From Stream to Pool: Dynamic Pricing Beyond i.i.d. Arrivals

Anonymous Author
Anonymous Institution

Abstract

Dynamic pricing under uncertainty is often formulated as a multi-armed bandit (MAB) problem where a **stream** of customers arrive sequentially, each with an independently and identically distributed (i.i.d.) valuation drawn from a fixed, unknown distribution. However, this formulation is not entirely reflective of the real world. In numerous retail scenarios such as fashion clothing and electronic goods, customers with high valuations tend to make purchases earlier and subsequently leave the market, leading to a gradual *shift* in the valuation distribution towards the lower end.

Thus motivated, we consider a single-item revenue maximization problem where a **pool** of unit-demand, non-strategic customers interact repeatedly with a single seller. Each customer monitors the price intermittently according to an independent Poisson process and makes a purchase if she observes a price lower than her valuation, whereupon she leaves the market permanently. As the main challenge, the valuations of the customers are unknown but come from a given family of instances.

1 Introduction

Consider selling fashion clothing in a retail store. Customers visit the store sporadically, primarily depending on their own availability. They will make the purchase if the price is lower than their private valuation. For many types of goods, such as clothing, books, and electronic devices, each customer needs at most one unit. In other words, a customer leaves the market once a purchase has been made.

This problem falls under the umbrella of dynamic pricing with unknown demand, which has been extensively studied, see e.g., surveys by [Elmaghraby and Keskinocak, 2003, den Boer, 2015]. Many existing works (e.g., [Kleinberg and Leighton, 2003, Besbes and Zeevi, 2009, Babaioff et al., 2015]) in this area employ what we call a **stream** model: A stream of customers arrives sequentially, each with an independent identically distributed valuation drawn from a fixed distribution.

However, the stream model fails to capture many real-world scenarios. For example, in the fashion clothing scenario, the i.i.d. assumption fails in two ways: The demands are neither identical nor independent. Specifically, the demands in different rounds are not stationary. Consider the fashion clothing example mentioned at the beginning, once high-valuation customers make the purchase, they are likely to remain inactive for the whole selling season, therefore leading to a gradual *shift* in the valuation distribution towards the lower end.

Moreover, the reward at each round is not independent. In fact, even for a fixed price policy, the reward at the current round may depend on the realized reward of the previous round. If the reward from previous rounds is high, then many customers have already left the market, and hence the reward in this round should be low.

There is work that separately addresses these two concerns, but they do not capture the key features of the problem. There is work on pricing with non-stationarity e.g. (SJ: cite), but mostly with exogenous non-stationarity such as seasonality or economic factors. In the example

Thus motivated, we consider a single-item revenue maximization model where a **pool** of unit-demand, non-strategic customers interact repeatedly with a single seller. Each customer monitors the price intermittently according to an independent Poisson process and makes a purchase if she observes a price lower than her valuation, whereupon she leaves the market permanently.

Preliminary work. Under review by AISTATS 2024. Do not distribute.

As the main challenge, the valuations of the customers are unknown but come from a given family of instances.

(SJ: End intro by saying what we will do in this paper)

1.1 Our contribution

We would like to highlight the following contributions.
1. **A Novel Model.** We introduce a novel *pool*-based pricing model: Each customer monitors the price according to an independent Poisson process, makes a purchase when the observed price is below the valuation, and leaves the market *permanently*. In contrast to the *stream*-based model in most existing work, our model better encapsulates the key features of many retailing scenarios where the customers have unit demand. We show that this problem is tractable if the instance is known through the following results.

a) **Price Monotonicity.** We show that the price sequence in any optimal non-adaptive policy is non-increasing.

b) **Optimal Non-adaptive Policy.** We present an efficient dynamic programming based algorithm that computes the optimal non-adaptive policy; see Theorem 3.4.

2. **Optimal Algorithm for Non-adaptive Policy.** We first consider *non-adaptive* policies, i.e., policies that predetermine how the price changes, regardless of observed demands. These policies are particularly compelling and practical because of their operational simplicity. We provide a **complete** settlement of this setting by showing the following results.

a) **A k-Competitive Algorithm.** We present an efficient algorithm that takes a family of instances as input and returns *one* non-adaptive policy. We show that our algorithm is k-competitive for any family of k-price instances, i.e., the output policy is guaranteed to procure a $(1/k)$-fraction of the expected revenue achievable by any (possibly adaptive) policy with *full* knowledge of the true instance; see Theorem 4.1.

b) **A $(\log \rho)$-Competitive Algorithm.** The above guarantee is weak for large k. To mitigate this, we propose a variant of our algorithm that restricts its attention to a subset of prices. We show that this algorithm is $(\log \rho)$-competitive, where ρ is the ratio between the highest and lowest prices; see Theorem 4.4.

c) **Optimality.** Our algorithm achieves the (maximin) optimal competitive ratio. Specifically, for each $k \geq 1$, we construct a family of k-price instances on which no non-adaptive policy guarantees more than $1/k$ fraction of the optimal revenue on *all* instances in this family; see Theorem 4.5.

3. **Adaptive Policy with Vanishing Regret.** We present an adaptive *learn-then-earn* policy, which effectively balances the process of learning the underlying demand and optimizing revenue. We show that this policy achieves a regret of $\tilde{O}(n^{3/4})$, see Theorem 5.2. (SJ: This parag needs to be expanded.)

1.2 Related Work

dyn stream / unknown demand

cluster 2: monitor/ repeated interaction/ pool/ unit-demand

cluster 3: markdown

Our work is influenced by and contributes to several streams of literature across operations research, computer science, and economics. In the following, we give an overview of the related work and connect them to our paper.

