eq. 2.10. Thanks to this latter property, we can easily use BLR in an *online learning* setting, where data arrive from time to time and we do not need to train the whole model every time.

2.1.4. Bayesian Nonparametric Monotone Regression

We focus on the scenario in which the relationship between the target variable and the input features is known to be monotonic due to the physical process involved. This setting is of particular interest for dynamic pricing, since the demand curve can be assumed to be decreasing in price for non-luxury products. To deal with this property in a Bayesian regression setting, we introduce a particular basis function expansion called *Bernstein Polynomial expansion*. The k-th Bernstein Polynomial basis function of order M is defined as:

$$\psi_k(x, M) = \binom{M}{k} x^k (1-x)^{M-k}, \;\; x \in [0,1], \tag{2.12}$$

thus, the regression formulation results in the following weighted combination

$$f(x) = \sum_{k=0}^{M} \psi_k(x, M)\beta_k = \boldsymbol{\Psi}\boldsymbol{\beta}. \tag{2.13}$$

The function in Eq. 2.13 is monotone if $\beta_0 = 0$ and $\beta_k \geq \beta_{k-1}$ for all $k = 1, \dots, M$. Following the same procedure used by McKay Curtis and Ghosh [31], we perform a reparametrization of the regression coefficients as $\boldsymbol{\theta} = A\boldsymbol{\beta}$, where

$$A = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \end{pmatrix}.$$

The resulting regression function, substituting in Eq. 2.13, is

$$f(x) = \boldsymbol{\Psi} A^{-1} \boldsymbol{\theta}. \tag{2.14}$$

Now, we can obtain a monotonically increasing function by imposing $\theta_k \geq 0$ for all $k = 1, \ldots, M$. In Fig. 2.1 are represented the Bernstein Polynomials basis function and its transformed version. In order to obtain a downward monotonic regression model is sufficient to flip the basis functions using $1 - \Psi A^{-1}$ instead of ΨA^{-1}.

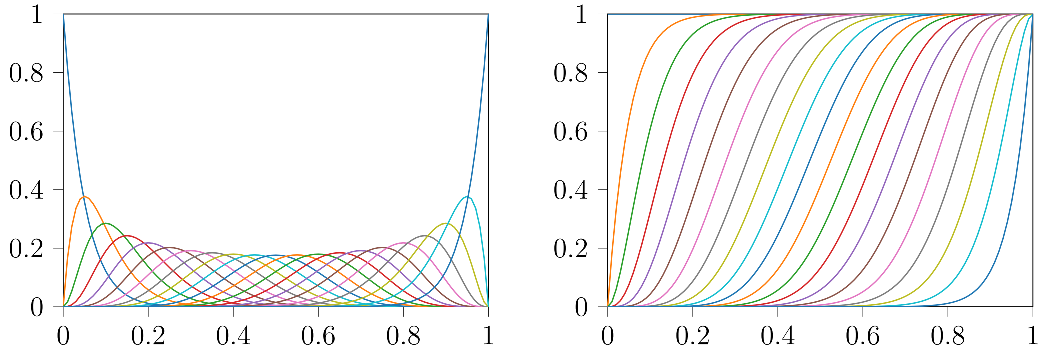


Figure 2.1: On the left, the 21 Bernstein Polynomials basis function Ψ of order $M = 20$, on the right their monotonic transformation $\boldsymbol{\Psi} A^{-1}$.

In Bayesian Linear Regression settings, if the probability distributions over the transformed coefficients $\boldsymbol{\theta}$ have positive support, we straightforwardly obtain positive values for them. The simplest choice is to use a *Lognormal* prior distribution:

$$\boldsymbol{\theta} \sim \mathcal{LN}(\boldsymbol{\theta}_0, \Sigma_0), \tag{2.15}$$

where $\boldsymbol{\theta}_0 \in \mathbb{R}^M$ and $\Sigma_0 \in \mathbb{R}^{M \times M}$. The posterior distribution can be estimated either via sampling approaches like *Hamiltonian Monte Carlo* [9] or via *Variational Inference* tehcnique [11] without relying on a closed form distribution.

2.2. Multi-Armed Bandits

In a MAB setting, the feedback can be either positive or negative, corresponding to a reward or a loss, respectively. The goal of the agent is to learn which option is the best, gaining the maximum possible reward or minimizing its cumulative loss while doing so: this is called *exploration-exploitation trade-off*. Such a task is crucial in many practical applications such as clinical trials [50], financial portfolio design [43], advertising campaigns optimization [36] and, last but not least, dynamic pricing [46, 48].

Let us define a set of k actions $\mathcal{A}$: each of the actions provides an expected or mean reward when played. If we assume to have a time horizon of N timestamps, we can define as A_t the action undertaken by the agent at timestamp $t \leq N$, with a reward of R_t. The goal is to learn the *value* of an action a, defined as:

$$q(a) \coloneqq \mathbb{E}[R_t | A_t = a].$$

Actions' values solve the MAB problem: an agent knowing which action has the higher value (namely, a *clairvoyant*) would always select that, resulting in a trivial strategy. However, rewards are drawn from unknown probability distributions and the agent has to deal with this uncertainty. So, in practice, knowing the true value of $q(a)$ is usually impossible even after a long exploration, the best that an agent can do is to produce a real-time estimation denoted as $Q_t(a)$, representing the estimated value of action a at time t. The estimation process can be performed in multiple ways and the choice of a way of estimation also affects the exploration behavior of the agent. Once an exploration method is selected, the agent is able to produce $Q_t(a)$ for each $t \leq N$: clearly $Q_t(a)$ is a real number for every $a \in \mathcal{A}$ and for every $t \leq N$, so at each timestamp t always exists $\max_{a \in \mathcal{A}} Q_t(a)$ We refer to a *policy* as a probability distribution over the possible actions of the agent: as time passes, the agent updates its policy towards the optimal one, the one known by the clairvoyant from the very beginning. A policy is said to be *greedy* ,if it puts all its probability mass over the actions maximizing $Q_t(a)$, in other words always choosing the action $a_t^* = \arg\max_{a \in \mathcal{A}} Q_t(a)$. Given a policy $\mathfrak{U}$, returning at every timestamp t an action a_t, we define the *pseudo-regret* of the policy as follows:

$$R_T(\mathfrak{U}) \coloneqq \sum_{t=1}^{T} q(a_t^*) - \mathbb{E}\left[\sum_{t=1}^{T} Q_t(A_t)\right], \tag{2.16}$$

where $q(a_t^*)$ is the expected value provided by a clairvoyant algorithm. The goal of the agent will be the minimization of the pseudo-regret $R_T(\mathfrak{U})$.

2.2.1. Thompson Sampling

Mathematical properties of bandit strategies have been extensively investigated across the last 20 years [5, 14], with a lot of attention on the Bayesian framework. *Thompson Sampling* is a golden standard among the Bayesian heuristics used to solve exploration-exploitation dilemma, and results on its convergence properties can be found in [2, 25, 37, 39]. The common assumption is that the reward R is drawn from a distribution defined by a set of parameters $\boldsymbol{\theta} \in \Theta$. Let $\mathbb{P}(\boldsymbol{\theta})$ be the prior distribution over the weights and $\mathcal{D} = \{(a_i, R_i)\}_{i=1:t}$ the action-reward tuples up to timestamp $t \leq N$. Using the Bayes Theorem, we can infer over the posterior distribution of $\mathbb{P}(\boldsymbol{\theta})$:

$$\mathbb{P}(\boldsymbol{\theta}|\mathcal{D}) \propto \mathbb{P}(\mathcal{D}|\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta}),$$

where $\mathbb{P}(\mathcal{D}|\boldsymbol{\theta})$ is the likelihood function. Using the posterior distribution on the weights, it is possible to compute a probability distribution over the reward of each action. In a Thompson Sampling strategy, the chosen action $a^* \in \mathcal{A}$ is the one with the maximum reward's expected value:

$$a^* = \arg\max_{a \in \mathcal{A}} \mathbb{E}[R_t|A_t = a, \boldsymbol{\theta}].$$

In Algorithm 1 the pseudo-code of a basic Thompson Sampling strategy is provided.

Algorithm 1 Thompson Sampling

```
Require: Time horizon $N \geq 0$, set of actions $\mathcal{A}$, priors $\mathbb{P}(\boldsymbol{\theta})$.
Ensure: Sequence $\mathcal{D} = \{(a_i, R_i)\}_{i=1:N}$.
$\mathcal{D} = \{\}$
for $t$ in $1 : N$ do
    $\mathbb{P}(\boldsymbol{\theta}|\mathcal{D}) \leftarrow \mathbb{P}(\mathcal{D}|\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta})$
    Sample $\boldsymbol{\theta}$ from $\mathbb{P}(\boldsymbol{\theta}|\mathcal{D})$
    $a^* \leftarrow \arg\max_{a \in \mathcal{A}} \mathbb{E}[R_t|A_t = a, \boldsymbol{\theta}]$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(a^*, R_t)\}$
end for
```

2.3. Dynamic Pricing

Dynamic Pricing refers to all algorithms used to modify prices of goods and services in an automated fashion, so as to learn optimal pricing policies.

2.3.1. MABs for Dynamic Pricing

Multi-Armed bandits have been extensively employed for dynamic pricing. Pricing policies heavily affect revenue and profit in many retail businesses and a sub-optimal pricing can lead to dramatic monetary losses. At the same time, optimal price has to be learned in some way. The resolution of this crucial exploration-exploitation trade-off let MAB be a suitable option: while learning the optimal price, there are guarantees on the losses in which the algorithm may occur during the learning process. Suppose we are pricing a given good, in a MAB setting for pricing we let the actions correspond to the possible prices, moreover a retailer can decide the time between two price changes that more suits its own business logic, deciding to change the price each hour, day, week or month. The reward consists in the product between the *demand curve*, that is the relationship between prices and sales quantities, and the corresponding prices. Retailers are seldom able to collect contextual information on customers or goods, but are more likely to collect transaction data and records of sells [47]. Aggregating transaction data over a given period of time allows us to associate prices to demand values in each timestamp in the required form of $\{a_t, R_t\}$.

Rothschild [40] provides one of the seminal works on the adoption of MAB algorithms for dynamic pricing. In this work, the author introduces a basic scenario in which a two-armed bandit has to decide which price is better between the two options. Using this work as a starting point, many others tried to face the dynamic pricing problem using MABs.

Kleinberg and Leighton [28] study the scenario in which a MAB strategy has to learn a continuous-demand function and proposes a discretization of the price values to provide theoretical guarantees on the algorithm's regret. This approach suffers from the drawback that the reward is assumed to admit a unique maximum in the price. Such an assumption is very restrictive in practice.

In Misra et al. [32] monotonic property of the demand function is used to guarantee a faster convergence. However, monotonicity is not forced as a model-specific feature, so decisions violating business logic can still be made during the learning process.

Trovò et al. [46, 48] assume that the demand function is monotonically decreasing and exploit this assumption in the learning algorithm to provide uncertainty bounds tighter than those of classical frequentist MAB algorithms, but neither monotonicity nor weak monotonicity are imposed by the model formulation to the estimated demand functions. The authors show how the monotonicity assumption does not improve the asymptotic bound of regret provided by MAB theory. On the other hand, exploiting monotonicity enables an empirical improvement in the performances, thus reducing the constant terms of the regret bound.

Research in this area has been primarily restricted to stationary settings (Besbes and Zeevi [7], Den Boer [19], Keskin and Zeevi [27], Kleinberg and Leighton [28]), in our work we address the non-stationarity environment problem by resorting to a multivariate fit of the demand function.

