

Dynamic Pricing with Volume Discounts in Online Settings

Marco Mussi^{1*}, Gianmarco Genalti^{1*}, Alessandro Nuara²,
Francesco Trovó¹, Marcello Restelli¹, and Nicola Gatti¹

¹Politecnico di Milano, Milan, Italy

²ML cube, Milan, Italy

{marco.mussi, gianmarco.genalti, francesco1.trovo, marcello.restelli, nicola.gatti}@polimi.it
alessandro.nuara@mlcube.com

Abstract

According to the main international reports, more pervasive industrial and business-process automation, thanks to machine learning and advanced analytic tools, will unlock more than 14 trillion USD worldwide annually by 2030. In the specific case of pricing problems—which constitute the class of problems we investigate in this paper—the estimated unlocked value will be about 0.5 trillion USD per year. In particular, this paper focuses on pricing in e-commerce when the objective function is profit maximization and only transaction data are available. This setting is one of the most common in real-world applications. Our work aims to find a pricing strategy that allows defining optimal prices at different volume thresholds to serve different classes of users. Furthermore, we face the major challenge, common in real-world settings, of dealing with limited data available. We design a two-phase *online learning* algorithm, namely PVD-B, capable of exploiting the data incrementally in an online fashion. The algorithm first estimates the demand curve and retrieves the optimal average price, and subsequently it offers discounts to differentiate the prices for each volume threshold. We ran a real-world 4-month-long A/B testing experiment in collaboration with an Italian e-commerce company, in which our algorithm PVD-B—corresponding to A configuration—has been compared with human pricing specialists—corresponding to B configuration. At the end of the experiment, our algorithm produced a total turnover of about 300 KEuros, outperforming the B configuration performance by about 55%. The Italian company we collaborated with decided to adopt our algorithm for more than 1,200 products since January 2022.

Introduction

Most international economic forecasts agree that nearly 50% of the annual value unlocked by the adoption of Artificial Intelligence (AI) from 2030 on will be in marketing&sales (Chui et al. 2018). Examples of activities in which AI tools can play a central role for marketing&sales include attracting and acquiring new customers, suggesting and recommending products, and optimizing customers’ retention and loyalty. In particular, AI can effectively automate these processes so as to increase their efficiency dramatically.

This paper focuses on pricing for e-commerce when, as it is usual, the objective is profit maximization and only transaction data are available. In particular, we focus on settings in which an e-commerce website sells goods other than luxury, Veblen, and Giffen. Thus, we can assume without loss of generality that the demand curve is monotonically decreasing in the price. Furthermore, we assume that the e-commerce website works with different classes of customers both in B2B and B2C scenarios. However, at the stage the price of the product is chosen and displayed to a user, the seller does not know whether the user comes from the former or latter scenario. Usually, volume discount is used to deal with multiple classes of users when it is not possible to distinguish the classes at the price formation stage. In particular, the rationale is to propose different *prices* for different *volume thresholds* thanks to the introduction of discounts. This approach allows showing the same thresholds and prices to all incoming users and, at the same time, it introduces price discrimination to provide a different pricing strategy for different classes of users. To the best of our knowledge, even if the problem of learning the price that maximizes the seller’s revenue has been extensively studied in the economic (Klenow and Malin 2010), game theory (Kopalle and Shumsky 2010) and learning (Den Boer 2015) fields, no dynamic pricing algorithm in the literature deals with volume discounts in a data-driven way.

Original Contribution In this work, we design an online-learning pricing algorithm, namely the *Pricing with Volume Discounts Bandit* (PVD-B) algorithm. We face the problem of assigning different prices to different volume thresholds using transaction data (coming from historical purchases and during the operational life of the e-commerce website). Given the complex dynamics of the problem, we decompose the algorithm into two phases: an optimal average price estimation and, based on the above estimation, a price adaptation method to provide different prices for the given volume discount thresholds. The adoption of tools from online learning guarantees the convergence to the optimal prices.

In collaboration with an Italian e-commerce website, we ran a real-world 4-month-long A/B testing experiment over a set of ≈ 300 products, in which our algorithm PVD-B—corresponding to A configuration—has been compared with human pricing specialists—corresponding to B configura-

*These authors contributed equally.

tion. At the beginning of the test, the available data concerned the purchases occurred in the previous 2 years. At the end of the experiment, the total turnover of A configuration was more than 300 KEuro and our algorithm PVD-B performed better than the B configuration in terms of the objective function (*i.e.*, total profit) for about 55%. The company we collaborated with decided to adopt our algorithm for more than 1,200 products since January 2022.

Related Works

A comprehensive analysis of the dynamic pricing literature is provided in Narahari et al. (2005); Bertsimas and Perakis (2006); Den Boer (2015). In particular, Multi-Armed Bandits (MAB) techniques have been extensively employed for dynamic pricing when the available information concerned the interactions between the e-commerce and customers.

Rothschild (1974) presents one of the seminal works on the adoption of MAB algorithms for dynamic pricing. This algorithm has been subsequently extended in several directions to capture the characteristics of different pricing settings. Kleinberg and Leighton (2003) study the problem of dealing with continuous-demand functions and proposes a discretization of the price values to provide theoretical guarantees on the regret of the algorithm. This approach suffers from the drawback that the reward is assumed to have a unique maximum in the price. Such an assumption is hard to be verified in practice. Instead, Trovò et al. (2015, 2018) relaxed this assumption, assuming that the demand function is monotonically decreasing and exploiting this assumption in the learning algorithm to provide uncertainty bounds tighter than those of classical frequentist MAB algorithms. However, the model formulation explicitly imposes neither monotonicity nor weak monotonicity on the estimated demand functions, so decisions that violate business logic can be allowed during the learning process. The authors show how the monotonicity assumption does not improve the asymptotic bound of regret provided by the MAB theory. On the other hand, exploiting monotonicity allows for an empirical improvement in performance (Mussi et al. 2022). The same argument also holds for the work proposed by Misra, Schwartz, and Abernethy (2019), where the monotonicity property of the demand function is used to ensure faster convergence. However, monotonicity is not forced as a model-specific feature. Besbes and Zeevi (2015) show that linear models are a suitable and efficient tool for modeling 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 that adopt a parametric formulation of the demand function are by Besbes and Zeevi (2009); Broder and Rusmevichientong (2012). These works assume stationary customer behavior. Bauer and Jannach (2018); Cope (2007) are two of the main works on Bayesian inference applied to dynamic pricing. They both fail to impose monotonic constraints on the model. Interestingly, Bauer and Jannach (2018) take into account non-stationary features (*e.g.*, competitors' prices). Araman and Caldentey (2009) use a Bayesian approach to dynamic pricing using a prior belief on the parameters to capture market-related information and force the model to be monotonic. Wang,