Dynamic pricing in the stream model. The tradeoff between learning and earning has been studied extensively in the areas of operations research, computer science, and economics (see, e.g., [Auer et al., 2002, Auer, 2002, Segal, 2003, Wang et al., 2014]). The exploration has centered on the context of demand models with parametric forms and customers arriving sequentially with independently and identically distributed valuations. For instance, [Broder and Rusmevichientong, 2012] studied a stylized dynamic pricing model where a monopolist prices a product to a sequence of customers based on a general parametric unknown demand model (e.g., linear, logit, exponential demand families). The authors presented a pricing policy based on maximum likelihood estimation to obtain a tight sublinear regret bound. Similar models have been explored in other studies, such as [Besbes and Zeevi, 2009, den Boer and Zwart, 2013, Keskin and Zeevi, 2014, den Boer and Zwart, 2015, Besbes and Zeevi, 2015, Cohen et al., 2021, Wang et al., 2021, Bastani et al., 2022]. Recently, [den Boer and Keskin, 2020] extended the dynamic pricing problem to an unknown and discontinuous demand function setting. [Chen et al., 2022b, Chen et al., 2023] studied a dynamic personalized pricing problem with unknown demand under privacy protection.

For a comprehensive overview, we refer readers to the survey conducted by [den Boer, 2015]. The majority of these studies operate within the context of customers arriving sequentially with independently and identically distributed valuations, which can be characterized as the *stream*-based model. In contrast, our paper adopts a unique perspective, assuming the presence of all customers at the outset of the selling season, a model we refer to as the pool-based model. We assume customers intermittently monitor prices according to a Poisson process. This modeling choice aligns more closely with real-world market dynamics.

Pricing with customers repeatedly monitoring the price. Another stream of research that is related to our paper is pricing with strategic customers who repeatedly monitor the price (see, e.g., [Aviv and Pazgal, 2008, Dasu and Tong, 2010, Borgs et al., 2014, Li et al., 2014, Correa et al., 2016, Moon et al., 2018, Correa et al., 2020, Aviv et al., 2019, Deng et al., 2023]). [Su, 2007] analyzed optimal dynamic pricing strategies when consumers strategically wait for markdowns and proved the optimality of markdown policies when high-value customers are impatient or when low-value customers are patient. [Briceño-Arias et al., 2017] studied the optimal price path to sell a single item when there are a random number of strategic buyers who arrive over time. [Lobel, 2020] generalized the problem of dynamic pricing to the case customers arrive deterministically with heterogeneous patience levels. The strategic customer framework is closely related to our work in the sense that they automatically form a pool when they wait for markdowns, and they repeatedly interact with the seller. In addition, we show that when the demand is known, the optimal non-adaptive policy in our model is also a markdown policy. However, we depart from the strategic customer setting and assume customers monitor prices via a constant-rate Poisson process. They make the purchase once the price falls below their private valuation.

Multi-armed bandit. Our work is also situated within the broader literature concerning bandit problems, which have their origins in seminal works by [Thompson, 1933] and [Robbins, 1952]. The multi-armed bandit framework has evolved into a paradigmatic framework for studying dynamic optimization under conditions of incomplete information. Extensive research on contextual bandits can be found in the fields of operations research, computer science, and machine learning (see, e.g., [Filippi et al., 2010, Rusmevichientong and Tsitsiklis, 2010, Abbasi-Yadkori et al., 2011, Shi et al., 2016, Yuan et al., 2021, Chen et al., 2022a, Li et al., 2023]). In the adaptive pricing section of our work, we employ a similar algorithmic approach to formulate a pricing policy that achieves sublinear regret, aligning with the principles of contextual bandit research.

Markdown Pricing. As we will soon see, any non-adaptive non-increasing price sequence. In the field of revenue management, such policies are often referred to as markdown policies. (SJ: TO DO: markdown with known demand model) There is a line of work that views the monotonicity as a hard constraint, and aims to achieve high regret with uncertainty in the demand model, see, e.g. [Chen, 2021, Jia et al., 2021, Jia et al.,].

2 Formulation

In this section, we formally describe our model. We consider the problem of selling a product with unlimited supply on a platform over a continuous time horizon of length $T \in \mathbb{R}_+$. There is a pool of n unit-demand customers with heterogeneous valuations for the product. We assume that each customer's valuation belongs to the set $\{v_1, v_2, \ldots, v_k\}$, where $v_1 \geq v_2 \geq \ldots \geq v_k$. For each $i \in [k]$, there are n_i customers with valuation v_i, and $\sum_{i\in[k]} n_i = n$. At each time step the platform offers a take-it-or-leave-it price for the product. Customers are myopic and vary in their behavior in the following way. We assume that each customer interacts with the platform according to an independent Poisson process. Each time the customer interacts with the platform they monitor the price and purchase once the price is below their valuation yielding revenue equal to the price for the platform. The rate of each customer's Poisson process is $\lambda > 0$, which we refer to as the monitoring rate. The platform's objective is to maximize its total revenue. An instance of the problem is given by the tuple $\mathcal{I} = (T, n, \{n_i\}_{i=1}^k, \{v_i\}_{i=1}^k, \lambda)$ specifying all of the relevant information.

2.1 Policies, Revenue, and Benchmarks

(SJ: TO do:) In our context, a policy is a stochastic process $\pi = (X_t)$ specifying a price in the set $\{v_1, \ldots, v_k\}$. Before defining the revenue of a policy π, we introduce some additional notation. For the j'th customer, let $Y^j := (Y_t^j)$ denote their independent Poisson monitoring process. Let Z_t^j be the indicator function for the event that customer j has not made a purchase prior to time t. This process can be recursively defined as follows:

$$Z_{t+1}^j := Z_t^j \cdot (1 - \mathbf{1}(v_j \geq X_t) \cdot Y_t^j), \quad Z_1^j = 1.$$

Here we have slightly abused notation to let v_j be customer j's valuation. Thus the random revenue from customer j at time t under policy π is given by:

$$R_t^j := X_t \cdot \mathbf{1}(v_j \geq X_t) \cdot Y_t^j Z_t^j,$$

and the total revenue under policy π is given by:

$$R_\pi = \sum_{j\in[n]} \sum_{t\in[T]} R_t^j.$$

The expected revenue of a policy π under a given instance $\mathcal{I}$ is denoted as

$$\text{Rev}(\pi, \mathcal{I}) = \mathbb{E}_{\mathcal{I}}[R_\pi].$$

Here, we use the subscript $\mathcal{I}$ in the expectation operator to indicate that the expectation is taken over all randomness implied by the instance.