2.3.2. Related Works on Dynamic Pricing

Besbes and Zeevi [8] show that linear models are a suitable and efficient tool to model a demand function. In their work, downward monotonicity is forced on a model-wise level, but it is only analyzed in a stationary environment.

Other works adopting a parametric formulation of the demand function are [12] and [7]. These works assume a stationary behavior of the customers.

Cope [17] and Bauer and Jannach [6] are two of the main works concerning Bayesian inference applied to dynamic pricing. They both fail in imposing monotonic constraints on the model. Interestingly, Bauer and Jannach [6] take into account non-stationary features (*e.g.*, competitors' prices).

Araman and Caldentey [4] use a Bayesian approach for dynamic pricing: here market-related information is captured in the model through a prior belief on parameters and the model has a monotonic formulation on the demand function.

Wang et al. [51] investigate non-parametric models for demand function estimation. In this case, the authors assume that the demand function is smooth.

Nambiar et al. [35] propose a model to tackle both the non-stationarity data and the model misspecification. However, contextual knowledge is required on a product-wise level that is usually not available to retailers.

In Table 2.1 we summarize the main assumptions and design choices behind relevant dynamic pricing works. Also, in the first row, this work is present. Despite most works, we do not assume customers' elasticity to be stationary: instead, we expect that the associated (decreasing) price-volumes curve may change over time. Due to this reason, our algorithm has to handle a broad family of possible demand functions that may allow more than one maximum over the reward function's domain. Another strength of our algorithm is that it only relies on transaction data, so no contextual information on customers and products is needed.

	Environment Assumptions				Model Design Choices				
	Monopolistic Market	**Unlimited Inventory**	**Non-Perishable Goods**	**Stationary Elasticity**	**Parametric Model**	**Monotonic Elasticity**	**Unimodal Reward**	**Context Data**	**Bandit Approach**
This Work	✓	✓	✓		✓	✓			✓
Araman [4]	✓		✓	✓	✓	✓	✓		
Bauer [6]		✓	✓					✓	
Besbes [7]	✓		✓	✓	✓*	✓*	✓*		
Besbes [8]	✓	✓	✓	✓	✓	✓	✓		
Broder [12]	✓	✓	✓	✓	✓	✓	✓		
Bu [13]	✓	✓	✓	✓	✓	✓	✓		✓
Cao [15]	✓		✓		✓	✓			
Cope [17]	✓	✓	✓	✓	✓				
Gallego [21]				✓*	✓	✓	✓	✓	
Harrison [22]	✓	✓	✓	✓	✓	✓			
Javanmard [24]	✓	✓	✓		✓	✓	✓	✓	
Keskin [27]	✓	✓	✓	✓	✓	✓	✓		
Kleinberg [28]	✓	✓	✓	✓			✓		✓
Levina [30]	✓								
Nambiar [35]	✓	✓	✓		✓	✓	✓	✓	
Misra [32]	✓	✓	✓	✓			✓		✓
Rothschild [40]	✓	✓	✓	✓					✓
Trovò [46]	✓	✓	✓	✓			✓		✓
Trovò [48]	✓	✓	✓	✓			✓		✓
Wang [51]	✓	✓*	✓	✓					✓

* Work present multiple versions of the algorithm and includes also the complementary condition.

Table 2.1: Environment assumptions and model design choices across the most relevant works in literature.

2.4. Volume Discount

With *volume discount*, we mean the possibility of offering multiple prices to a customer, bound to a minimum quantity of product's units to be purchased in order to get a certain price. Literature lacks a data-driven methodology to find an optimal volume discount pricing schedule aimed at maximizing retailer's profits and revenues in a B2C environment.

In Monahan [33], benefits of such a pricing schedule are described from a vendor's point of view: the model, anticipating the buyer's behavior, should increase average orders' size, allowing the retailer to access supplier's rebate on large restocks, reduce processing costs and anticipate cash flow through the budget year.

Rubin and Benton [41] and Hilmola [23], focus on the *Economic Order Quantity* (EOQ) model that requires demand size over an annual budget and stock size.

Sadrian and Yoon [42] relax the hypothesis of EOQ and provides a rational pricing strategy and then calculating volume thresholds and the corresponding discounts afterwards. In that work, the authors show the importance of volume discounts when increasing the sales of higher priced products.

3 | Problem Formulation

In this chapter, we study the scenario in which a monopolistic e-commerce website sells a set of non-perishable products with unlimited availability and the demand function is monotonically decreasing in the price, possibly nonstationary over time. The two assumptions, common in literature and practical scenarios, are motivated by the efficient, large stocking possibilities of modern retailers (such as e-commerce) and by the fact that we are assuming to price non-Giffen [20] and non-Veblen goods [26].

3.1. Pricing Formulation

We are given a set of products which are not subject to dependencies, and we are asked to find their optimal prices. For the sake of presentation, thanks to the independence assumption, in the following we focus on the problem of pricing a single product.

We denote with $t \in \mathcal{T} = \{0, \dots, T\}$ a *time unit* for our problem, the chosen unit may vary from retailer to retailer, depending on their business logic, *e.g.*, a hour, day, week, month. At every time t, we are faced with the choice of a, potentially different, price $p_t \in \mathcal{P}$, where $\mathcal{P}$ is a finite set of feasible values of price. Furthermore, at every time t, there is a number of potential customers interested in buying the product, and each of them is proposed the same price p_t. The actual average number of sales (a.k.a. volumes) at time t when choosing price p_t is denoted with $v_t(p_t)$. In particular, we assume that the volumes depend on both price and time due to, *e.g.*, seasonality and market trend, and we denote the volumes curve function with $\mathcal{V}(p_t, t)$, where $\mathcal{V} : \mathcal{T} \times \mathcal{P} \to \mathbb{R}^+$. At every time t, we have $v_t := \mathcal{V}(p_t, t)$. Finally, every unit sold, the agent gains a margin $m_t := p_t - c$, where $c \in \mathbb{R}^+$ is the cost of the product. In this formulation, the cost of a product is assumed to be constant: in this corner scenario, customers' reaction to price and to net margin are assumed to be equal, allowing us to study price-elasticity effect directly on margins. This usually happens when costs are assumed to be similar across the whole industry (*i.e.* competitors share the same supplier, a common setting in many e-commerce scenario), an investigation of costs' spread across actors in the same market can be found in [18]. The objective function to maximize is defined as a convex combination with parameter $\lambda \in [0,1]$ between *turnover* and *operating cash flow margin*. Formally, the maximization problem is as follows:

$$p_t^* = \operatorname*{argmax}_{p_t \in \mathcal{P}} f(p_t), \tag{3.1}$$

where:

$$f(p_t) = \lambda \frac{p_t \; v_t(p_t)}{\max_{p_t \in \mathcal{P}}\{p_t \; v_t(p_t)\}} + (1-\lambda)\frac{m_t \; v_t(p_t)}{\max_{p_t \in \mathcal{P}}\{m_t \; v_t(p_t)\}}. \tag{3.2}$$

The first term of the right hand of Eq. (3.2) is the turnover at time t when choosing price p_t, and $\max_{p_t}\{p_t \; v_t(p_t)\}$ is the maximum achievable value. The second term of the right hand of Eq. (3.2) is the operating cash flow margin at time t when choosing price p_t, and $\max_{p_t}\{m_t \; v_t(p_t)\}$ is the maximum achievable value. The need for normalization of the two components of Eq. (3.2) is given by the fact that the two have to be of the same magnitude in order to provide a meaningful balancing: typically, revenue and profit are in a whole different range of values. Moreover, the normalization needs a dependency on the price to avoid maximizing Eq. (3.2) simply maximizing the rough price.

In real-world scenarios, functions $\mathcal{V}(p_t, t)$ and, *a fortiori*, $v_t(p_t)$ are not *a priori* known and need to be estimated online. Thus, our problem can naturally formulated as an online learning problem (see, *e.g.*, Auer et al. [5] for a comprehensive survey on online learning) where the goal is to properly balance the acquisition of information on the stochastic functions, while minimizing the cumulative regret. Such a problem is also commonly known as exploration-exploitation dilemma.

Formally, in an online learning problem, we are given a set of available options (a.k.a. *arms*), and we can choose an arm per time i. In our case, the arms are the possible values of price $p_t \in \mathcal{P}$, while $\mathcal{V}(\cdot,\cdot)$ and $v_t(\cdot)$ is a stochastic function that we need to estimate during the time horizon T.

3.2. Volume Discounts

Retailer's revenue is usually given by both walk-in and loyal customers. For the first group, the retailer aims at maximizing the revenue given by the single basket they generate. For the second ones, the retailer aims at providing offers that match with their volumes need. A common way to deal with these issues is to provide volume discounts, *i.e.* providing different prices depending on the number of units bought by the customer.

Assuming that the price does not affect the customer needs for a given product, with volume discounts we try to sell more units at a lower price to mitigate the risk of customers meeting their needs with future purchases from our competitors. In order to define the problem, we consider that our customers have a need N for the considered product and we estimate the probability γ that a generic customer will buyback that product from our shop. The value of N acts as a hyper-parameter of the problem that implicitly defines the horizon along which the effects of the discounts are to be evaluated: the higher is N, the greater the number of repurchases that will be considered.

Consider a vector of η volume thresholds $\boldsymbol{\omega} = [\omega_1, \omega_2, \ldots, \omega_\eta]$, with $\omega_i > \omega_h, \forall i > h$ and $\omega_1 = 1$. The price of the product is a piece-wise constant function of the volume, which assigns the same price to all volumes between two consecutive volume thresholds. Let $p_t^{(i)}$ denote the price associated with the volumes between ω_i (included) and ω_{i+1} (excluded).

The goal is to define the discount δ_i that we can apply to the price for a unit volume ($\bar{p} = c + \bar{m}$) in order to get the price for the i-th volume range. To avoid negative margins, we apply the discount directly to the margin: $p_t^{(i)} = c + (1 - \delta_i)\bar{m}$. The discount δ_i should guarantee, for a customer who needs N product units and has a buyback probability γ, that the expected margin with multiple-unit orders is no lower than the one obtainable with N single-unit orders.

4 | Proposed Algorithm

The goal of this algorithm is to propose a pricing schedule for a given product. A pricing schedule consists of a sequence of prices coupled with volume thresholds, namely a minimum number of units to be purchased to access the corresponding price. At each time t, the algorithm receives transaction data collected in $t-1$ and promptly computes a new pricing schedule modifying the current one.

4.1. Pricing without Volume Discounts

The estimation procedure for the demand function is summarized in Algorithm 2. For every single product, the algorithm takes in input the data of past order records, which include information on the time of the sale, the customer id, the number of units sold, and the price of the sale. The algorithm returns, for every time t, the average price of the sales p_t and the total volume v_t. Notice that the actual prices. collected and the list prices may differ due to promotions and discounts (see Section 4.3).

We use the above data to estimate the demand function specifying the volume for each value of price. In our estimation model, we also take into account seasonality and market trends.

In particular, our estimation algorithm is based on a Bayesian Linear Regression [45] (from now on, BLR). This class of algorithms allow the estimation of the uncertainty over the predictions, thus allowing the adoption of multi-armed bandit approaches to balance exploration and exploitation (see Section 4.2).