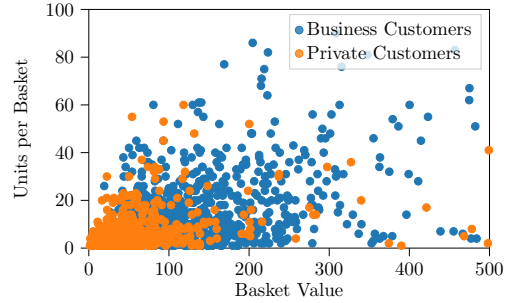


Figure 1: Units per basket and basket values for different classes of users.

Chen, and Simchi-Levi (2021) investigates non-parametric models for demand function estimation. In this case, the authors assume that the demand function is smooth. Finally, Nambiar, Simchi-Levi, and Wang (2019) propose a model to deal with both the non-stationarity data and the model misspecification. However, the required contextual knowledge at a product-wise level is not usually available in practice.

To the best of our knowledge, the literature lacks a data-driven methodology for finding an optimal volume discount pricing schedule to maximize retailers' profits and revenues in a B2C environment. The works from Hilmola (2021); Rubin and Benton (2003) focus on the *Economic Order Quantity* (EOQ) model that requires demand size over an annual budget and stock size. Sadrian and Yoon (1992) relax the EOQ hypothesis and provide a rational and straightforward pricing strategy that forces a lower bound on the company's expected profit by calculating volume thresholds and corresponding discounts afterward. The authors show the importance of volume discounts when increasing higher-priced products sales.

Problem Formulation

We study the scenario in which an e-commerce website sells non-perishable products with unlimited availability. The assumption of independence among the products allows us to focus singularly on every product. The extension to the case with a set of products is straightforward.

Commonly, the behavior of the users purchasing items from the e-commerce website is fragmented into multiple classes. For instance, Figure 1 provides the shopping baskets cardinality (in terms of units and economic value) for the e-commerce under analysis in this work, distinct classes of users: privates and businesses. The figure highlights how the privates purchase smaller amounts of products while the business is transacting with larger amounts. This fact suggests that an optimal seller strategy may include different pricing for the two. However, the user classes are not disclosed until payment is made, and, therefore, a pricing scheme that explicitly uses such a feature is not a viable option. In this work, we circumvent the issue of lack of customer information by using a *discount threshold scheme* that differentiates the per-unit price of the items by using the number of items purchased in a single transaction as a proxy

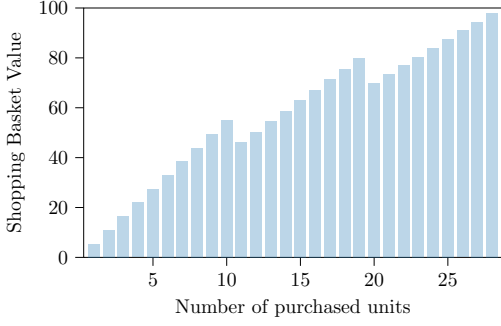


Figure 2: Effect on the basket value given a volume discounts scheme.

to distinguish the two classes.¹² In addition, further benefits of such a pricing scheme have been shown in the economic literature, *e.g.*, in Monahan (1984). Indeed, this model has been shown to anticipate buyer behavior and increase average order size, allowing the retailer to access the supplier's rebate on large restocks, reduce processing costs, and anticipate cash flows through the fiscal year.

Formally, for a given time t , let us define a vector of volume thresholds $\omega_t := [\omega_{1t}, \dots, \omega_{\eta t}] \in \mathbb{N}^\eta$, with $\omega_{it} > \omega_{ht}$, for each $i > h$ and $\omega_{1t} = 1$. The corresponding price vector will be $\mathbf{p}_t := [p_{1t}, \dots, p_{\eta t}] \in \mathcal{P}^\eta$, with $p_{1t} > \dots > p_{\eta t} > 0$. \mathcal{P} is the set of feasible prices, and $\eta \in \mathbb{N}$ is the number of thresholds. The i -th element p_{it} of \mathbf{p}_t denotes the price proposed for each product when a user wishes to purchase a number of them in $\{\omega_i, \dots, \omega_{i+1} - 1\}$.³ Figure 2 exemplifies the mechanism of the thresholded volume discounts when $\eta = 3$. The seller's objective is to maximize the *per-round profit*, defined as:

$$R_t(\mathbf{p}_t, \omega_t) := \sum_{i=1}^{\eta} (p_{it} - c) \cdot v_i(\mathbf{p}_t, \omega_t, t), \quad (1)$$

where $c \in \mathbb{R}^+$ is the unit cost of an item (assumed constant) and $v_i(\mathbf{p}_t, \omega_t, t) \in \mathbb{N}$ is the number of items sold at time t and at price p_{it} , when the purchase consists of a number of item in $\{\omega_i, \dots, \omega_{i+1} - 1\}$ and the overall seller strategy consists of prices \mathbf{p}_t and thresholds ω_t .⁴

However, in a real-world scenario, the functions $\{v_i(\cdot, \cdot, \cdot)\}_{i=1:\eta}$ are unknown to the seller and, therefore, need to be estimated using the transactions collected over

¹Notice that for the sake of presentation, the example presented two classes, but multiple (> 2) behaviors may exist, requiring multiple prices and thresholds.