For a given class of policies Π, we denote the in-class optimum expected revenue as

$$\mathrm{OPT}(\mathcal{I}, \Pi) := \sup_{\pi \in \Pi} \mathrm{Rev}(\pi, \mathcal{I}).$$

Note that here the optimal in-class policy is given complete knowledge of the underlying instance $\mathcal{I}$. We will study the performance of policies that lack complete knowledge of the underlying instance and compare them to some benchmark policy that has this knowledge. The class of policies we will consider as a benchmark is the class of non-adaptive policies $\Pi_{\mathrm{NA}} = \{(x_1, x_2, \ldots, x_T) \mid \forall t \in [T], \exists i \in [k] \text{ such that } x_t = v_i\}$, specifying fixed sequences of prices. (SJ: Notation is not consistent with the non-adaptive part.) In Section 4 we discuss this class of policies further and characterize the optimal non-adaptive policy as being a markdown policy, i.e., one in which $x_t \geq x_{t'}$ for $t \leq t'$. When it is clear, we will drop the dependency on Π_{NA} and denote our benchmark as $\mathrm{OPT}(\mathcal{I}) := \mathrm{OPT}(\mathcal{I}, \Pi_{\mathrm{NA}})$.

3 Preliminaries

3.1 Adaptivity of Policy

In this section, we consider *non-adaptive policies*, i.e., policies where the price sequence and time at each price are determined in advance, regardless of the realized demands. We formally define it as follows.

Definition 3.1 (Non-adaptive Policy)**.** A policy $(X_s)_{s \in [0,T]}$ is *non-adaptive*, if there exist an integer $m > 0$ and two sequences $\mathbf{p} = (p_i)_{i \in [m]}$ and $\mathbf{t} = (t_i)_{i \in [m+1]}$ where $\mathbf{t}$ is non-decreasing, such that for any $s \in [t_i, t_{i+1})$ and $i \in [m]$ we have $X_s = p_i$ almost surely. Without loss of generality, we assume $t_1 = 0$ and $t_{m+1} = T$.

3.2 Known Demand

To further analyze the property of non-adaptive policy, we first derive a structural result of optimal non-adaptive policy. In Lemma 3.2, we show that the optimal non-adaptive policy is monotonically decreasing over time, which asserts that the optimal non-adaptive policy is a markdown policy.

Lemma 3.2 (Price Monotonicity)**.** *Suppose π is an optimal non-adaptive policy with price sequence $(p_1, p_2, \cdots, p_m)$. Then, $p_1 \geq p_2, \ldots, \geq p_m$.*

Essentially, this structural result follows from a swapping argument. Suppose we have a non-adaptive policy that deviates from being a markdown policy. In this scenario, we can increase its expected revenue as follows: Suppose the price is p_L in the interval $[t-\varepsilon, t]$ and rises to p_H in $[t, t+\varepsilon]$ for some $\varepsilon > 0$. We claim that exchanging p_H and p_L will result in a higher expected revenue. The reason is that customers with valuations lower than p_H remain unaffected since they cannot make purchases in the high-price interval. The customers impacted by the swap are those with valuations exceeding p_H. After the swap, customers with high valuations are more likely to purchase the product at the price p_H, leading to an increase in revenue for the non-adaptive policy. Using this swapping argument for any pair of adjacent prices, we conclude that the optimal non-adaptive pricing policy should be a markdown policy.(SJ: Make this para shorter)

We will subsequently restrict our attention to markdown policies. Suppose $m = k$, i.e., the non-adaptive policy can take as many prices as customers' valuations. Consequently, we will represent a policy as a sequence $(t_1, \cdots, t_{k+1})$, where $X_s = v_i$ for $s \in [t_i, t_{i+1}])$ and $i \in [k]$. In the following proposition, we establish the expected revenue expression for any non-adaptive markdown policy. Recall that for each $i \in [k]$, we denote R_i as the (random) revenue generated from a customer with valuation v_i.(SJ: OK)

Proposition 3.3 (Closed Form Expected Revenue)**.** *For any instance $\mathcal{I} = (T, n, \{n_i\}_{i=1}^k, \{v_i\}_{i=1}^k, \lambda)$, the expected total revenue of a non-adaptive markdown policy π is*

$$\begin{aligned}\mathrm{Rev}(\pi, \mathcal{I}) &= \sum_{i \in [k]} n_i \cdot \mathbb{E}[R_i] \\ &= \sum_{i \in [k]} n_i \sum_{j \in [k]: j \geq i} v_j e^{-\lambda(t_j - t_i)} \cdot \left(1 - e^{-\lambda(t_{j+1} - t_j)}\right).\end{aligned}$$

Note that each component within the inner summation corresponds to the expected revenue generated by a customer with valuation v_i during the j-th time interval. In more detail, the term $e^{-\lambda(t_j - t_i)}$ represents the probability that a customer of type i remains active in the market until time t_j, while the term $1 - e^{-\lambda(t_j - t_i)}$ represents the probability of the customer making a purchase during the j-th interval. Summing over all customer types, we derive the expected revenue for any non-adaptive markdown policy. (SJ: OK.)

Building upon the monotonicity result, we are now ready to outline a dynamic program approach for computing an optimal non-adaptive policy. At a high level, we encode the monotonicity of optimal non-adaptive policy into the Bellman equation and utilize an algorithm similar to the knapsack problem to compute the optimal price sequences.