The input space is denoted with $\mathcal{T} \times \mathcal{P}$, representing the possible combinations of time and price. Instead, the output space is denoted with $\mathcal{V}$, representing the volume. Furthermore, we introduce two features spaces $\mathcal{U}$ and $\mathcal{D}$, corresponding to seasonality&trend and price. We define $\mathcal{J}$ and $\mathcal{K}$ as the sets containing the indices of time and price features, respectively.

In particular, we introduce the function $\chi : \mathcal{T} \to \mathcal{U} \subset \mathbb{R}^{|\mathcal{J}|}$ mapping a time t

into its seasonality&trend features. The transformation is as follows:

$$\chi : t \mapsto \underline{u}_t := \left[u_t^{(j)}\right]_{j\in\mathcal{J}} \in \mathcal{U}. \tag{4.1}$$

We represent all the features related to seasonality (*e.g.*, the number of weeks) in polar coordinates in order to ensure a consistent behaviour between the periods (*e.g.*, years), and then we apply the basis functions transformation.

Furthermore, we introduce the function $\xi : \mathcal{P} \to \mathcal{D} \subset \mathbb{R}^{|\mathcal{K}|}$ mapping a price p_t into its features. The transformation is as follows:

$$\xi : p_t \mapsto \underline{d}_t := \left[d_t^{(k)}\right]_{k\in\mathcal{K}} \in \mathcal{D}. \tag{4.2}$$

This set of features $\mathcal{K}$ is composed by transformations that are actually monotonically decreasing in order to model the inverse relation binding price and volumes. Thus, the predicted curve results to be monotonically decreasing w.r.t. the price if feature weights are forced to be non-negative.

Bayesian Regression allows the definition of prior distributions of probability over the weights. We force such distributions to be non-negative, constraining the support to be in $[0, +\infty)$ (such as Lognormal distribution, see Wilson et al. [52]). Time-related features, instead, do not require to set any constraint and their prior distributions have support over $\mathbb{R}$. In particular, we adopt the Normal distribution. Thus, we have:

$$\theta_j \sim \mathcal{N}(\mu_j, \sigma_j^2), \forall j \in \mathcal{J},$$

$$\theta_k \sim \mathcal{LN}(\mu_k, \sigma_k^2), \forall k \in \mathcal{K},$$

where $\mathcal{LN}(\mu, \sigma^2)$ represent the *Lognormal* distribution with mean μ and variance σ^2, and $\mathcal{N}(\mu, \sigma^2)$ represent the *Normal* distribution with mean μ and variance σ^2.

4.2. Exploration Strategy

The procedure addressing the exploration-exploitation dilemma is summarized in Algorithm 3, and shown in Figure 4.1.

In particular, we resort to Thompson Sampling (TS) [1]. By construction, a Bayesian model provides a probability distribution of the posteriors on the weights. Such a probability distribution provides a measure of the uncertainty over the estimates and can be effectively used to guide the exploration during the learning process. More precisely, Thompson Sampling randomly generates

Algorithm 2 Demand Function Estimation

```
Require: Initialized BLR model $\mathcal{M}$, set of times $\mathcal{T}$ s.t. $|\mathcal{T}| = T$, transaction
    data $\mathcal{D}_t$ for $t \in \mathcal{T}$
Ensure: Trained BLR model $\mathcal{M}$
for $t \in \mathcal{T}$ do
    Estimate $v_t$ and $p_t$ from $\mathcal{D}$
    $\underline{d}_t \leftarrow \xi(p_t)$
    $\underline{u}_t \leftarrow \chi(t)$
end for
$W \leftarrow [v_1, v_2, \cdots v_T]$
$D \leftarrow [\underline{d}_1, \underline{d}_2, \cdots \underline{d}_T]$
$U \leftarrow [\underline{u}_1, \underline{u}_2, \cdots \underline{u}_T]$
Train BLR model $\mathcal{M}$ using $(U, D)$ as features and $W$ as target.
```

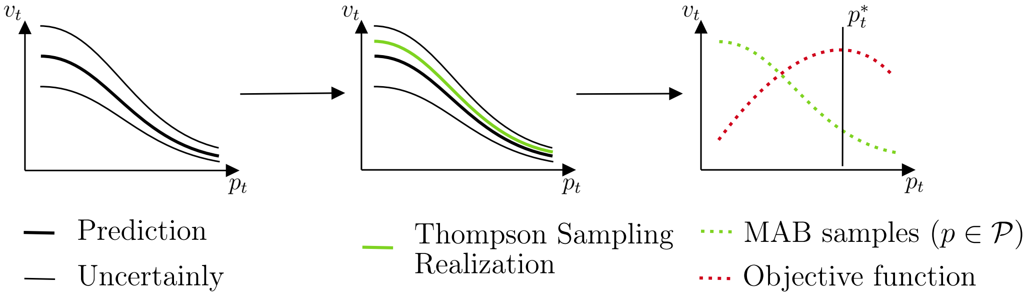


Figure 4.1: Optimal price p_t^* estimation process.

samples from the posterior distribution of the weights of BLR, retrieving in this way a realization of the posterior binding features from time and price to the volumes' curve. Now, given time h, fixing related features vector $\underline{u}_h$, we can evaluate volumes with respect to only to price values.

Consider a Multi-Armed Bandit approach in which we select the best arm over a finite set of possible prices $\mathcal{P}$. We can compute the value of the expected objective function $\hat{f}(p), \forall p \in \mathcal{P}$ as provided in Equation 3.2, and select the best arm corresponding to:

$$p_h^* = \underset{p \in \mathcal{P}}{\operatorname{argmax}} \left[\hat{f}(p) \right]_{p \in \mathcal{P}}, \tag{4.3}$$

where $\hat{f}(p)$ is the objective function estimated using volumes $\hat{v}_t$, the latter coming from Thompson Sampling over the model.

Algorithm 3 Exploration Strategy

```
Require: Trained BLR model $\mathcal{M}$, time $h$, set of prices $\mathcal{P}$
Ensure: Optimal price $p_h^*$ at time $h$.
  $\underline{u}_h \leftarrow \chi(h)$
  Sample $\tilde{\boldsymbol{\theta}}$ from $\mathcal{M}$ weights' distributions.
  Initialize BLR model $\widetilde{\mathcal{M}}$ using $\tilde{\boldsymbol{\theta}}$ as weights.
  $\left[\hat{f}(p)\right]_{p\in\mathcal{P}} \leftarrow \widetilde{\mathcal{M}}(\underline{u}_h, \mathcal{P})$
  $p_h^* \leftarrow \text{argmax}_{p\in\mathcal{P}}\left(\left[\hat{f}(p)\right]_{p\in\mathcal{P}}\right)$
```

4.3. Pricing with Volume Discounts

Let η be the desired number of volumes thresholds to propose along with as many different prices. In real-world scenarios, customers generate shopping baskets with different products in an arbitrary number of units each. We focus on a given product and the consider only the baskets containing at least 1 unit of it. Let β_z, with $z \in \mathbb{N}$, be the proportion of baskets containing the product with a volume of z. Average volume for the product is $\bar{V} = \sum_{i=1}^{\infty} \beta_i \cdot i$. Given the threshold ω_k, the total proportion of baskets inside that range is given by:

$$\bar{\beta}_k = \sum_{i=\omega_k}^{\omega_{(k+1)}-1} \beta_i. \tag{4.4}$$

The average volume inside the threshold is consequently defined as:

$$\bar{V}_k = \frac{\sum_{i=\omega_k}^{\omega_{(k+1)}-1} \beta_i \cdot i}{\sum_{i=\omega_k}^{\omega_{(k+1)}-1} \beta_i}. \tag{4.5}$$

Suppose a customer has a need of N units of the given product, this can be fulfilled across any number of time steps, buying each time a number of units (or *volume*) between ω_k and ω_{k+1} for some k. After the customer bought the product in any volume, he has a probability γ of coming back to the same retailer buying another batch of the same size. This kind of modelling of the user's behavior is particularly consistent with some kind of goods, *i.e.* consumer goods, food and beverages or any short lifespan good for which a customer is led to schedule periodical purchases. With probability $1 - \gamma$ the customer is acquired from competition and will not return the next time. Let $\bar{m}_1$ denote the desired margin when a single unit is purchased. It follows that expected margin $\bar{\mu}$ coming from a customer with a need of N units and who performs only single-unit orders is, exploiting the truncated geometric series identity:

$$\bar{\mu}_1 = \sum_{\tau=1}^{N} \gamma^{\tau-1}\bar{m}_1 = \frac{1-\gamma^N}{1-\gamma}\bar{m}_1. \tag{4.6}$$

A customer with the same need but whose orders contain a number of units between ω_k and ω_{k+1}, which are associated with a margin $\bar{m}_k$, will generate the following expected margin:

$$\bar{\mu}_k = \sum_{\tau=1}^{\left\lceil \frac{N}{\bar{V}_k} \right\rceil} \gamma^{\tau-1}\bar{m}_k\bar{V}_k = \frac{1-\gamma^{\left\lceil \frac{N}{\bar{V}_k} \right\rceil}}{1-\gamma}(1-\delta_k)\bar{m}_1\bar{V}_k, \tag{4.7}$$

where δ_k is the discount applied to the single-unit margin $\bar{m}_1$, namely:

$$m_k = m_1(1-\delta_k), \qquad k = 1, \dots, \eta. \tag{4.8}$$

By imposing $\bar{\mu}_k \geq \bar{\mu}_1$, we get:

$$\delta_k \leq 1 - \frac{1-\gamma^N}{\bar{V}_k\left(1-\gamma^{\left\lceil \frac{N}{\bar{V}_k} \right\rceil}\right)}. \tag{4.9}$$

Given the desired margin $m_t^* = p_t^* - c$ derived in the previous section, the total expected margin without any discount can be computed as $m_t^*\bar{V}$. Suppose we are applying a volume discount policy, we expect that it would not decrease the total expected margin given without. Unit-volume margin $\bar{m}_1$ can be computed by imposing that expected margin without any discount policy coincides with the one including them:

$$\sum_{k=1}^{\eta} \bar{V}_k\bar{m}_k = m_t^*\bar{V}. \tag{4.10}$$

Substituting Eq. 4.8 in Eq. 4.10, we get:

$$\bar{m}_1 = \frac{m_t^*\bar{V}}{\sum_{k=1}^{\eta}(1-\delta_k)\bar{V}_k}. \tag{4.11}$$

Finally, the margins $\bar{m}_1, \bar{m}_2, \dots, \bar{m}_\eta$ for the different volume thresholds are determined by $\bar{m}_k = \bar{m}_1(1-\delta_k)$, for $k = 1, \dots, \eta$.

The volume discount algorithm is summarized in Figure 4.2.

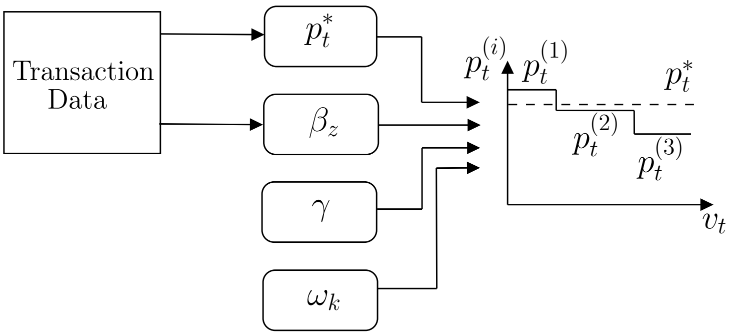


Figure 4.2: Volume discount estimation process.