²We remark that all the different prices (and related quantity thresholds) are displayed to any customer visiting the product web page.

³For instance, if $\omega_t = [1, 3, 5]$ and $\mathbf{p}_t = [6, 5, 4]$, a customer purchasing 2 units will pay $2 \cdot 6 = 12$, and a customer purchasing 4 units will pay them $4 \cdot 5 = 20$.

⁴Let us remark that this formulation can be extended in a straightforward way if the seller's goal also concerns the turnover, *i.e.*, by defining $R_t(\mathbf{p}_t, \omega_t)$ as a convex combination of turnover and per-round profit.

time. Notice that the volumes $v_i(\mathbf{p}_t, \omega_t, t)$ for the i -th volume interval also depend on the choices of the other prices and thresholds, as users might be more prone to purchase more items if there is a significant difference in price than buying fewer products in a single round. In this way, the problem can naturally be cast as a Multi-Armed Bandit (MAB) problem (see, *e.g.*, Lattimore and Szepesvári (2020) for a comprehensive review of MAB methods) where the goal is to properly balance the acquisition of information on the functions $v_i(\cdot, \cdot, \cdot)$, while maximizing the cumulative reward, *a.k.a.* exploration/exploitation dilemma. Formally, in a MAB problem, we are given a set of available options (*a.k.a.* *arms*), and we choose an arm at each time t . In our case, the arms are all the possible prices \mathbf{p}_t and thresholds ω_t , and the goal is to maximize the reward (in our setting profit) over a time horizon of T round. A policy \mathcal{U} is an algorithm that returns at each time t a pair (\mathbf{p}_t, ω_t) based on the information, *i.e.*, volumes \tilde{v}_{it} and corresponding prices p_{it} , we collected in the previous $t - 1$ rounds. A policy is evaluated in terms of *average total reward*, *i.e.*, our goal is to design policies that maximize:

$$R_T(\mathcal{U}) := \sum_{t=1}^T \sum_{i=1}^{\eta} (p_{it} - c) \cdot v_i(\mathbf{p}_t, \omega_t, t). \quad (2)$$

It is common in the MAB literature to use regret instead of reward as a performance metric. However, the minimization of the former corresponds to the maximization of the latter; therefore, our goal is the standard in MAB settings. Here, the total reward has been selected as a performance metric since it does not require knowledge of the optimum price strategy, which is unknown in the real world.

Algorithm

The problem presented before is computationally heavy (*i.e.*, exponential in the number of thresholds η) and cannot be addressed effectively in the presence of scarce data. Indeed, estimating the volume functions $v_i(\mathbf{p}_t, \omega_t, t)$, each of which has $2\eta + 1$ input parameters, would take a long time due to the requirement of collecting a large amount of transaction data. In what follows, we approximate the original problem in two different directions to allow learning the volume functions in short time.

We assume that the i -th volume function depends only on the price p_{it} selected for the i -th interval and on time t , or, formally, $v_i(\mathbf{p}_t, \omega_t, t) = v_i(p_{it}, t)$. Let us define the function of the total volumes provided by a pricing strategy (\mathbf{p}_t, ω_t) as:

$$\bar{v}(\bar{p}_t, t) := \sum_{i=1}^{\eta} v_i(p_{it}, t), \quad (3)$$

where \bar{p}_t is a weighted average value of the prices vector \mathbf{p}_t , formally:

$$\bar{p}_t = \sum_{i=1}^{\eta} \alpha_{it} \cdot p_{it}, \quad (4)$$

where $\alpha_i \in [0, 1], \forall i \in \{1, \dots, \eta\}$ must be estimated guaranteeing that on average a threshold pricing strategy

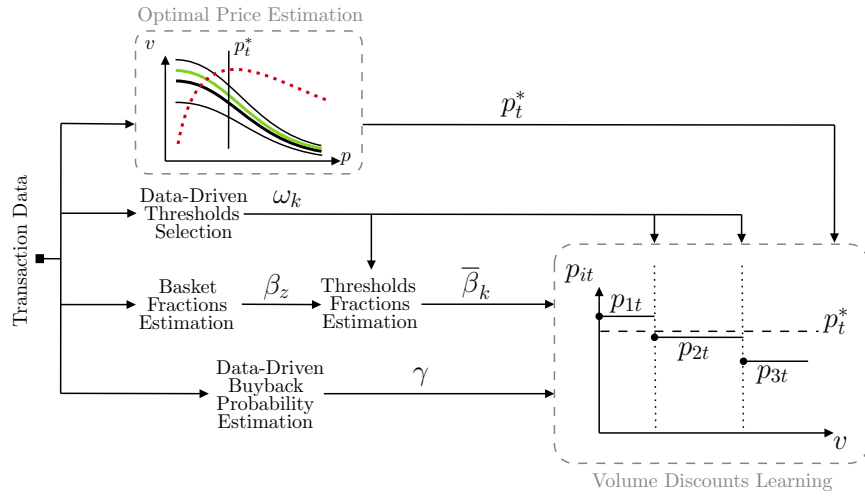


Figure 3: General overview of the PVD-B algorithm.

$\{\omega_{it}, p_{it}\}_{i=1}^{\eta}$ yields a reward greater than or equal to the theoretical one formulated as follows:

$$\bar{R}_t(\bar{p}_t) := (\bar{p}_t - c) \cdot \bar{v}(\bar{p}_t, t). \quad (5)$$

Thanks to the previous definitions, we can reformulate the optimization problem into two consecutive steps:

- Finding a single optimal price p_t^* that maximizes the revenue provided by the total volume function defined as $R_t^*(p_t^*) := (p_t^* - c) \cdot v(p_t^*, t)$;
- Given p_t^* , find a pricing strategy (p_t^*, ω_t^*) whose weighted average (see Equation 4) to the optimal price p_t^* .