Theorem 3.4 (Optimal Non-adaptive Policy)**.** *For any instance $\mathcal{I} = (T, n, \{n_i\}_{i=1}^k, \{v_i\}_{i=1}^k, \lambda)$, there exists an* *optimal non-adaptive policy, which can be computed with dynamic programming in polynomial time.*

So far we have seen that the problem is largely tractible if the demand was known. In the next two sections, we consider the scenario where the demand is unknown, focusing on non-adaptive and adaptive policies respectively.

4 Non-adaptive Policy

When dealing with known instances, we can efficiently compute an optimal non-adaptive policy using dynamic programming. However, in real-world scenarios, the demand for a product is often uncertain or known with certain prediction errors. We next show that there exists a non-adaptive policy that is guaranteed to earn a $1/k$ fraction of the optimal revenue under any valuation distribution with known, finite support. Formally, we have the following theorem.

Theorem 4.1 (Competitive Ratio Lower Bound)**.** *For any instance $\mathcal{I} = (T, n, \{n_i\}_{i=1}^k, \{v_i\}_{i=1}^k, \lambda)$ where $\{n_i\}_{i=1}^k$ is unknown to the seller, we can compute in polynomial time a nonadaptive policy $\pi = (t_1, \cdots, t_{k+1})$ such that*

$$\frac{\mathrm{Rev}(\pi, \mathcal{I})}{\mathrm{OPT}(\mathcal{I})} \geq \frac{1}{k}.$$

We first explain the idea for establishing this result. At first sight, one may attempt to solve the optimization problem of worst-case competitive ratio. However, a notable challenge arises due to the absence of a closed-form solution for the expected revenue generated by the optimal non-adaptive policy. To circumvent this, we adopt an alternative approach by considering an upper bound on the expected revenue of the optimal policy. More precisely, we define this upper bound as follows:

$$\mathrm{UB}(\mathcal{I}) := \sum_{i \in [k]} q_i v_i (1 - e^{-\lambda T}).$$

In the following lemma, we show that for any non-adaptive policy π, $UB(\mathcal{I})$ is always an upper bound of $\mathrm{Rev}(\pi, \mathcal{I})$.

Lemma 4.2. *For any instance $\mathcal{I} = (T, n, \{n_i\}_{i=1}^k, \{v_i\}_{i=1}^k, \lambda)$ and non-adaptive policy π,*

$$\mathrm{Rev}(\pi, \mathcal{I}) \leq \mathrm{UB}(\mathcal{I}).$$

To see why $\mathrm{UB}(\mathcal{I})$ is an upper bound, consider another price policy where for customers with valuation v_i, the seller offers them a price exactly as their valuation v_i. Note that $v_i(1 - e^{-\lambda T})$ is the largest revenue the seller can extract from customers with valuation v_i, which is the product of the highest price a customer with valuation v_i will accept and the monitoring probability in time T. No other price policy can achieve higher revenue. Therefore, $\mathrm{UB}(\mathcal{I})$ is an upper bound for any (possibly adaptive) policy.

Assuming $\{n_i\}_{i=1}^k$ is unknown in an instance. For a non-adaptive policy π, instead of computing the competitive ratio against the expected revenue of the optimal non-adaptive policy, we compute the competitive ratio against the upper bound $\mathrm{UB}(\mathcal{I})$. Then, the worst-case competitive ratio can be reduced to a robust optimization:

$$\max_{\pi \in \Pi_{NA}} \min_{\{n_i\}_{i=1}^k} \frac{\mathrm{Rev}(\pi, \mathcal{I})}{\mathrm{UB}(\mathcal{I})}.$$

However, this max-min problem is still hard to solve since $\mathrm{Rev}(\pi, \mathcal{I})$ and $\mathrm{UB}(\mathcal{I})$ are non-linear in time t. In order to provide a lower bound for the above ratio, we consider a linear surrogate of $\mathrm{Rev}(\pi, \mathcal{I})$ and $\mathrm{UB}(\mathcal{I})$. Without loss of generality, we assume $T = 1$. Define the linear surrogate of $\mathrm{UB}(\mathcal{I})$ and $\mathrm{Rev}(\pi, \mathcal{I})$ as,

$$UB(\mathcal{I}) \approx \mathrm{UB}'(\mathcal{I}) := \sum_{i \in [k]} q_i v_i \lambda,$$

$$Rev(\pi, \mathcal{I}) \approx \mathrm{Rev}'(\pi, \mathcal{I}) := \sum_{i \in [k]} q_i \sum_{j \in [k]: j \geq i} \lambda v_j \left(t_{j+1} - t_j\right).$$

In the following lemma, We show that any competitive ratio on the surrogate carries over the true competitive ratio.

Lemma 4.3 (Surrogate of Competitive Ratio)**.** *For any instance $\mathcal{I} = (T, n, \{n_i\}_{i=1}^k, \{v_i\}_{i=1}^k, \lambda)$ and non-adaptive policy π, we have*

$$\frac{\mathrm{Rev}(\pi, \mathcal{I})}{\mathrm{UB}(\mathcal{I})} \geq \frac{\mathrm{Rev}'(\pi, \mathcal{I})}{\mathrm{UB}'(\mathcal{I})}$$

To see why the above is true, observe that the function $h(x) := \frac{1-e^{-x}}{x}$ is decreasing in x. Thus, for any positive $x \leq y$, we have

$$\frac{1 - e^{-x}}{x} \geq \frac{1 - e^{-y}}{y},$$

i.e.,

$$\frac{1 - e^{-x}}{1 - e^{-y}} \geq \frac{x}{y}.$$

The result of Lemma 4.3 implies that any lower bound on the competitive ratio of the linear surrogates will be a lower bound of the true competitive ratio. Therefore, instead of optimizing the real competitive ratio, we consider the non-adaptive policy which maximizes the surrogate competitive ratio.