5 | Experimental Evaluation

In this chapter, we experimentally evaluate our algorithms. In particular, we evaluate them both in a simulated environment and in a real-world scenario. In the first, the main focus of our evaluation is on the pricing to check whether it successfully converges to the optimal price faster than a non-shape-constrained variant. In the latter, the focus of the evaluation is on the business value of the whole algorithm, quantified during an experiment lasted about 4 months on a large Italian e-commerce.

5.1. Evaluation in Synthetic Environment

We compare the dynamic pricing algorithm we propose with the non-shape-constrained version having a Normal prior instead of a Lognormal one. Both algorithms have their performances benchmarked by a *clairvoyant* algorithm, namely a policy choosing at every time the optimal arm and able to achieve its expected reward. In particular, we compare the two algorithms (shape-constrained exploration and free-shape) over settings with noisy data, outliers and nonstationarity in the customers' demand curve.

5.1.1. Robustness to Noise and Outliers

One of the main advantages that we expect to gain using a shape-constrained algorithm is a better robustness to stochastic perturbations of the environment. Moreover, the success of dynamic pricing algorithms is often doomed by the presence of outliers. Customers performing very large orders, or weeks with particularly low sales due to external exogenous factors (*i.e.* a competitor's action) are not uncommon in practice.

Our goal is to compare the performances of the two versions of the algorithm varying the standard deviation of a zero-mean Gaussian noise which is additive with respect to the true, hidden demand function used in the synthetic simulation jointly modifying the percentage of outliers generated among the synthetic data. Outlier generation is implemented as follows: with a probability that is given as a simulation hyper-parameter, in some timesteps a zero-mean Gaussian noise which has a noise equal to 10 times the base one.

We test the two algorithms in the very same scenario, using the same random seed and the same demand curve generating function: in the simulation we assume the base demand (seasonality and trend) to be fixed and known and we focus on the hidden demand curve learning, reported in Fig. 5.1.

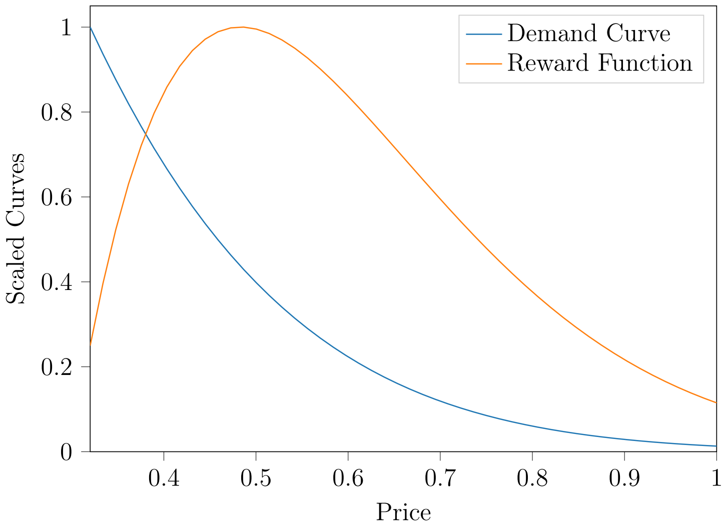


Figure 5.1: Demand curve used in the simulation and corresponding reward function obtained using $\lambda = 0$.

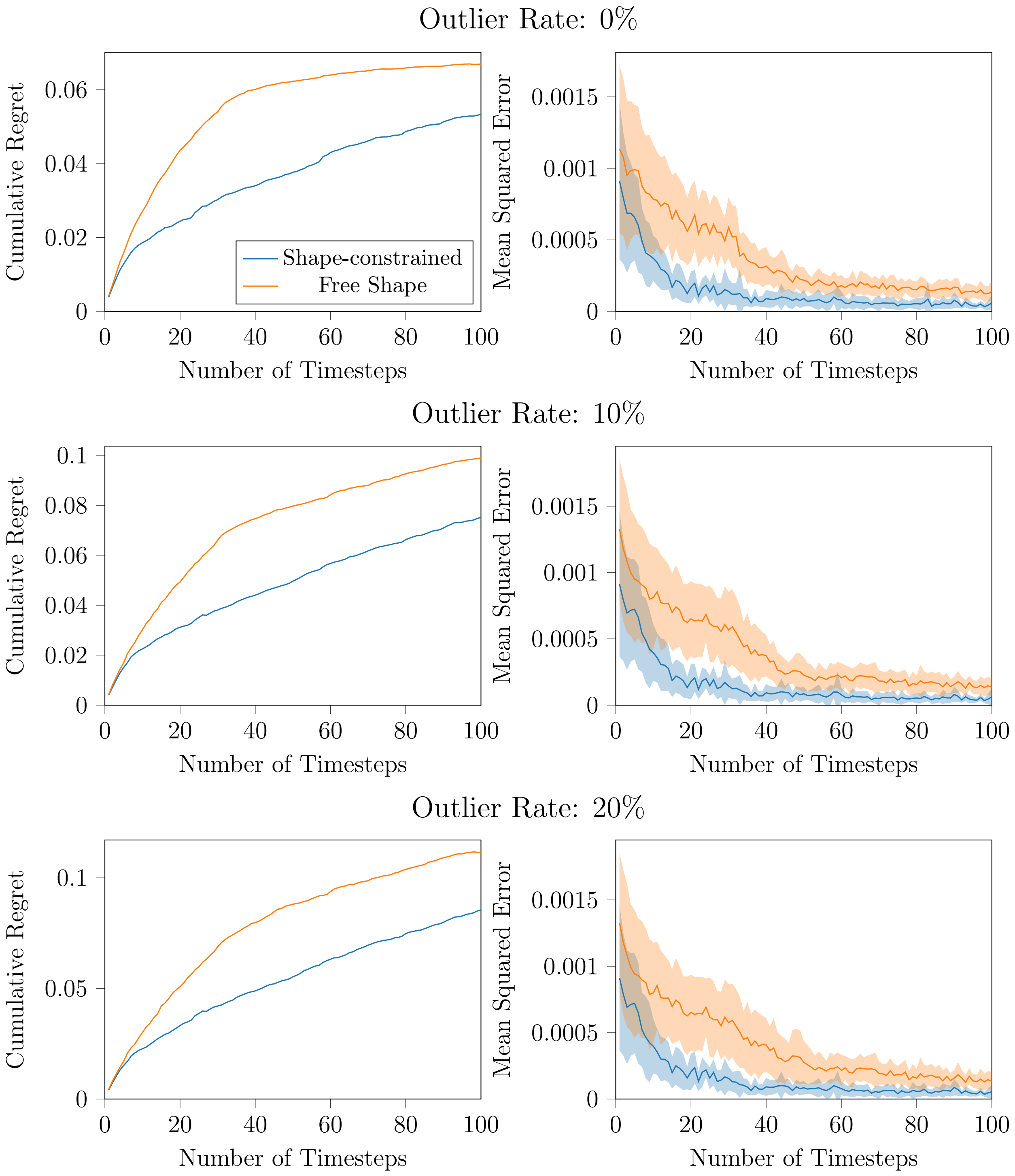
The equation generating the curve in Fig. 5.1 is $f(x) = 2e^{-(x+1.2)^{\frac{5}{2}}}$ which results, setting $\lambda = 0$, in the reward function $f(x) = 2(x - 0.3)e^{-(x+1.2)^{\frac{5}{2}}}$.

We perform a simulation for each combination of noise value and % of outliers among the generated data. In particular, the grid is generated by noises in $\{0.001, 0.005, 0.01\}$ and outlier % in $\{0, 10, 20\}$. This procedure is iterated for 15 times and the results correspond to the average.

The demand curve is expanded in a Bernstein Polynomial Basis with $M = 75$, then it is fed to a BLR model using either lognormal or normal priors on the parameters. We tuned the model hyper-parameters, obtaining in particular a standard deviation of 0.75 for the lognormal prior and 2.0 for the normal one. The algorithm runs for 100 timesteps, starting from no samples. At each timestep, BLR is fitted over the available data and using Thompson Sampling over its posterior distribution an estimation of the demand curve is obtained: then the next price is set as the one maximizing the corresponding estimated reward function. Thompson Sampling consider 50 arms corresponding to linearly spaced prices across the price domain $[0.32, 1]$, where the cost is 0.3. After setting the new price, the next sample will be generated from the true function's evaluation in that value and collected by the algorithm as a new data.

In Fig. 5.2 and 5.3 we report cumulative regret plots (on the left) and the instantaneous Mean Squared Error of the demand curve fit (on the right) for two different noise values and three different outlier % values. We also report the standard deviation of MSE to quantify its aleatoric uncertainty across the 15 simulations.

The two algorithms' curves have been reported within the same plot and in the same scale.



Outlier Rate: 0%

Outlier Rate: 10%

Outlier Rate: 20%

Figure 5.2: Cumulative Regrets and MSE, Noise: 0.001

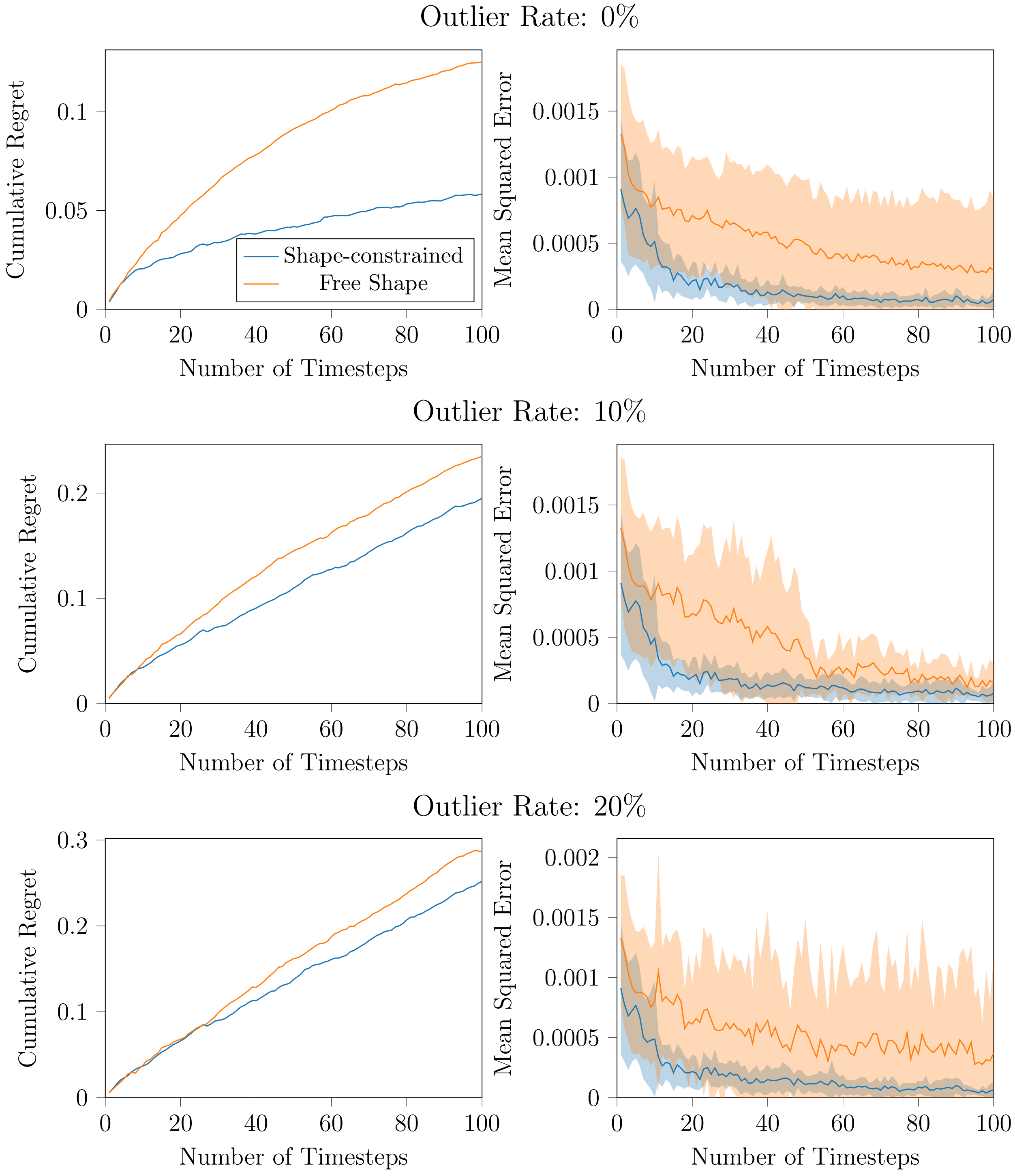


Figure 5.3: Cumulative Regrets and MSE, Noise: 0.005

Shape-constrained BLR outperforms the free-shape model (that uses normal priors) in each scenario, both in cumulative regret and fit error over time. As well as being better performing, shape-constrained model is also more stable than the other one, as we can observe from the standard deviations of MSEs: not only the error's decrease over time is steeper in the first model, but is also less uncertain. This particular behavior suggests that the monotonic formulation is more robust to noise and outlier.