Notice that the second step allows the algorithm to use all the data available for estimating the function $v(\cdot, \cdot)$, instead of partitioning them into η disjoint sets and independently estimating the $v_i(\cdot, \cdot)$ functions. This allows us to speed up the learning process. In what follows, we detail the two phases of the PVD-B algorithm, which corresponds to the solution of the above problem, *i.e.*, the *Optimal Price Estimation* and *Volume Discounts Learning* phases. The overall procedure is depicted in Figure 3. More specifically, the former phase aims to estimate the optimal price p_t^* for the total volumes relying on the transaction data. Instead, the latter phase combines the previous estimate of the optimal price p_t^* with the parameters extracted from the transaction data to compute the optimal thresholds ω_t^* and the pricing strategy p_t^* .

Optimal Price Estimation As a first step, we estimate the optimal average price p_t^* . The algorithm takes as input the records of past orders, *i.e.*, tuples $(\tilde{p}_{it}, \tilde{v}_{it}, t)$ with the price, volume and time of each user purchase occurred in the past time instants. For each time t , it computes the tuple $(\bar{p}_t, \bar{v}_t, t)$, where the average price \bar{p}_t and the total volume \bar{v}_t is computed as described in Equations (4) and (3), respectively, relying on the above-mentioned collected data.

These data are used to compute an estimate $\hat{v}(\cdot, \cdot)$ of the total volume function $v(\cdot, \cdot)$. The PVD-B algorithm resorts to a Bayesian Linear Regression (BLR, Tipping 2001)

model to approximate the function $v(\cdot, \cdot)$. Formally, the estimates of the total volume for price p at time τ is equal to:

$$\hat{v}(p, \tau) = \sum_{u=1}^U \theta_u \phi_u(p) + \sum_{d=1}^D \theta_d \phi_d(\tau), \quad (6)$$

where $\phi_1(p), \dots, \phi_U(p)$ are the basis functions constructed over the price p having as prior a Lognormal distribution and $\phi_1(\tau), \dots, \phi_D(\tau)$ are the basis functions constructed over the time τ having as prior a Gaussian distribution. Two remarks are necessary. First, the basis $\phi_d(\cdot)$ has been introduced to consider the seasonality that affects the selling process of the e-commerce website. Second, the choice of Lognormal prior for the price basis $\phi_u(\cdot)$ forces such distributions to be non-negative, which in turn induces the monotonicity of the approximating function w.r.t. the price p (see Wilson et al. (2020) for more details). As discussed before, this property is also commonly reflected by real demand functions and allows for fast learning of such curves.

The output of the BLR regression model provides a distribution $\hat{v}(p, \tau)$ for each p , allowing the use of MAB algorithms and, in particular, the use of a Thompson Sampling (TS)-like (Kaufmann, Korda, and Munos 2012; Agrawal and Goyal 2012) approach, as a strategy to find a value for the optimal price p_τ^* balancing exploration and exploitation. The corresponding procedure is summarized in Figure 4. More specifically, from the distribution $\hat{v}(\cdot, \tau)$ (Left, represented as expected values and uncertainty bounds), we sample a function $\hat{v}_{TS}(\cdot, \tau)$ (Center, represented in green), and, finally, we perform the optimization (Right) of the profit as follows:

$$p_\tau^* \in \arg \max_{p \in \mathcal{P}} (p - c) \cdot \hat{v}_{TS}(p, \tau). \quad (7)$$

Volume Discounts Learning Let η be the number of volume thresholds to show to customers, along with the corresponding prices.⁵ Let β_z , with $z \in \mathbb{N}$, be the proportion of

⁵Here, we assume the e-commerce experts provide this value. If not provided, clustering techniques over historical data can be used

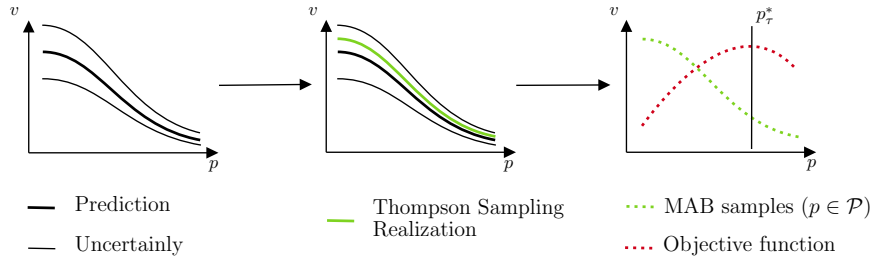


Figure 4: Procedure to retrieve optimal price p_τ^* at time τ using a TS-like approach.

baskets containing the product with a volume of z . The average volume for the product in each basket 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:

$$\bar{\beta}_k = \sum_{i=\omega_k}^{\omega_{k+1}-1} \beta_i. \quad (8)$$

The average volume of products for the baskets in a given threshold ω_k is consequently defined as:

$$\bar{V}_k = \frac{\sum_{i=\omega_k}^{\omega_{k+1}-1} \beta_i \cdot i}{\sum_{i=\omega_k}^{\omega_{k+1}-1} \beta_i}. \quad (9)$$

Suppose a customer needs N units of the given product. This need can be fulfilled by dividing his/her order across any number of time steps, *i.e.*, performing a purchase of units (or a *volume*) in a range $\{\omega_k, \dots, \omega_{k+1} - 1\}$ for a specific k and repeating such a purchase until the required amount of units is reached. After the customer bought the product, he/she has a probability γ of returning to the same retailer buying another batch of the same size. This kind of modeling of the user's behavior is reflecting accurately those customer buying goods with a short lifespan and, in general, the ones for which a customer is led to schedule periodic purchases (*i.e.*, toilet paper, consumable office supplies). With probability $1 - \gamma$, the customer will not return the next time. We consider γ a property of the system, so we assume that the price does not affect the buyback probability. In what follows, we use historical transaction data to estimate the value of γ for a given product.