In particular, we reduce the max-min problem of the worst-case linear surrogate competitive ratio to:

$$\max_{\pi \in \Pi_{NA}} \min_{\{n_i\}_{i=1}^k} \frac{\sum_{i \in [k]} n_i \sum_{j \in [k]: j \geq i} v_j \left(t_{j+1} - t_j\right)}{\sum_i n_i v_i}.$$

(SJ: explain in words, why the inner min is easy to solve, from bilevel to single level.)

Note that the above optimization problem can be easily solved. In fact, the inner min is always obtained for one-hot vector, i.e., $n_{i^*} = n$, where $\frac{\sum_{j=i}^{k} v_j t_j}{v_i} \forall i \in [k]$ is minimized at i^*.

Thus, the above max-min problem can be reformulated as linear programming.

$$\begin{aligned} \max \quad & c \\ s.t. \quad & c \leq \frac{\sum_{j \in [k]: j \geq i} v_j \left(t_{j+1} - t_j\right)}{v_i}, \forall i \in [k], \\ & 0 \leq t_i \leq t_{i+1} \leq 1, \forall i \in [k]. \end{aligned}$$

One can easily verify that the optimal solution is attained when all the inequalities are binding. The optimal solution t_i^* satisfies,

$$t_{i+1}^* - t_i^* = \left(1 - \frac{v_{i+1}}{v_i}\right)(1 - t_k^*), \quad \forall i < k.$$

This solves to $t_k^* = 1 - \frac{1}{k - \sum_{i \in [k-1]} v_{i+1}/v_i}$. We denote the performance guarantees by $\rho(v_1, \cdots, v_k) = 1 - t_k^* = \frac{1}{k - \sum_{i=1}^{k-1} v_{i+1}/v_i} \geq \frac{1}{k}$.

So far, we have a performance guarantee for fixed $v_1, \cdots, v_k$. Next, we characterize the worst-case performance guarantee overall v_is, i.e., the worst-case competitive ratio. By simple calculation, one can verify that $\rho(v_1, \cdots, v_k)$ is at least $1/k$ for any $v_1, \cdots, v_k$.

Note that when k grows, the above result gets weaker and weaker. This motivates us to employ a core set of valuation distributions. More precisely, let $a > 0$ and b be the minimum and maximum of all the v_i respectively. For any $\epsilon > 0$, we can paritition the interval $[a, b]$ into "bucket" $[a(1+\varepsilon)^{j-1}, a(1+\varepsilon)^j)$ for $j = 1$ to $\rho = \log(b/a)/\log(1+\varepsilon)$. Then the competitive ratio will be at least $\frac{1}{\log(\rho)(1+\epsilon)}$.

Theorem 4.4 (Competitive Ratio Lower Bound)**.** *For any instance $\mathcal{I} = (T, n, \{n_i\}_{i=1}^k, \{v_i\}_{i=1}^k, \lambda)$ where $\{n_i\}_{i=1}^k$ is unknown to the seller, we can compute in polynomial time a nonadaptive policy $\pi = (t_1, \cdots, t_{k+1})$ such that*

$$\frac{\text{Rev}(\pi, \mathcal{I})}{\text{OPT}(\mathcal{I})} \geq \frac{1}{(1+\epsilon)\log\rho}.$$

4.1 Upper bound on competitive ratio

In this subsection, we show that the above lower bound $1/k$ turns out to be optimal. No algorithm can achieve a fraction larger than $1/k$ of the optimal revenue. In Theorem 4.5, we formally demonstrate that for any non-adaptive policy π, we can construct an instance, such that the policy can achieve at most $\frac{1}{k} + \epsilon$ fraction of the optimal revenue.

Theorem 4.5 (Upper Bound on Competitive Ratio)**.** *For any policy π, integer $k > 0$ and $\varepsilon > 0$, there exists an instance $\mathcal{I}_{\varepsilon,k}$ such that*

$$\frac{\mathbb{E}[\text{Rev}(\pi, \mathcal{I}_{\varepsilon,k})]}{\text{OPT}(\mathcal{I}_{\varepsilon,k})} \leq \frac{1}{k} + \epsilon.$$

5 Adaptive Policy

Now we consider adaptive policies in the presence of an unknown demand. Specifically, we assume only that the platform knows the time horizon T, the total number of customers n[1], the price levels, and the monitoring rate λ, but not the number of customers n_i at each price level. Our policies attempt to simultaneously learn the demand and optimize their pricing decisions given a finite time horizon. As is standard in the demand learning literature, we will analyze the *regret* of our policies. As stated before, we consider the optimal non-adaptive policy as our benchmark and denote its expected revenue under instance $\mathcal{I}$ as $\text{OPT}(\mathcal{I})$. Thus we may define the worst case regret as follows.

Definition 5.1 (Worst-Case Regret)**.** For a policy π, its worst-case regret is

$$\text{Regret}(\pi) := \sup_{\mathcal{I}} \left(\text{OPT}(\mathcal{I}) - \text{Rev}(\pi, \mathcal{I})\right)$$

where the supremum is taken over all instances $\mathcal{I}$ where the time horizon T, total number of customers n, price levels $\{v_i\}_{i=1}^k$, and monitoring rate λ are fixed. The demand at each price level n_i is allowed to vary arbitrarily subject to the constraint $\sum_{i \in [k]} n_i = n$.

Our goal is to find policies with low regret. To this end, the main result of this sections is as follows.