In this noisy setting is interesting to observe how increase noise and outliers' % results in an increase of the total regret, as we can see from Table 5.1 and Table 5.2. The two tables quantify how much less regret a shape-constrained model is able to achieve w.r.t the free-shape one.

Noise	**Outlier Proportion**		
	0%	**10%**	**20%**
0.001	0.053327	0.075171	0.085442
0.005	0.058318	0.195081	0.251743
0.01	0.092141	0.476388	0.544845

Table 5.1: Total regrets of shape-constrained model varying noise and outlier proportion among simulations.

Noise	**Outlier Proportion**		
	0%	**10%**	**20%**
0.001	0.066960	0.098937	0.111343
0.005	0.125297	0.234983	0.286624
0.01	0.204927	0.505995	0.550301

Table 5.2: Total regrets of free-shape model varying noise and outlier proportion among simulations.

5.1.2. Robustness to Nonstationarity

In many real-world scenarios the intrinsic nonstationarity of the customers' demand may lead many algorithms to perform sub-optimally. Imagine that a given customer has its own demand curve, with a particular shape and magnitude. While we can deal with the magnitude's estimation through market's seasonality or trend as in the previous simulations, in this experiment we assume that the shape may change over time. A demand curve changing over time, usually in an abrupt way due to events conditioning the market (*i.e.* the entry of a competitor or a new product), is usually faced using a *sliding window* approach [49]. In this experiment, we compare the two models (and the clairvoyant) by varying the number of abrupt changes in the demand curve and the size of the sliding window.

The two algorithms are tested in the same setting and the goal is the minimization of the regret when the goal is to maximize the net cash flow margin (*i.e.* $\lambda = 0$ in Eq. 3.2). The hidden demand curves are represented in Fig. 5.4:

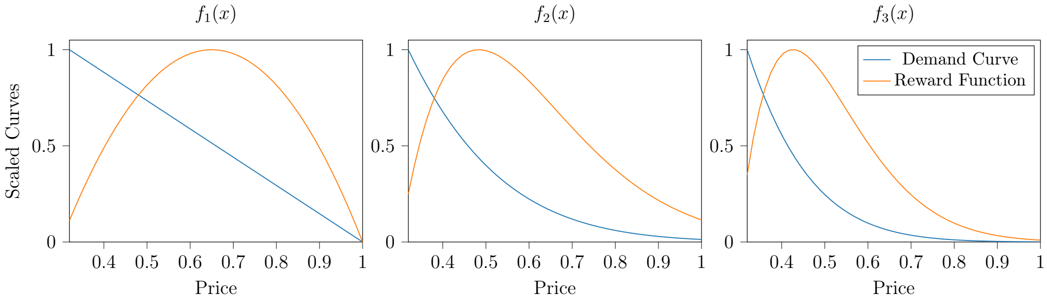


Figure 5.4: Demand curves used in the simulation and corresponding reward function obtained using $\lambda = 0$.

Equations for demand curves in Fig. 5.4 are, from left to right:

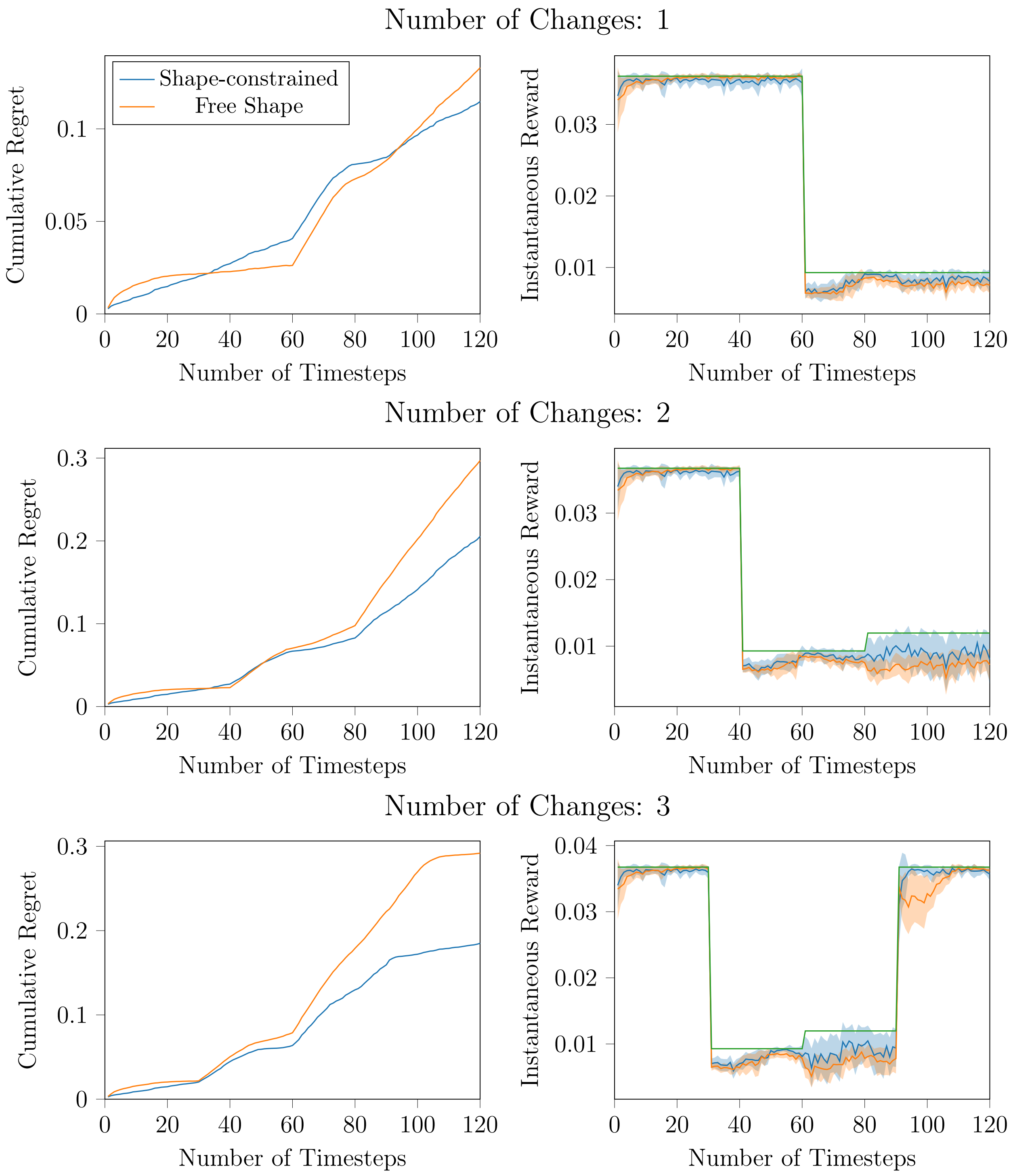
$$f_1(x) = \frac{3}{10}(1-x), \quad f_2(x) = 2e^{-(x+1.2)^{\frac{5}{2}}}, \quad f_3(x) = 7e^{-(x+1.2)^3}.$$

We consider the scenario in which the number of abrupt changes may vary in $1, 2, 3$ and the size of the sliding window in $\{20, 30, 40\}$. In particular, the order in which the demand curves changes is the same as in Fig. 5.4, repeating $f_1(x)$ both as the first and the last function to appear in case of 3 abrupt changes. For example: if the number of changes is 1, them after half of the timesteps the function will switch from $f_1(x)$ to $f_2(x)$, while if it changes 3 times it will follow the sequence $f_1(x)$, $f_2(x)$, $f_3(x)$, $f_1(x)$. Noise is fixed to 0.001 and there are outlier's probability to 0.

Model design is identical to the previous simulations: a Bernstein Polynomial Basis of order 75 is used and standard deviations of lognormal priors and normal priors are 0.75 and 2.0 respectively. Algorithm runs for 120 timesteps in order to ensure that for each number of abrupt changes the demand curves persists for the same number of timesteps. Arms, price domain and Thompson Sampling procedure are the same as before.

The procedure is iterated 15 times and the result is taken on average.

In Fig. 5.5 and Fig. 5.6 we report cumulative regret plots (on the left) and the instantaneous reward collected by the agent at each timestep along with their 1-standard deviation confidence intervals to quantify aleatoric uncertainty, also clairvoyant's reward is reported. Curves in the same plot are all in the same scale and we report the ones concerning 20-samples and 40-samples large window sizes varying the number of abrupt changes in demand curve.



Number of Changes: 1

Number of Changes: 2

Number of Changes: 3

Figure 5.5: Cumulative Regrets and Intantaneous Rewards, Sliding Window Size: 20.