Let \bar{m} denote the desired margin when a single unit is purchased. It follows that the expected margin $\bar{\mu}$ coming from a customer with a need of N units and who performs only single-unit orders is:

$$\bar{\mu} = \sum_{\tau=1}^N \gamma^{\tau-1} \bar{m} = \frac{1 - \gamma^N}{1 - \gamma} \bar{m}, \quad (10)$$

where the last equality comes from the truncated geometric series identity. A customer with the same need, but whose orders contain a number of units in $\{\omega_k, \dots, \omega_{k+1} - 1\}$, which are associated with a margin \bar{m}_k , will generate the

to determine the optimal number of groups in the user distribution, *e.g.*, Pelleg, Moore et al. (2000).

following expected margin:

$$\bar{\mu}_k = \sum_{\tau=1}^{\left\lceil \frac{N}{\bar{V}_k} \right\rceil} \gamma^{\tau-1} \bar{m}_k \bar{V}_k = \frac{1 - \gamma^{\left\lceil \frac{N}{\bar{V}_k} \right\rceil}}{1 - \gamma} (1 - \delta_k) \bar{m} \bar{V}_k, \quad (11)$$

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

$$\bar{m}_k = \bar{m}(1 - \delta_k), \quad k \in \{1, \dots, \eta\}, \quad (12)$$

where $\delta_1 = 0$. By imposing $\bar{\mu}_k \geq \bar{\mu}_1$, we get:

$$\delta_k \leq 1 - \frac{1 - \gamma^N}{\bar{V}_k \left(1 - \gamma^{\left\lceil \frac{N}{\bar{V}_k} \right\rceil}\right)}. \quad (13)$$

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

$$\sum_{k=1}^{\eta} \bar{V}_k \bar{m}_k = m_t^* \bar{V}. \quad (14)$$

Substituting Eq. 12 into Eq. 14, we get:

$$\bar{m} = \frac{m_t^* \bar{V}}{\sum_{k=1}^{\eta} (1 - \delta_k) \bar{V}_k}. \quad (15)$$

Finally, the margins $\bar{m}_1, \dots, \bar{m}_\eta$ for the different volume thresholds are determined as $\bar{m}_k = \bar{m}(1 - \delta_k)$, for $k \in \{1, \dots, \eta\}$, where $\delta_1 = 0$. The complete algorithm, including both optimal average price estimation and volume discounts, is summarized in Figure 3.

Data-Driven Buyback Probability Estimation Even if estimating γ in an online fashion would be a natural approach, it is prohibitive due to our environment's strong seasonality and non-stationary nature. Indeed, studying customers' churn usually requires a large amount of contextual data and is a challenging task for many fields (Kamalraj and Malathi 2013). Instead, we propose a methodology purely based on the available transaction data (where the customers are uniquely identified) to estimate γ in an offline fashion. We define two time-intervals: a "*measure*" period $\mathcal{T}_M \in \mathcal{T}$

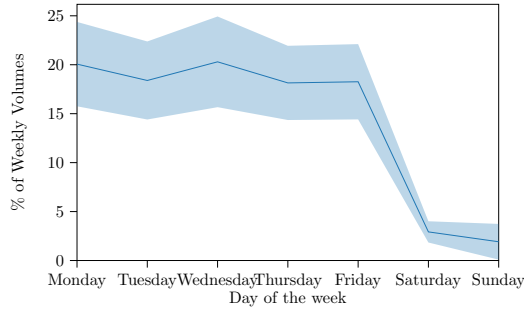


Figure 5: Seasonality over a single week (mean \pm std).

and a “control” one $\mathcal{T}_C \in \bar{\mathcal{T}}$, where $\mathcal{T}_M \cap \mathcal{T}_C = \emptyset$ and $\bar{\mathcal{T}}$ is the set of time periods *i.e.*, contiguous sequences of times. Intuitively, we observe which customers buy during the “measure” period and compute what percentage of them come back in the subsequent period, the “control” one. Formally, given a set of customers $\mathcal{G} := \{g_1, \dots, g_L\}$, we define a function $\mathcal{H} : \mathcal{G} \times \bar{\mathcal{T}} \rightarrow \mathbb{N}$ that associates a customer to the number of purchases made in a specific period. We also introduce $h : \bar{\mathcal{T}} \rightarrow \mathcal{P}(\mathcal{G})$, which maps a period of time into the subset of unique customers who made at least one purchase in that period.⁶ Thus, we are able to compute the total number of returns \mathcal{R} and non-returning customers \mathcal{A} , formally $\mathcal{R} = \sum_{g \in \mathcal{G}} [\mathcal{H}(g, \mathcal{T}_M) - |h(\mathcal{T}_M)| + |h(\mathcal{T}_M) \cap h(\mathcal{T}_C)|]$, and $\mathcal{A} = |h(\mathcal{T}_M)| - |h(\mathcal{T}_M) \cap h(\mathcal{T}_C)|$. Notice that \mathcal{R} is composed by those customers that have already occurred during the “measure” period (since a customer that purchased n times during \mathcal{T}_M already returned $n-1$ times) and those happened during the “control” period (customers seen in both periods). Instead, \mathcal{A} is the number of customers who purchased at least one time in the “measure” period and did not show up during the “control” one. Notice that the functions \mathcal{H} and h can be easily calculated starting from transaction data once the two periods have been defined. In our test, we decided to use the 6 months before the experimental campaign as a “control” period and the previous 6 months as a “measure” one. Finally, thanks to the two quantities defined above, the value of γ can be approximated as $\gamma = \frac{\mathcal{R}}{\mathcal{R} + \mathcal{A}}$.