Theorem 5.2. *There exists an adaptive policy π^{LTE} which does not know the demand at each price level which satisfies* $\text{Regret}(\pi^{\text{LTE}}) = O(f(k) \cdot n^{3/4})$.[2]

Our policy is formally described below in Algorithm 1. The policy operates in two phases. In the first phase, we briefly explore each of the price levels $v_1, v_2, \ldots, v_{k-1}$ for fixed intervals of length $s_1, s_2, \ldots, s_{k-1}$. While exploring the i'th price level, we keep track of the realized demand D_i. We then use the values $\{D_i\}_{i=1}^{k-1}$ to construct estimates $\{\hat{n}_i\}_{i=1}^k$ of the original demand at each price level. From there we construct a "pseudo-instance" $\hat{\mathcal{I}} = (T - \sum_i s_i, n, \{\hat{n}_i\}_{i=1}^k, \{v_i\}_{i=1}^k, \lambda)$ and compute an optimal non-adaptive policy for this instance. By construction this policy is of length $T' = T - \sum_{i<k} s_i$ and is feasible for the remaining instance.

[1] It is enough to have a good estimate of n, say within XXX additive error. TODO: Analyze what happens when n is mis-specified and fill in something for XXX.

[2] TODO: Determine the $f(k)$ needed for the analysis to go through and fill this in

In order to show that this policy has low regret, first we must show that we may take $s_1, s_2, \ldots, s_{k-1}$ small enough so that the exploration period incurs low regret. Next, we need to show that the estimates $\{\hat{n}_i\}_{i=1}^k$ are good enough so that the optimal policy on the "pseudo-instance" $\hat{\mathcal{I}}$ gives us close to the optimal revenue in expectation. To demonstrate the key ideas in our analysis, in the next section we first restrict to the case of $k = 2$ different price levels.

Algorithm 1: Adaptive Learn-then-Earn Policy π^{LTE}

```
Data: Partial Instance $(T, n, \{v_i\}_{i=1}^k, \lambda)$,
      Exploration times $(s_1, s_2, \ldots, s_{k-1})$
Result: Policy $\pi^{\text{LTE}}$
for $i = 1, 2, \ldots, k-1$ do
    Use price $v_i$ for time $s_i$
    Observe sales $D_i$
end
//Construct estimates $\{\hat{n}_i\}_{i=1}^k$
for $i = 1, 2, \ldots, k-1$ do
    $\hat{n}_i \leftarrow \frac{D_i}{q(s_i)} - \sum_{j<i}(\hat{n}_j - D_j)$
end
$\hat{n}_k \leftarrow n - \sum_{i<k} \hat{n}_i$
$\hat{\mathcal{I}} \leftarrow (T - \sum_{i<k} s_i, n, \{\hat{n}_i\}_{i=1}^k, \{v_i\}_{i=1}^k, \lambda)$
Compute an optimal non-adaptive policy $\hat{t}_1, \ldots, \hat{t}_k$
  for the instance $\hat{\mathcal{I}}$
for $i = 1, 2, \ldots, k$ do
    Use price $v_i$ for time $\hat{t}_i$
end
```

5.1 The Case of Two Price Levels

In the case that there are only $k = 2$ price levels $v_1 > v_2$, our proposed policy simplifies greatly. First we specify an exploration time $s \in [0, T]$, then set the price $X_t = v_1$ for all $t \leq s$. Let D be the demand (number of sales) observed in this period. We construct an estimate $\hat{n}_1$ of the number of customers with valuation v_1. We do this based on the following observation. Let $q(s) = 1 - \exp(-\lambda s)$ be the probability that a customer monitors the platform at least once in the interval $[0, s]$. Then we have $D \sim \text{Binomial}(n_1, q(s))$, and so $\hat{n}_1 = D/q(s)$ is an unbiased estimate of n_1. Since $n_1 + n_2 = n$, it follows that $\hat{n}_2 = n - \hat{n}_1$ is an unbiased estimate of n_2 as well. Using these estimates, we construct the "pseudo-instance" $\hat{\mathcal{I}} = (T - s, n - D, \{\hat{n}_1 - D, \hat{n}_2\}, \{v_1, v_2\}, \lambda))$ and compute a non-adaptive policy achieving expected revenue $\text{OPT}(\hat{\mathcal{I}})$ for the remaining time horizon. We then follow this policy for the remaining time horizon.

(Proof sketch) choose the exploration period to be $n^{-1/4}$. We then show that this results in an estimation $\hat{n}_1, \hat{n}_2$ such that

$$|n_1 - \hat{n}_1| \leq n^{3/4}.$$

To see this, note that the observed demand at the end of the exploration period follows a Binomial distribution $\text{Bin}(n_1, 1 - e^{-\lambda s})$. By Taylor expansion, for small s we have $1 - e^{-\lambda s} \sim q(s) := \lambda s$. Applying standard concentration bounds to $\text{Bin}(n_1, q)$, we obtain

$$|n_1 - \hat{n}_1| \leq \frac{\sqrt{n}}{q}$$

with probability n^{-10}. The $n^{3/4}$ error then immediately follows by choosing $s = n^{-1/4}$.

Our goal is to show that the worst-case regret of this policy is $\tilde{O}(n^{3/4})$. This policy incurs regret in a few ways. One way we incur regret is due to exploration and using a smaller time horizon for the pseudo-instance. Another way we incur regret is due to estimation error in $\hat{n}_1$ and $\hat{n}_2$ affecting the revenue we get in the "earning" phase of the algorithm. We will show that both of these contributions are small.

Proceeding with the formal analysis, we first define and characterize the expected revenue of an optimal non-adaptive policy.

(SJ: Note: possibly we need to move this up front, to Section 2)

Definition 5.3. For $s \in [0, T]$ let $R(s; \mathcal{I})$ be the expected revenue of the non-adaptive policy which uses price v_1 for s steps then price v_2 for $T - s$ steps on instance $\mathcal{I} = (T, n, \{n_1, n_2\}, \{v_1, v_2\}, \lambda)$.