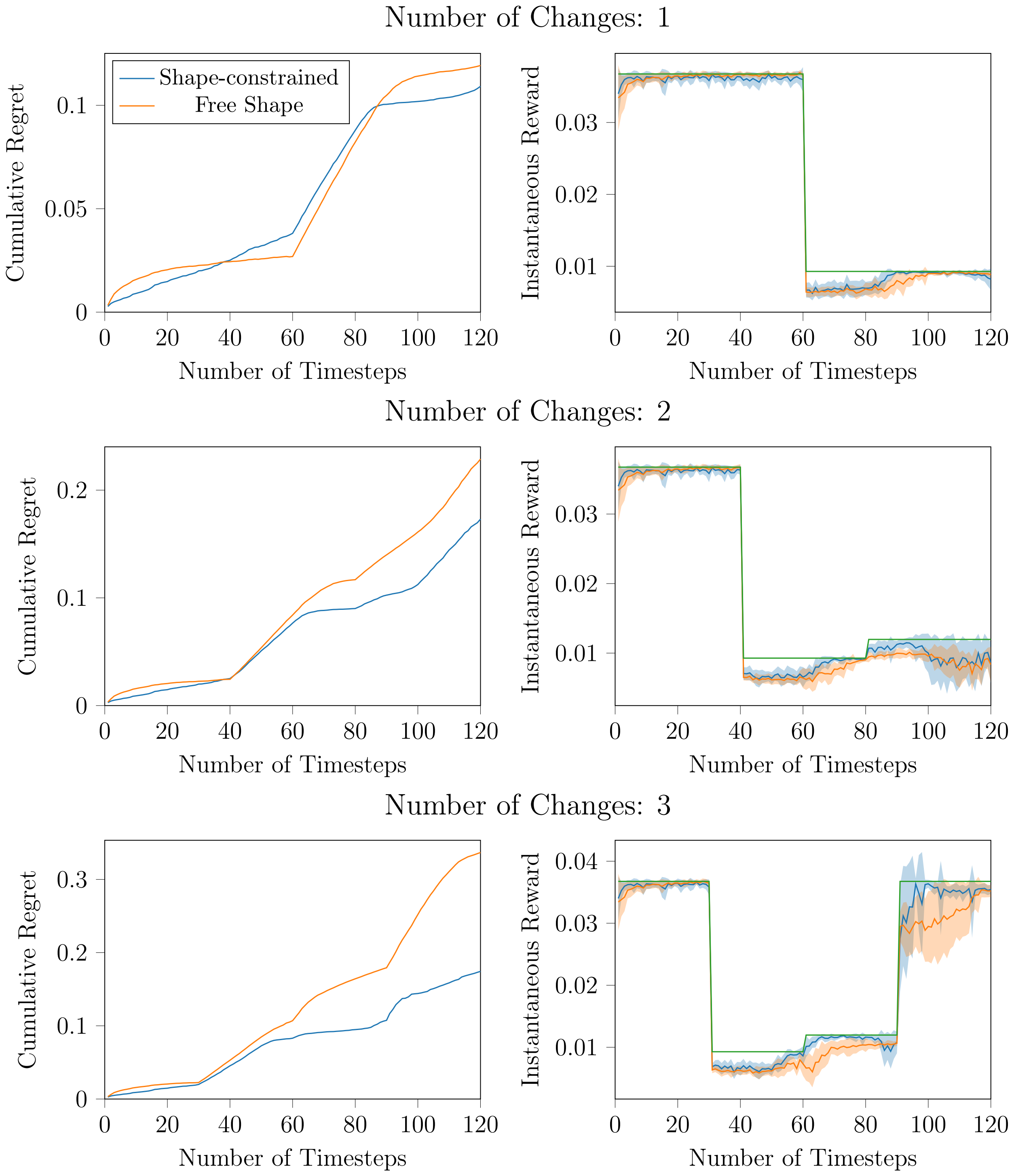


Figure 5.6: Cumulative Regrets and Instantaneous Rewards, Sliding Window Size: 40.

Shape constrained model always outperforms the free-shape one in terms of total regret (see Table 5.3 and Table 5.4). Is it interesting to see how the two mantain similar performances (or even better for normal priors model) until a certain point, where the two divides. The point in which the two cumulative regret curves fall apart seems to come sooner when the number of abrupt changes is greater, suggesting a worst nonstationarity-handling coming from the free-shape model.

The total regrets reported in Table 5.3 and Table 5.4 suggests that in scenarios in which the abrupt change is only one or two a larger window size is better performing, while if the environment is more nonstationary (*i.e.* 3 demand curve changes) a smaller window can achieve better results. Having more data samples allow BLR to better fit the true demand curve, but having samples generated by another process (usually the older samples) can derange it. The window size hyper-parameter allow us to control this trade-off.

Window Size	**Demand Curve Changes**		
	1	**2**	**3**
20	0.114751	0.205427	0.184926
30	0.125555	0.220044	0.193747
40	0.109177	0.173122	0.174389

Table 5.3: Total regrets of shape-constrained model varying window size and number of demand curve changes among simulations.

Window Size	**Demand Curve Changes**		
	1	**2**	**3**
20	0.133005	0.297048	0.291925
30	0.133695	0.288935	0.362537
40	0.119264	0.228905	0.336763

Table 5.4: Total regrets of free-shape model varying window size and number of demand curve changes among simulations.

5.2. Evaluation in Real-World Environment

We put our algorithm into production on an italian e-commerce selling consumer goods (non-Giffen). In this real-world experiment, we optimize the prices in presence of volume discounts, computing a full pricing schedule. Focus will be on business value unlocked by an optimal pricing proposal and on the dynamics generated by volume discounts on customer demand.

5.2.1. Experimental Setting

We perform a real-world experiment by optimizing the price of an e-commerce adopting a long-tail economic model [3]. The focus of the experimental campaign is on high volumes products, over which we can reasonably apply also volume discounts.

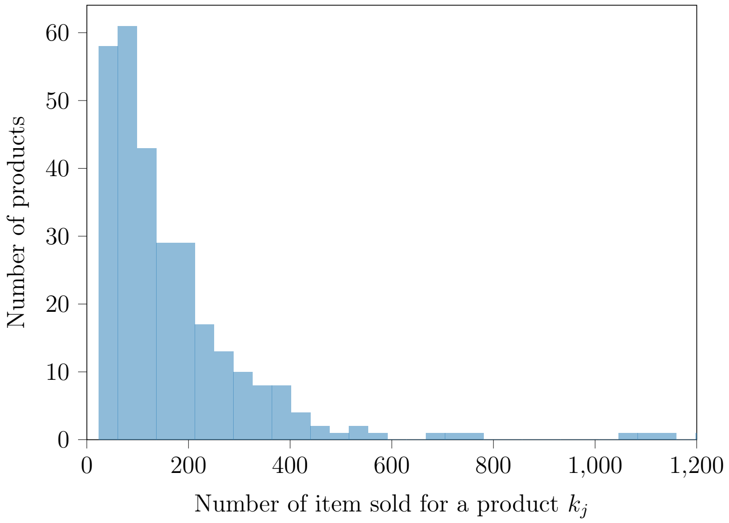


Figure 5.7: Distribution of the volumes of the products in the test set over the previous period of 2021.

To evaluate the algorithms performance we perform an online A/B test campaign.

The experimental campaign is conducted in one of the main category of the e-commerce, with a test set (A) composed by $N_t = 295$ products and a control set (B) composed by $N_c = 33$ products of the same category with the same characteristics. The test and the control sets have been defined by e-commerce specialists according both technical and market aspects.

The test is about products with a turnover of 300 KEuros and a net margin of 83 KEuros. Volumes of the products in the test set is shown in Figure 5.7.

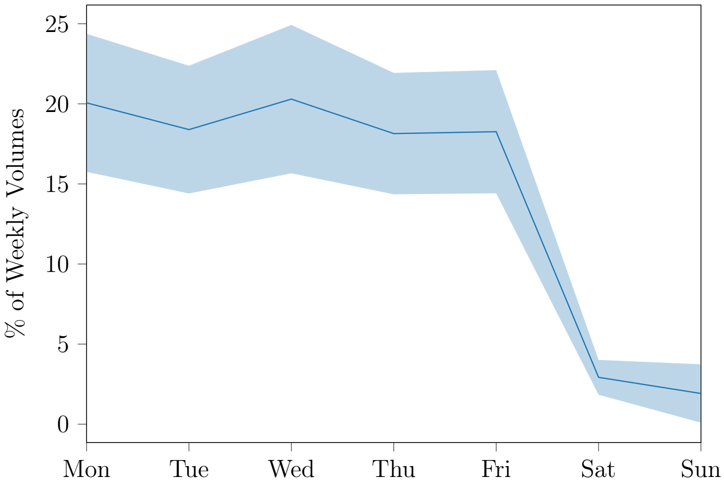


Figure 5.8: Seasonality over a single week.

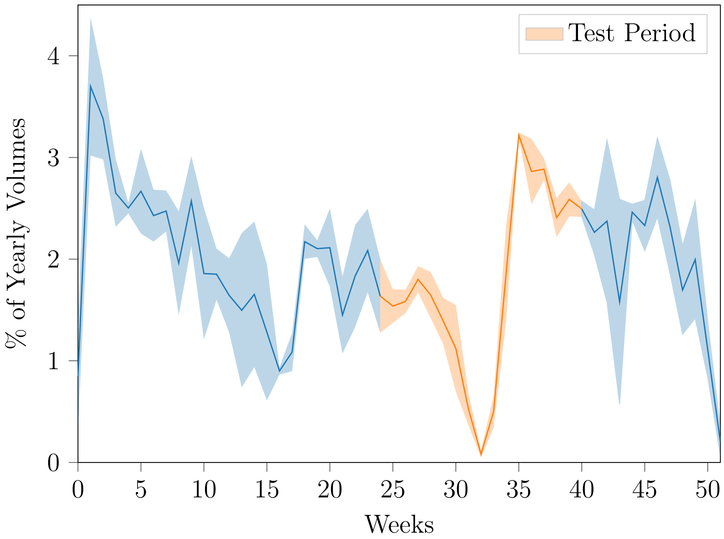


Figure 5.9: Seasonality over different weeks in an year.

We update the prices every 7 days, since a significant and stable intra-week seasonality has been observed (see Figure 5.8). This kind of e-commerce is subject to a significant seasonality over the different periods of the years, as shown in Figure 5.9.

Due to the particular kind of goods sold by the e-commerce and the nature of target customer segment, volume discounts are a really important business component that affects customers' loyalty and the logistic organization of the company. E-commerce specialists asked us to impose $\eta = 3$ volume's thresholds to each product of the test set, and optimize the corresponding proposed discounts.

The test lasted for 17 weeks, from 16 June 2021 to 17 October 2021 and there are no impacting factors that can influence the performance of the test set (A) w.r.t. control set (B) and vice-versa (*i.e.* variations in advertising expenditures).

On the Monopolistic Assumption

Given the availability of competitors' prices in the period from 13 July 2020 to 16 June 2021 (start of the test), we perform a statistical testing to assess the correlation of that information with the performances of the e-commerce under analysis. More in detail, we compared the latter's volumes with the price difference found with competitors. We selected the 35 products from the test set for which competitor's data were available, then we performed a Spearman's rank correlation test (for a detailed explanation see [29]) comparing their volumes with the difference in price of the same product sold by each of the 3 main competitors of the market. In Figure 5.10 are represented the *p-values* of correlation test for each product against each competitor: the null hypothesis of the test is "The two variables are uncorrelated", we accept the null hypothesis every time since p-values are always bigger than our threshold set to 0.05, in order to achieve a 95% confidence. Thus, being this quantities statistically uncorrelated, we consider our monopolistic assumption valid.

Performance Metric

The business goal is to maximize the net margin (*i.e.* $\lambda = 0$ in Equation (3.2)).

To assess the goodness of the choice of the products' set w.r.t. this performance metric, we performed a statistical test to check if the two groups are comparable. For each product, we computed the average weekly net margins during the first six months of 2021: we were interested to see if the median and the mean of this value across the products of the test set are higher than the ones of the control set. We performed one-sided permutation tests with the null-hypothesis being "The test set has not a higher median/mean of net margin w.r.t. the control set": Figure 5.11 represents the distributions of the tests' statistics together with the observed one, the resulting p-values concerning medians and means are respectively 0.54 and 0.45, resulting in a null-hypothesis acceptance. Thus, we can conclude that the test population has not a higher median/mean of the chosen performance metric at the start of the test.