Data-Driven Threshold Selection Threshold values $\{\omega_k\}_{k=1}^\eta$ can be selected in several ways. Our solution is to define a split criterion that divides the products within the shopping baskets into η sets of equal cardinality. Formally, we define $q : \mathbb{N} \rightarrow \mathbb{N}$ as the function that maps a number of units into the number of shopping baskets that contain that many units. Notice that if B is the total number of shopping baskets over the period examined, then $q(z) = B \cdot \beta_z$, where the values of β_z and B can be estimated from the transaction data, and, consequently, $q(\cdot)$ can be computed entirely from data. Intuitively, we can build a data set where each $z \in \mathbb{N}$ is repeated $q(z) \cdot z$ times, defined as:

$$Q := \{\underbrace{1, \dots, 1}_{q(1) \text{ times}}, \dots, \underbrace{d, \dots, d}_{d \cdot q(d) \text{ times}}, \dots\}. \quad (16)$$

⁶With $\mathcal{P}(A)$ we denote the power set of A .

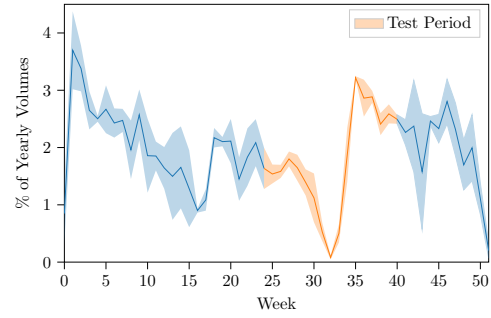


Figure 6: Seasonality over the weeks of an year (mean \pm std).

To get the k -th threshold, we extract the $\lceil |Q| \cdot \frac{k}{\eta} \rceil$ -th element from the sorted data set Q defined as in Eq. 16.

Experimental Evaluation

We performed a real-world experiment in collaboration with an Italian e-commerce company in which our algorithms priced a set of products adopting a long-tail economic model (Anderson 2006). The e-commerce website collects data on each purchase (date and time), as a row of a transaction data set, including features such as the identifier of the purchased product, the number of units sold, the price, the cost, and the class of the customer (business or private) inferred from the fiscal status observed after the purchase. The experimental campaign focused on products usually bought with high volumes to evaluate better our algorithms.

We resorted to an online A/B. The experimental campaign was conducted in one of the main categories of the e-commerce website, with a test set (A) composed by $N_t = 295$ products and a control set (B) composed of $N_c = 33$ products of the same category and with the same characteristics.⁷ The test included products with a yearly turnover of 300 KEuros and a total profit of 83 KEuros.

The algorithm produces new prices every 7 days since a significant intra-week seasonality has been observed (see Figure 5). Moreover, the products sold by the e-commerce website are subject to a significant seasonality over different periods of the years, as shown in Figure 6. Due to the particular kind of products sold we dealt with and the nature of the target customer segment, volume discounts are crucial to the business since they affect customers’ loyalty and the logistic organization of the company. The e-commerce website’s specialists defined the number $\eta = 3$ of volume thresholds that should be displayed for every product. The test was conducted for 17 weeks, from 16 June 2021 to 17 October 2021, during which no communication and marketing actions were performed in attempt not to influence the customers’ behavior.

The business goal was to maximize test set’s (A) average profit $R_T^{(A)}$, as defined in Eq. (2), where $T = 17$. This score is to be compared with that one achieved by the B set

⁷The test and the control sets were defined by e-commerce specialists according to both technical and market issues.

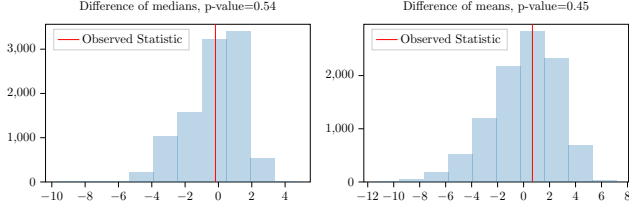


Figure 7: Distribution of the two-sided permutation tests statistics before the test, $R = 10000$ random permutations.

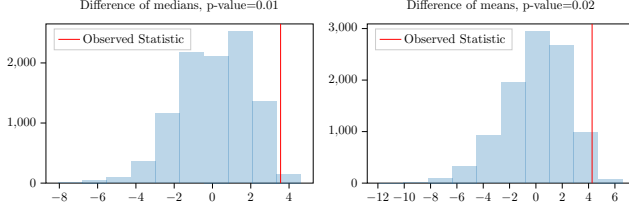


Figure 8: Distribution of the two-sided permutation tests statistics after the test, $R = 10000$ random permutations.

$R_T^{(B)}$ over the same period. To evaluate the performance of our algorithm, we performed a statistical test applied to the product margins to check whether the two groups are comparable. More specifically, for each product, we computed the average weekly net margins (R_t as in Eq. 1) during the first six months of 2021 (*i.e.*, t is in the first 26 weeks of 2021) and we design a test to check if the median and mean of the net margin across the products of the A set are larger than the ones of the B set. We performed one-sided permutation tests with the null hypothesis being “The A set has not a higher median/mean of net margin w.r.t. the B set”. Figure 7 shows the distributions of the tests’ statistics together with the observed ones, in which the resulting p -values concerning medians and means are respectively 0.54 and 0.45, and, therefore, resulting in the fact that there is not enough statistical evidence to say the two are different. This shows that set A has not a larger median/mean w.r.t. set B on the chosen performance metric before the beginning of the test.

We use the BLR basis functions following different criteria to perform the test mentioned above. To model the price elasticity over the customers’ base, we choose *reverted hyperbolic tangent* functions. Instead, to grasp the irregular nature of the e-commerce’s seasonality, we choose *Radial Basis Functions* (RBF). Finally, the trend is modeled by choosing polynomial basis functions. Both RBF and reverted hyperbolic tangents are evaluated with different shifts and scales, while polynomial features with different degrees.