(SJ: Lower bound analysis also uses this result)

We have the following clsoed-form formula for the expected revenue. Recall that $q(s) = 1 - \exp(-\lambda s)$.

Proposition 5.4 (Optimal Revenue)**.** *We have* $\text{OPT}(\mathcal{I}) = \max_{s \in [0,T]} R(s; \mathcal{I})$, *and for any* $s \in [0, T]$,

$$R(s; \mathcal{I}) = n_1 v_1 q(s) + n_1 v_2 (1 - q(s)) q(T - s) + n_2 v_2 q(T - s).$$

Proof. The first statement follows from the definition of $\text{OPT}(\mathcal{I})$ and the fact that the optimal non-adaptive policy can be taken in the class of policies defined by pairs $(s, T - s)$ in which we use price v_1 for time $s \in [0, T]$ and price v_2 for time $T - s$. The expression for $R(s; \mathcal{I})$ follows by considering the expected revenue from a customer with each valuation type then using linearity of expectation. □

To analyze the regret of our proposed policy, we first need to decompose the regret into individual terms that are tractable to analyze. Let $\hat{s}(D)$ be the random amount of time we remain at price v_1 after the exploration phase, i.e., $\hat{s}(D) = \arg\max_{s \in [0, T-s]} R(s; \hat{\mathcal{I}})$.

Additionally, we define two "intermediate" instances $\mathcal{I}'$ and $\mathcal{I}''$, that we make use of in the analysis. We define $\mathcal{I}' := (T-s, n-D, \{n_1-D, n_2\}, \{v_1, v_2\}, \lambda)$ to be the remaining instance after the exploration period. Note that this instance is random since it depends on the random demand D observed in the exploration period. We also define $\mathcal{I}'' := (T-s, n, \{n_1, n_2\}, \{v_1, v_2\}, \lambda)$, which is almost identical to the original instance except that the time horizon is shortened to $T-s$. The following lemma decomposes the regret of our policy in terms of the quantities defined above.

Lemma 5.5 (Regret Decomposition). *For the proposed policy π^{LTE}, we have that*

$$\text{Regret}(\pi^{\text{LTE}}) \leq \eta_1 + \eta_2 + \eta_3$$

where η_1, η_2, and η_3 are defined as follows:

- *(Optimization)* $\eta_1 := |\mathbb{E}[R(\hat{s}; \mathcal{I}')] - \mathbb{E}[\text{OPT}(\hat{\mathcal{I}})]|$
- *(Estimation)* $\eta_2 := |\mathbb{E}[\text{OPT}(\hat{\mathcal{I}})] - \mathbb{E}[\text{OPT}(\mathcal{I}'')]|$
- *(Exploration)* $\eta_3 := |\mathbb{E}[\text{OPT}(\mathcal{I}'') - \text{OPT}(\mathcal{I})]|$

Proof Sketch. Expand the regret of our policy as a telescoping sum involving the terms above, then apply the triangle inequality. □

Continuing with the analysis, we will analyze each of the error terms defined above to derive a regret bound in terms of the length of the exploration period s. Optimizing the choice of s will yield the regret bound stated in Theorem 5.2.

We first show that $\eta_1 = 0$. This is essentially because we use unbiased estimates and the error is linear in $\hat{n}_i - n_i$.

Lemma 5.6 (Analysis of η_1). *For our policy π^{LTE}, we have $\eta_1 = 0$.*

Proof. First we write $\mathbb{E}[R(\hat{s}; \mathcal{I}')] - \mathbb{E}[\text{OPT}(\hat{\mathcal{I}})]$ as $(\mathbb{E}[R(\hat{s}; \mathcal{I}')] - \mathbb{E}[R(\hat{s}; \hat{\mathcal{I}})]) + (\mathbb{E}[R(\hat{s}; \hat{\mathcal{I}})] - \mathbb{E}[\text{OPT}(\hat{\mathcal{I}})])$ and we will show that each of these terms is 0. For the first term, we condition on $\hat{s} = s$ and consider the conditional expectation $\mathbb{E}[R(\hat{s}; \mathcal{I}') - R(\hat{s}; \hat{\mathcal{I}}) \mid \hat{s} = s]$. For all $s \in [0, T]$, we have

$$\begin{aligned} R(s; \mathcal{I}') - R(s; \hat{\mathcal{I}}) &= (n_1 - D - (\hat{n}_1 - D)) v_1 q(s) \\ &\quad + (n_1 - D - (\hat{n}_1 - D)) v_2 (1 - q(s)) q(T - s) \\ &\quad + (n_2 - \hat{n}_2) v_2 q(T - s) \end{aligned}$$

Thus the conditional expectation above is 0. Applying the law of total expectation implies the first term is 0. For the second term, this immediately follows from the definition of $\hat{s}$ since it witnesses an optimal non-adaptive policy for the instance $\hat{\mathcal{I}}$. □

Lemma 5.7 (Analysis of η_2). *For our policy π^{LTE}, we have $\eta_2 \leq XXX$*

Proof. Proof:

□

References

[Abbasi-Yadkori et al., 2011] Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24.

[Auer, 2002] Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422.

[Auer et al., 2002] Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77.

[Aviv and Pazgal, 2008] Aviv, Y. and Pazgal, A. (2008). Optimal pricing of seasonal products in the presence of forward-looking consumers. *Manufacturing & service operations management*, 10(3):339–359.

[Aviv et al., 2019] Aviv, Y., Wei, M. M., and Zhang, F. (2019). Responsive pricing of fashion products: The effects of demand learning and strategic consumer behavior. *Management Science*, 65(7):2982–3000.

[Babaioff et al., 2015] Babaioff, M., Dughmi, S., Kleinberg, R., and Slivkins, A. (2015). Dynamic pricing with limited supply.

[Bastani et al., 2022] Bastani, H., Simchi-Levi, D., and Zhu, R. (2022). Meta dynamic pricing: Transfer learning across experiments. *Management Science*, 68(3):1865–1881.