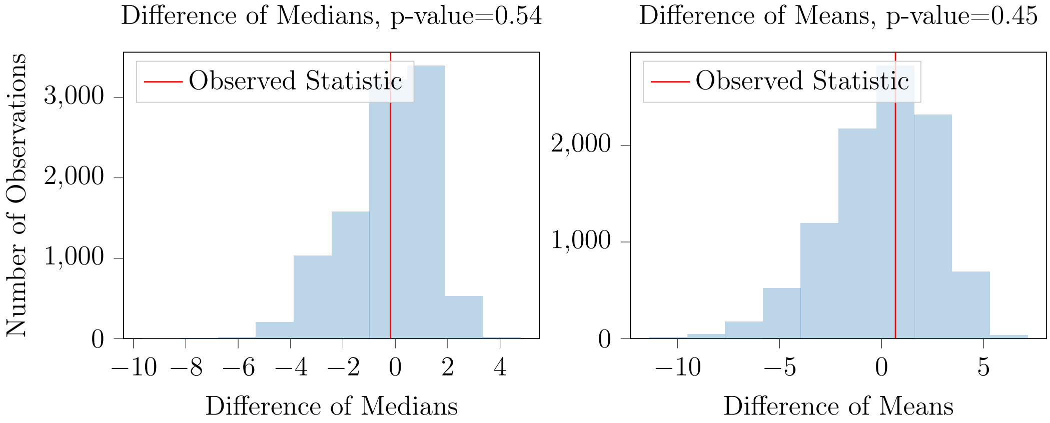


Figure 5.11: Distribution of the two-sided permutation tests statistics before the test, $R = 10000$ random permutations.

5.2.2. Experimental Results

Performance Overview The goods priced by PSV-B performed an improvement in the performance metric of +55% w.r.t the control set of goods. After 17 weeks, we performed the same statistical test on the weekly performance metric between the two sets of products during the test period. In Figure 5.12 are represented the distribution of the the test's statistics, difference of medians and difference of means, along with the observed ones.

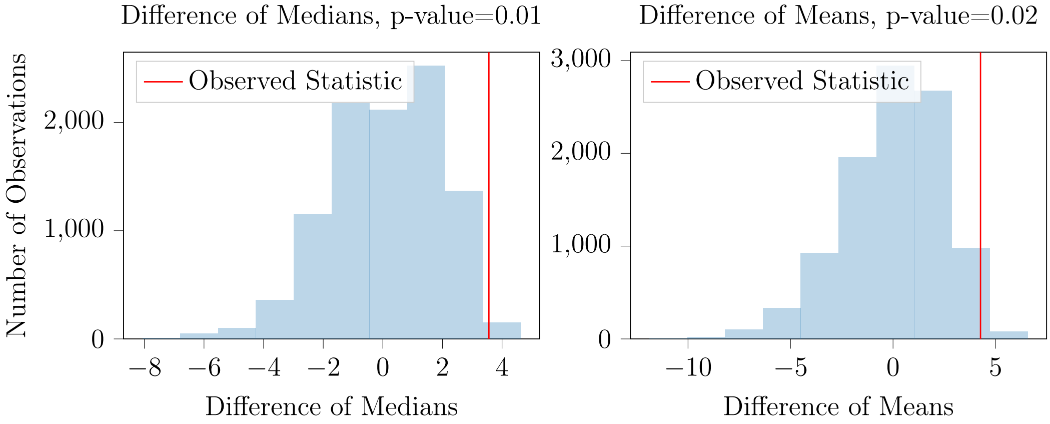


Figure 5.12: Distribution of the two-sided permutation tests statistics after the test, $R = 10000$ random permutations.

The two tests, performed with the same seed and number of random permutations of the previous, yielded this time p-values on the medians and on the means of respectively 0.01 and 0.02, allowing us to reject the null hypothesis and conclude that the test set of products has both a higher median and a higher mean of the average weekly performance metric with respect to the control set of products, with a confidence of $> 97.5\%$.

Looking at the performances on a product-wise level, we report in Figure 5.13 the sorted % of improvement on the performance metric w.r.t. to the period of 2021 preceding the test, for each single product.

In the test set, 138 products over 295 ($\sim$ 47%) improved their average performance with respect to the previous period of 2021, while in the control set only 8 products over 33 ($\sim$ 25%) did so.

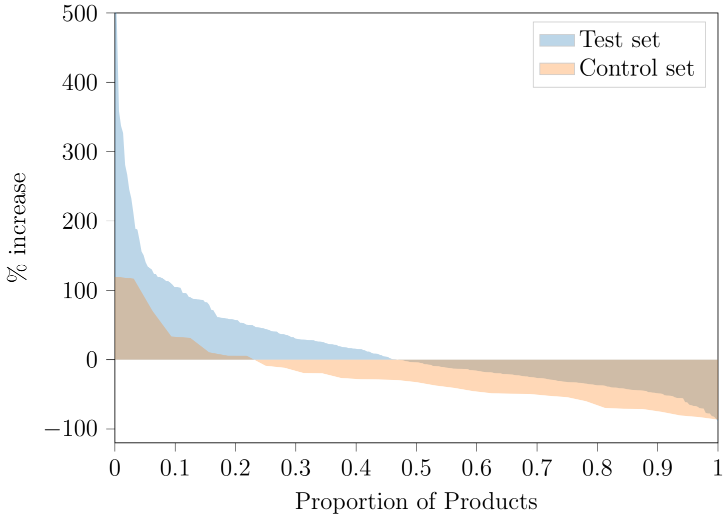


Figure 5.13: Margin Improvement over products.

Basis Functions To perform the test mentioned above, we carefully design the BLR with a particular focus on the basis functions' choices. In order to grasp the irregular nature of the e-commerce's seasonality, we choose radial basis functions (*RBF*). Trend is modelled by choosing polynomial basis functions. RBF are evaluated with different shifts and scales, while polynomial features with different degrees.

Effect of Volume Discounts The goal of the volume discounts algorithm is to modify the probability distribution of the units count of the same product in a basket. In other words, we want to alter $\bar{\beta}_k$: since $k \in \{1, 2, 3\}$, the goal is to move mass from $\bar{\beta}_1$ to $\bar{\beta}_2$ or $\bar{\beta}_3$.

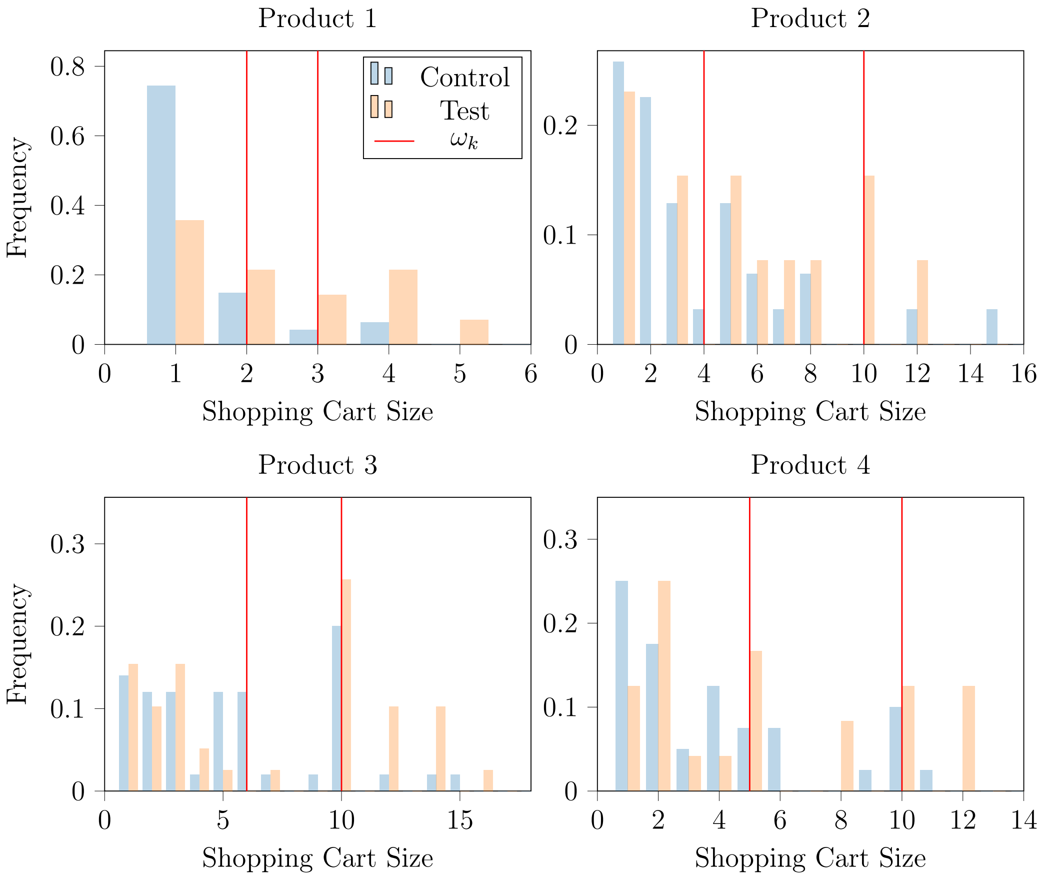


Figure 5.14: β distribution over the 4 representative products.

In Figure 5.14 are represented the distributions of β for 4 most representative products of the category in terms of total yearly revenue. There we can distinguish two different behaviors between the test period and the previous one. In Table 5.6 are reported the variations of the three $\bar{\beta}_k$: during the test we achieved an improvement of $\bar{\beta}_2$ and $\bar{\beta}_3$ at the expense of $\bar{\beta}_1$.

Product	$\Delta\bar{\beta}_1$	$\Delta\bar{\beta}_2$	$\Delta\bar{\beta}_3$
1	-32%	+10%	+22%
2	-26%	+25%	+1%
3	-15%	+4%	+11%
4	-5%	+1%	+4%
Mean	-19.5%	+10%	+9.5%

Table 5.5: Variations of $\bar{\beta}_k$ after test.

In Table 5.6 are reported the variations in the average units per basket of the

4 products after the test. As we can see, the volume discounts not only modify cart distribution, but also increase the number of units purchased per time.

Product 1	Product 2	Product 3	Product 4	**Mean**
+63%	+43%	+11%	+14%	+33%

Table 5.6: Variation of units per basket after test.