The algorithm ran in a Docker container with *Python 3.8* environment on *Linux*. Every week, the algorithm made an SQL query to retrieve that data about the products and then returned the prices. The hardware was a *Quad-core Intel Core i7 8th Gen* with *8Gb DDR4 RAM*. The time required for a run of the algorithm over all the products is about 25 minutes. Given that the algorithm is applied to each product independently of the others, the running time scale linearly w.r.t. the number of products.

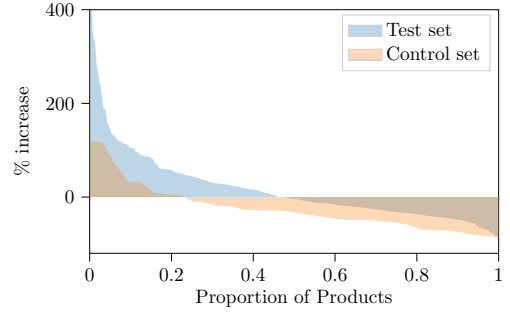


Figure 9: Distribution of the objective function improvements in Test and Control set.

Results The goods priced by PVD-B during the testing period provided an improvement (on average) in term of the performance metric $R_T^{(A)}$ of +55% w.r.t. the one $R_T^{(B)}$ of control set of goods, or formally $\frac{R_T^{(A)}}{R_T^{(B)}} = 1.55$. After 17 weeks, we performed the same statistical test on the weekly performance metric obtained between the two sets of products during the test period. Figure 8 shows the distribution of the test’s statistics along with the observed ones. 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 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 larger median and mean of the average weekly performance metric w.r.t. the control set of products, with at least a confidence of 98%.

Regarding the performances on a product-wise level, we report in Figure 9 the sorted percentages of improvement on the performance metric w.r.t. to the period of 2021 preceding the test for every single product. In the test set, 138 products over 295 ($\approx 47\%$) improved their average performance w.r.t. the corresponding period of 2021, while in control set only 8 products over 33 ($\approx 25\%$) were able to improve. This corroborates the idea that the proposed method is able to improve the performance of the e-commerce website by influencing the purchase process of a large number of products.

Effect of Volume Discounts

A final analysis consists in evaluating how the volume discounts algorithm can modify the probability distribution of the units count of the same product in a basket. More precisely, we need to check whether the algorithm affects the volumes $\bar{\beta}_k$ to increase the profit. In our specific setting, this corresponds to check if the value $\bar{\beta}_1$ decrease in favor of $\bar{\beta}_2$ and/or $\bar{\beta}_3$. For this analysis, the parameters of the volume-discount algorithm have been estimated using the period from 16 June 2019 to 16 June 2021, and the estimation of γ was performed on the data split in an estimation period \mathcal{T}_M from 17 June 2019 to 16 June 2020 and a control one \mathcal{T}_C from 17 June 2020 to 16 June 2021.

We analyze the context from both customers and products perspectives. Figure 10 shows the distribution of the number of orders performed by the customers. The histogram shows almost half of the customers perform a single order and leave

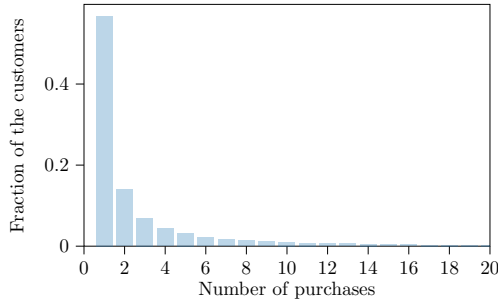


Figure 10: Number of purchases done by the customers.

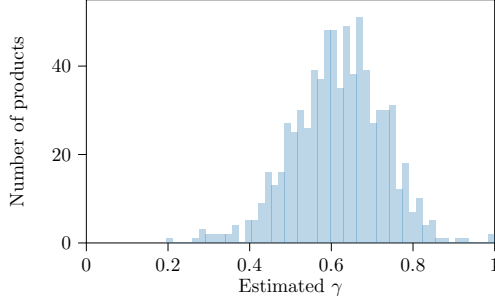


Figure 11: Distribution of the parameter γ .

the shop without coming back. Figure 11 analyze the same phenomenon from the product perspective. Given a product, we can interpret the γ parameter as the probability that a customer buying that product will, sooner or later, buy such product again. Higher is the γ , and more conservative will be the discount strategy we adopt.

Results The estimated per-product discounts between the three thresholds are presented in Figure 12. The PVD-B algorithm applies an average discount of $\approx 10\%$ for the second volume interval and $\approx 20\%$ for the third one. This implies a shift in the number of purchases among the intervals. In Table 1, the variations of the three $\bar{\beta}_k$ are reported: during the test, we achieved an increase of the values $\bar{\beta}_2$ and $\bar{\beta}_3$ while observing a reduction in $\bar{\beta}_1$. Finally, in Table 2 we report the average variations in terms of average units per basket of the 4 above-mentioned products during the test period. The effect of applying the PVD-B algorithm and, therefore, introducing volume discounts modifies the basket’s average size by increasing the units purchased by $\approx 33\%$.

Considerations After the A/B Test

After the end of the A/B test, the e-commerce specialists were satisfied with the achieved results, including the performance of the volume-discount algorithm. Thus, the company extended the adoption of our algorithm to all the products presenting a sufficient amount of volumes in the catalog of the e-commerce website (≈ 1200 products). Currently, our algorithm prices about 1,200 products generating a cumulative annual revenue of about 1.5 MEuro, which corresponds to about 50% of the total e-commerce website turnover. Furthermore, the algorithm now runs in the cloud

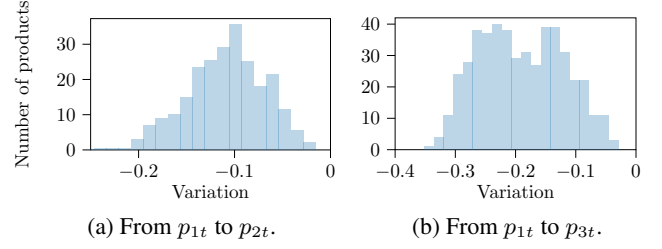


Figure 12: Average (on time) discounts between volumes’ thresholds in test products.