[Besbes and Zeevi, 2009] 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 and Zeevi, 2015] 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.

[Borgs et al., 2014] Borgs, C., Candogan, O., Chayes, J., Lobel, I., and Nazerzadeh, H. (2014). Optimal multiperiod pricing with service guarantees. *Management Science*, 60(7):1792–1811.

[Briceño-Arias et al., 2017] Briceño-Arias, L., Correa, J. R., and Perlroth, A. (2017). Optimal continuous pricing with strategic consumers. *Management Science*, 63(8):2741–2755.

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

[Chen et al., 2022a] Chen, B., Simchi-Levi, D., Wang, Y., and Zhou, Y. (2022a). Dynamic pricing and inventory control with fixed ordering cost and incomplete demand information. *Management Science*, 68(8):5684–5703.

[Chen, 2021] Chen, N. (2021). Multi-armed bandit requiring monotone arm sequences. *Advances in Neural Information Processing Systems*, 34:16093–16103.

[Chen et al., 2023] Chen, X., Miao, S., and Wang, Y. (2023). Differential privacy in personalized pricing with nonparametric demand models. *Operations Research*, 71(2):581–602.

[Chen et al., 2022b] Chen, X., Simchi-Levi, D., and Wang, Y. (2022b). Privacy-preserving dynamic personalized pricing with demand learning. *Management Science*, 68(7):4878–4898.

[Cohen et al., 2021] Cohen, M. C., Perakis, G., and Pindyck, R. S. (2021). A simple rule for pricing with limited knowledge of demand. *Management Science*, 67(3):1608–1621.

[Correa et al., 2016] Correa, J., Montoya, R., and Thraves, C. (2016). Contingent preannounced pricing policies with strategic consumers. *Operations Research*, 64(1):251–272.

[Correa et al., 2020] Correa, J. R., Pizarro, D., and Vulcano, G. (2020). The value of observability in dynamic pricing. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 275–290.

[Dasu and Tong, 2010] Dasu, S. and Tong, C. (2010). Dynamic pricing when consumers are strategic: Analysis of posted and contingent pricing schemes. *European Journal of Operational Research*, 204(3):662–671.

[den Boer, 2015] 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.

[den Boer and Keskin, 2020] den Boer, A. V. and Keskin, N. B. (2020). Discontinuous demand functions: Estimation and pricing. *Management Science*, 66(10):4516–4534.

[den Boer and Zwart, 2013] den Boer, A. V. and Zwart, B. (2013). Simultaneously learning and optimizing using controlled variance pricing. *Management science*, 60(3):770–783.

[den Boer and Zwart, 2015] den Boer, A. V. and Zwart, B. (2015). Dynamic pricing and learning with finite inventories. *Operations research*, 63(4):965–978.

[Deng et al., 2023] Deng, Y., Mao, J., Sivan, B., and Wang, K. (2023). Optimal pricing schemes for an impatient buyer. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 382–398. SIAM.

[Elmaghraby and Keskinocak, 2003] Elmaghraby, W. and Keskinocak, P. (2003). Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management science*, 49(10):1287–1309.

[Filippi et al., 2010] Filippi, S., Cappe, O., Garivier, A., and Szepesvári, C. (2010). Parametric bandits: The generalized linear case. *Advances in neural information processing systems*, 23.

[Jia et al., 2021] Jia, S., Li, A., and Ravi, R. (2021). Markdown pricing under unknown demand. *Available at SSRN 3861379.*

[Jia et al.,] Jia, S., Li, A. A., and Ravi, R. Dynamic pricing with monotonicity constraint under unknown parametric demand model. In *Advances in Neural Information Processing Systems.*

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

[Kleinberg and Leighton, 2003] 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.*, pages 594–605. IEEE.

[Li et al., 2014] Li, J., Granados, N., and Netessine, S. (2014). Are consumers strategic? structural estimation from the air-travel industry. *Management Science*, 60(9):2114–2137.

[Li et al., 2023] Li, Y., Wang, Y., and Zhou, Y. (2023). Nearly minimax-optimal regret for linearly parameterized bandits. *IEEE Transactions on Information Theory.*

[Lobel, 2020] Lobel, I. (2020). Dynamic pricing with heterogeneous patience levels. *Operations Research*, 68(4):1038–1046.

[Moon et al., 2018] Moon, K., Bimpikis, K., and Mendelson, H. (2018). Randomized markdowns and online monitoring. *Management Science*, 64(3):1271–1290.

[Robbins, 1952] Robbins, H. (1952). Some aspects of the sequential design of experiments.

[Rusmevichientong and Tsitsiklis, 2010] Rusmevichientong, P. and Tsitsiklis, J. N. (2010). Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411.

[Segal, 2003] Segal, I. (2003). Optimal pricing mechanisms with unknown demand. *American Economic Review*, 93(3):509–529.

[Shi et al., 2016] Shi, C., Chen, W., and Duenyas, I. (2016). Nonparametric data-driven algorithms for multiproduct inventory systems with censored demand. *Operations Research*, 64(2):362–370.

[Su, 2007] Su, X. (2007). Intertemporal pricing with strategic customer behavior. *Management Science*, 53(5):726–741.

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

[Wang et al., 2021] Wang, Y., Chen, B., and Simchi-Levi, D. (2021). Multimodal dynamic pricing. *Management Science*, 67(10):6136–6152.

[Wang et al., 2014] Wang, Z., Deng, S., and Ye, Y. (2014). Close the gaps: A learning-while-doing algorithm for single-product revenue management problems. *Operations Research*, 62(2):318–331.

[Yuan et al., 2021] Yuan, H., Luo, Q., and Shi, C. (2021). Marrying stochastic gradient descent with bandits: Learning algorithms for inventory systems with fixed costs. *Management Science*, 67(10):6089–6115.