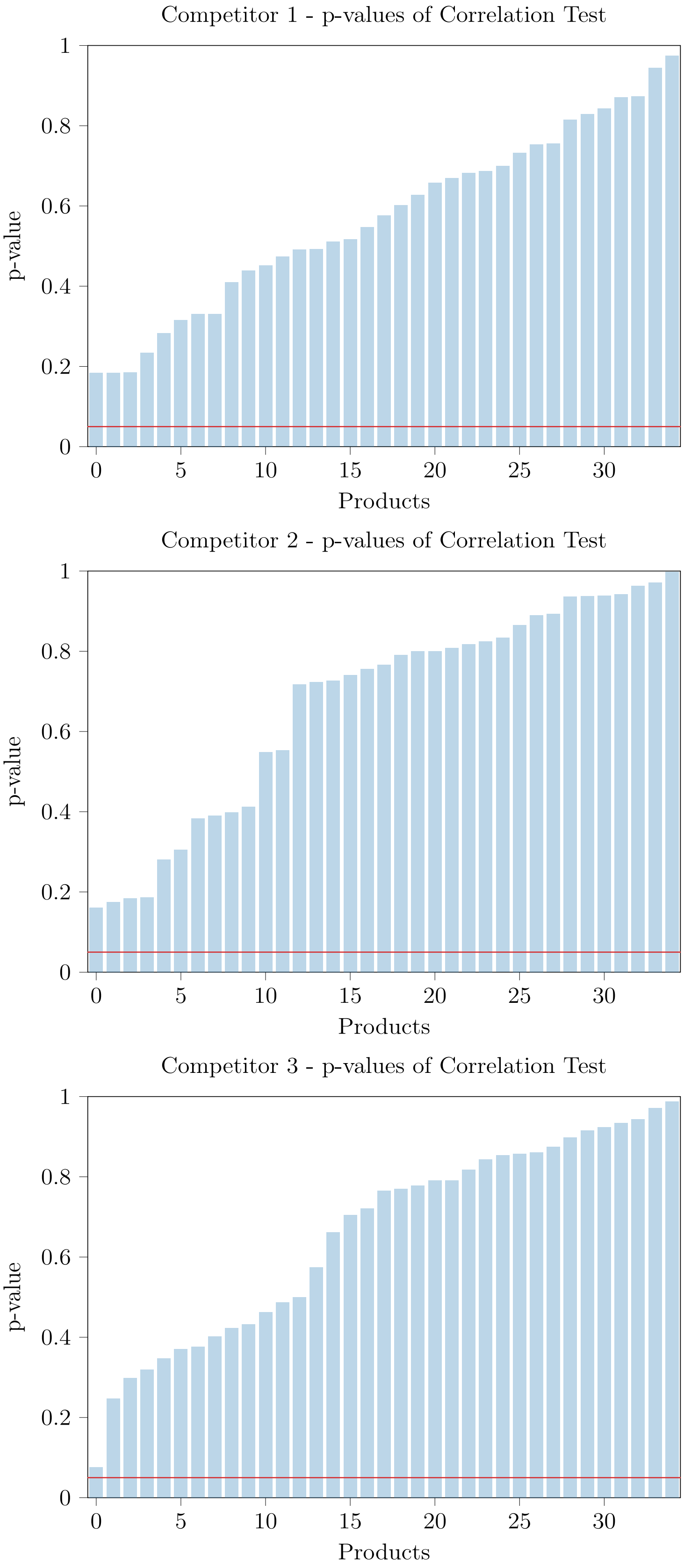


Figure 5.10: Distribution of the p-values of Spearman's rank correlation test.

6 | Conclusions and Future Works

6.1. Conclusions

In this work, we introduce a novel dynamic pricing algorithm to deal with typical real-world scenarios. In particular, the main features distinguishing our algorithm, namely PSV-B, from most of those available in the literature are as follows.

- The possibility to compute a pricing schedule integrating a data-driven volume discounts policy.
- A way to deal with seasonality that is built-in in the model and requires no assumptions nor additional hyper-parameters.
- An extensive evaluation test performed both in a simulated environment and in a real-world context.

We validate the design choices of our solution in a synthetic environment, then a real-world experimental campaign has been conducted to assess the added economic value that our algorithm can provide.

In each of the simulations we conducted, we observed a clear evidence in favor of the shape-constrained model: our design choice revealed itself as an important advantage in many situations (*i.e.* nonstationary demand or noisy data):

- Shape-constrained model achieved a better total regret in every simulation setting.
- In noisy environments, the shape-constrained model is more robust in the task of learning the true, hidden demand curve, resulting in a more stable estimation over time.
- In nonstationary environments, the shape-constrained model is more reactive to demand curve's abrupt changes.

We performed an online A/B test on an e-commerce selling consumer goods. The algorithm priced products for a total turnover of 300 KEuros (set A) and improved by 55% the net cash flow margin with respect to the set B. Additionally, the data-driven volume discounts policy successfully impacted customers' shopping behaviors in most of the products. We reported the example of 4 important products having their average units per shopping cart increased by +33% with respect to the previous period of time.

6.2. Future Works

An important research direction in dynamic pricing literature concerns the joint pricing of a large number of products possibly interacting with each other. Recent works, such as Mueller et al. [34], highlight how sales of a given product may depend on seasonality, its own price, and even the pricing of the other products. An extension to this work would be to take into account products' interactions and do a joint pricing on subgroups of products. Such an approach would avoid the so-called *cannibalization* phenomena among products, namely the scenario in which a product absorbs sales from another reducing the retailer's business. In other fields, such as advertising [37], the problem of dealing with different markets' interdependencies.

Another task, which is particularly crucial for companies and in particular the ones operating on the web (*i.e.* e-commerce, marketplaces) is to deal with pricing and advertising jointly. Over the last years, AI based algorithms for online advertising campaign optimization have been proposed and extensively analyzed (see [53] for an overview). Many of them rely on bandit approaches and online optimization techniques, being somehow close to pricing in their assumptions and methodologies [36, 38].

The aforementioned extensions to our model can be in principle made by feeding new features to the BLR, modelling the relationship between other products' prices or advertising expenses and sales of a given product.

Bibliography

[1] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1. JMLR Workshop and Conference Proceedings, 2012.

[2] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for thompson sampling. In *Artificial intelligence and statistics*, pages 99–107. PMLR, 2013.

[3] Chris Anderson. *The long tail: Why the future of business is selling less of more.* Hachette Books, 2006.

[4] Victor F Araman and René Caldentey. Dynamic pricing for nonperishable products with demand learning. *Operations research*, 57(5):1169–1188, 2009.

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

[6] Josef Bauer and Dietmar Jannach. Optimal pricing in e-commerce based on sparse and noisy data. *Decision Support Systems*, 106:53–63, 2018.

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

[8] Omar Besbes and Assaf Zeevi. On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science*, 61(4):723–739, 2015.

[9] Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[10] Christopher M Bishop. Pattern recognition. *Machine learning*, 128(9), 2006.

[11] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

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

[13] Jinzhi Bu, David Simchi-Levi, and Yunzong Xu. Online pricing with offline data: Phase transition and inverse square law. In *International Conference on Machine Learning*, pages 1202–1210. PMLR, 2020.

[14] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

[15] Ping Cao, Nenggui Zhao, and Jie Wu. Dynamic pricing with bayesian demand learning and reference price effect. *European Journal of Operational Research*, 279(2):540–556, 2019.

[16] Michael Chui, James Manyika, Mehdi Miremadi, Nicolaus Henke, Rita Chung, Pieter Nel, and Sankalp Malhotra. Notes from the ai frontier: Insights from hundreds of use cases. *McKinsey Global Institute*, 2018.

[17] Eric Cope. Bayesian strategies for dynamic pricing in e-commerce. *Naval Research Logistics (NRL)*, 54(3):265–281, 2007.

[18] Keith Cowling and Michael Waterson. Price-cost margins and market structure. *Economica*, 43(171):267–274, 1976.

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

[20] William R Dougan. Giffen goods and the law of demand. *Journal of Political Economy*, 90(4):809–815, 1982.

[21] Guillermo Gallego and Ming Hu. Dynamic pricing of perishable assets under competition. *Management Science*, 60(5):1241–1259, 2014.

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

[23] Olli-Pekka Hilmola. Quantity discount algorithm in an e-commerce environment. In *Proceedings of International Conference on Communication and Computational Technologies*, pages 437–446. Springer, 2021.

[24] Adel Javanmard. Perishability of data: dynamic pricing under varying-coefficient models. *The Journal of Machine Learning Research*, 18(1): 1714–1744, 2017.

[25] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *International conference on algorithmic learning theory*, pages 199–213. Springer, 2012.

[26] Simon Kemp. Perceiving luxury and necessity. *Journal of economic psychology*, 19(5):591–606, 1998.

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

[28] Robert Kleinberg and Tom Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 594–605. IEEE, 2003.

[29] Stephen Kokoska and Daniel Zwillinger. *CRC standard probability and statistics tables and formulae*. Crc Press, 2000.

[30] Tatsiana Levina, Yuri Levin, Jeff McGill, and Mikhail Nediak. Dynamic pricing with online learning and strategic consumers: an application of the aggregating algorithm. *Operations research*, 57(2):327–341, 2009.

[31] S McKay Curtis and Sujit K Ghosh. A variable selection approach to monotonic regression with bernstein polynomials. *Journal of Applied Statistics*, 38(5):961–976, 2011.

[32] Kanishka Misra, Eric M Schwartz, and Jacob Abernethy. Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*, 38(2):226–252, 2019.

[33] James P Monahan. A quantity discount pricing model to increase vendor profits. *Management science*, 30(6):720–726, 1984.

[34] Jonas Mueller, Vasilis Syrgkanis, and Matt Taddy. Low-rank bandit methods for high-dimensional dynamic pricing. *arXiv preprint arXiv:1801.10242*, 2018.

[35] Mila Nambiar, David Simchi-Levi, and He Wang. Dynamic learning and pricing with model misspecification. *Management Science*, 65(11):4980–5000, 2019.

[36] Alessandro Nuara, Francesco Trovo, Nicola Gatti, and Marcello Restelli. A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[37] Alessandro Nuara, Francesco Trovò, Dominic Crippa, Nicola Gatti, and Marcello Restelli. Driving exploration by maximum distribution in gaussian process bandits. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020*, pages 948–956. International Foundation for Autonomous Agents and Multiagent Systems, 2020.

[38] Alessandro Nuara, Francesco Trovò, Nicola Gatti, and Marcello Restelli. Online joint bid/daily budget optimization of internet advertising campaigns. *arXiv preprint arXiv:2003.01452*, 2020.

[39] Stefano Paladino, Francesco Trovò, Marcello Restelli, and Nicola Gatti. Unimodal thompson sampling for graph-structured arms. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 2457–2463, 2017.

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

[41] Paul A Rubin and WC Benton. A generalized framework for quantity discount pricing schedules. *Decision Sciences*, 34(1):173–188, 2003.

[42] Amir A Sadrian and Yong S Yoon. Business volume discount: A new perspective on discount pricing strategy. *International Journal of Purchasing and Materials Management*, 28(2):43–46, 1992.

[43] Weiwei Shen, Jun Wang, Yu-Gang Jiang, and Hongyuan Zha. Portfolio choices with orthogonal bandit learning. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[44] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[45] Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

[46] Francesco Trovo, Stefano Paladino, Marcello Restelli, and Nicola Gatti. Multi-armed bandit for pricing. In *12th European Workshop on Reinforcement Learning*, pages 1–9, 2015.

[47] Francesco Trovò, Stefano Paladino, Paolo Simone, Marcello Restelli, and Nicola Gatti. Risk-averse trees for learning from logged bandit feedback. In *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*, pages 976–983, 2017.

[48] Francesco Trovò, Stefano Paladino, Marcello Restelli, and Nicola Gatti. Improving multi-armed bandit algorithms in online pricing settings. *International Journal of Approximate Reasoning*, 98:196–235, 2018.

[49] Francesco Trovò, Marcello Restelli, and Nicola Gatti. Sliding-window thompson sampling for non-stationary settings. *J. Artif. Intell. Res.*, 68: 311–364, 2020.

[50] Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statis-*

tical science: a review journal of the Institute of Mathematical Statistics, 30(2):199, 2015.

[51] Yining Wang, Boxiao Chen, and David Simchi-Levi. Multimodal dynamic pricing. *Management Science*, 2021.

[52] Ander Wilson, Jessica Tryner, Christian L'Orange, and John Volckens. Bayesian nonparametric monotone regression. *Environmetrics*, 31(8): e2642, 2020.

[53] Yong Yuan, Feiyue Wang, Juanjuan Li, and Rui Qin. A survey on real time bidding advertising. In *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*, pages 418–423. IEEE, 2014.