Table 1: Variations of $\bar{\beta}_k$ after the test period.

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 2: Variation of units per basket after the test period.

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

in a SaaS fashion. An automatized routine runs a query on the dataset of the e-commerce website, extracting the transaction data needed and then running the algorithm. The results are provided to the business unit in a `csv` file.

Conclusion and Future Works

In this paper, we present PVD-B, an algorithm capable of defining the price and volume discounts in an online settings. Our approach exploits the transaction data of the e-commerce website to optimize the pricing strategy in an online fashion. We test our approach in a real-world 4-months experiment by optimizing the price of 295 products of an e-commerce website. The results show that our approach increases the e-commerce website profits by outperforming the previous management and gaining an increase of 55%.

In future works, we plan to insert in the model time correlations between the purchases and the effect of loyalty in increasing revenue. Furthermore, in this work, we price products independently, while cross-selling approaches could further increase profits for some classes of products. The design of algorithms taking into account also these dependencies constitutes an interesting new line of work.

References

- Agrawal, S.; and Goyal, N. 2012. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, 39–1. JMLR Workshop and Conference Proceedings.
- Anderson, C. 2006. *The long tail: Why the future of business is selling less of more*. Hachette Books.
- Araman, V. F.; and Caldentey, R. 2009. Dynamic pricing for nonperishable products with demand learning. *Operations research*, 57(5): 1169–1188.
- Bauer, J.; and Jannach, D. 2018. Optimal pricing in e-commerce based on sparse and noisy data. *Decision Support Systems*, 106: 53–63.
- Bertsimas, D.; and Perakis, G. 2006. Dynamic pricing: A learning approach. *Mathematical and computational models for congestion charging*, 45–79.
- Besbes, O.; and Zeevi, A. 2009. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6): 1407–1420.
- Besbes, O.; and Zeevi, A. 2015. On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science*, 61(4): 723–739.
- Broder, J.; and Rusmevichientong, P. 2012. Dynamic pricing under a general parametric choice model. *Operations Research*, 60(4): 965–980.
- Chui, M.; Manyika, J.; Miremadi, M.; Henke, N.; Chung, R.; Nel, P.; and Malhotra, S. 2018. Notes from the AI frontier: Insights from hundreds of use cases. *McKinsey Global Institute*.
- Cope, E. 2007. Bayesian strategies for dynamic pricing in e-commerce. *Naval Research Logistics (NRL)*, 54(3): 265–281.
- Den Boer, A. V. 2015. Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in operations research and management science*, 20(1): 1–18.
- Hilmola, O.-P. 2021. Quantity Discount Algorithm in an E-Commerce Environment. In *Proceedings of International Conference on Communication and Computational Technologies*, 437–446. Springer.
- Kamalraj, N.; and Malathi, A. 2013. A survey on churn prediction techniques in communication sector. *International Journal of Computer Applications*, 64(5): 39–42.
- Kaufmann, E.; Korda, N.; and Munos, R. 2012. Thompson sampling: An asymptotically optimal finite-time analysis. In *International conference on Algorithmic Learning Theory*, 199–213. Springer.
- Kleinberg, R.; and Leighton, T. 2003. 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.*, 594–605. IEEE.
- Klenow, P. J.; and Malin, B. A. 2010. Microeconomic evidence on price-setting. In *Handbook of monetary economics*, volume 3, 231–284. Elsevier.
- Kopalle, P.; and Shumsky, R. A. 2010. Game Theory Models of Pricing. *Handbook of Pricing Management*.
- Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.
- Misra, K.; Schwartz, E. M.; and Abernethy, J. 2019. Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*, 38(2): 226–252.
- Monahan, J. P. 1984. A quantity discount pricing model to increase vendor profits. *Management science*, 30(6): 720–726.
- Mussi, M.; Genalti, G.; Trovò, F.; Nuara, A.; Gatti, N.; and Restelli, M. 2022. Pricing the Long Tail by Explainable Product Aggregation and Monotonic Bandits. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3623–3633.
- Nambiar, M.; Simchi-Levi, D.; and Wang, H. 2019. Dynamic learning and pricing with model misspecification. *Management Science*, 65(11): 4980–5000.
- Narahari, Y.; Raju, C.; Ravikumar, K.; and Shah, S. 2005. Dynamic pricing models for electronic business. *Sadhana*, 30(2): 231–256.
- Pelleg, D.; Moore, A. W.; et al. 2000. X-means: Extending k-means with efficient estimation of the number of clusters. In *International Conference on Machine Learning*, volume 1, 727–734.
- Rothschild, M. 1974. A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 9(2): 185–202.
- Rubin, P. A.; and Benton, W. 2003. A generalized framework for quantity discount pricing schedules. *Decision Sciences*, 34(1): 173–188.
- Sadrian, A. A.; and Yoon, Y. S. 1992. Business volume discount: A new perspective on discount pricing strategy. *International Journal of Purchasing and Materials Management*, 28(2): 43–46.
- Tipping, M. E. 2001. Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun): 211–244.
- Trovò, F.; Paladino, S.; Restelli, M.; and Gatti, N. 2015. Multi-armed bandit for pricing. In *European Workshop on Reinforcement Learning*, 1–9.
- Trovò, F.; Paladino, S.; Restelli, M.; and Gatti, N. 2018. Improving multi-armed bandit algorithms in online pricing settings. *International Journal of Approximate Reasoning*, 98: 196–235.
- Wang, Y.; Chen, B.; and Simchi-Levi, D. 2021. Multimodal dynamic pricing. *Management Science*.
- Wilson, A.; Tryner, J.; L’Orange, C.; and Volckens, J. 2020. Bayesian nonparametric monotone regression. *Environmetrics*, 31(8): e2642.