Dynamic Pricing and Demand Learning with Limited Price Experimentation

Wang Chi Cheung
Institute of High Performance Computing, Agency of Science, Technology And Research, Singapore, cheungwc@ihpc.a-star.edu.sg

David Simchi-Levi
Department of Civil and Environmental Engineering, Institute for Data, Systems, and Society, Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, dslevi@mit.edu

He Wang
School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, he.wang@isye.gatech.edu

In a dynamic pricing problem where the demand function is not known a priori, price experimentation can be used as a demand learning tool. Existing literature usually assumes no constraint on price changes, but in practice sellers often face business constraints that prevent them from conducting extensive experimentation. We consider a dynamic pricing model where the demand function is unknown but belongs to a known finite set. The seller is allowed to make at most \( m \) price changes during \( T \) periods. The objective is to minimize the worst case regret, i.e. the expected total revenue loss compared to a clairvoyant who knows the demand distribution in advance. We demonstrate a pricing policy that incurs a regret of \( O(\log^m T) \), or \( m \) iterations of the logarithm. Furthermore, we describe an implementation of this pricing policy at Groupon, a large e-commerce marketplace for daily deals. The field study shows significant impact on revenue and bookings.

Key words: revenue management, dynamic pricing, learning-earning tradeoff, price experimentation

1. Introduction

This paper considers a general learning and pricing problem motivated by the daily deal website Groupon. Groupon is a large e-commerce marketplace where customers can purchase discount deals from local merchants such as restaurants, spas and house cleaning services. Every day, many new deals are launched on Groupon’s website. Due to its business model, Groupon is faced with high level of demand uncertainty, mainly because there is no previous sale for the newly launched deals that can be used for demand forecasting. This challenge presents an opportunity for Groupon to learn about customer demand using real time sales data after deals have been launched so as to obtain more accurate demand estimation and adjust prices.

Generally, in revenue management, when the underlying relationship between demand and price is unknown a priori, the seller can use price experimentation for demand learning.
In this paper, we consider a dynamic pricing model where the exact demand function is unknown but belongs to a finite set of possible demand functions, or demand hypotheses. The seller faces an exploration-exploitation tradeoff between actively adjusting price to gather demand information and optimizing price for revenue maximization.

Dynamic pricing under a finite set of demand hypotheses has previously been considered by Rothschild (1974) and Harrison et al. (2012). But unlike the current paper, both of these papers focus on customized pricing, where price is changed for every arriving customer. For example, the motivation of Harrison et al. (2012) is pricing for financial services such as consumer and auto loans, where sellers can quote a different interest rate for each customer. However, for many e-commerce companies like Groupon, charging a different price for each arriving customer is impossible, either because of implementation constraints, or for fear of confusing customers and receiving negative customer response. In our collaboration, Groupon stipulated as a rule that the number of price changes has to be as few as possible for each deal, so that the customers would not observe frequent price changes. Motivated by this practical business constraint on price experimentation, the model in this paper includes an explicit constraint on the number of price changes during the sales horizon. We quantify the impact of this constraint on the seller’s revenue using regret, defined as the gap between the revenue of a clairvoyant who has full information on the demand function and the revenue achieved by a seller facing unknown demand.

Our main finding is a characterization of regret as a function of the number of price changes allowed. When there are $T$ periods in the sales horizon, we propose a pricing policy with at most $m$ price changes, whose regret is bounded by $O(\log^m T)$, or $m$ iterations of the logarithm.

A natural question is how frequently one needs to change price to achieve a constant regret. Harrison et al. (2012) shows that a semi-myopic policy can achieve a constant regret, but the policy requires changing price for every time period. To answer this question, we show that a modified version of our algorithm with no more than $O(\log^* T)$ price changes, where $\log^* T$ is the smallest number $m$ such that $\log^m T \leq 1$, achieves a constant regret.

This characterization of the regret bound shows that the incremental effect of price changes decreases quickly. The first price change reduces regret from $O(T)$ to $O(\log T)$; each additional price change thereafter compounds a logarithm to the order of regret. As a result, the first few price changes generate most of the benefit of dynamic pricing.
Interestingly, while the value $\log^* T$ is unbounded when $T$ increases, its growth rate is extremely slow. For example, if the number of time periods $T$ is less than 3 million, we still have $\log^* T$ no greater than 3.

Motivated by these results, we implemented a pricing strategy at Groupon where each deal can have at most one price change. The result of a field experiment shows significant improvement in both revenue and deal bookings at Groupon.

2. Related Literature

Joint learning-and-pricing problems have received extensive research attention over the last decade. Recent surveys by Aviv and Vulcano (2012) and den Boer (2015) provide a comprehensive overview of this area. Some papers that consider price experimentation for learning demand curves include Besbes and Zeevi (2009), Boyaci and Özer (2010), Wang et al. (2014) and Besbes and Zeevi (2015). These problems typically focus on the tradeoffs between learning and earning, which is closely related to the multi-armed bandit literature (e.g. Kleinberg and Leighton 2003, Mersereau et al. 2009, Rusmevichientong and Tsitsiklis 2010).

Recently, a stream of papers focuses on semi-myopic pricing policies using various learning methods. Examples include maximum likelihood estimation (Broder and Rusmevichientong 2012), Bayesian methods (Harrison et al. 2012), maximum quasi-likelihood estimation (den Boer and Zwart 2014, den Boer 2014) and iterative least-squares estimation (Keskin and Zeevi 2014).

Unlike the current paper, all the literature mentioned above does not assume any constraint on price experimentation. In the dynamic pricing literature with known demand distribution, several papers consider limited price changes; examples include Feng and Gallego (1995), Bitran and Mondschein (1997), Netessine (2006) and Chen et al. (2015). Caro and Gallien (2012) reports that the fashion retailer Zara uses a clearance pricing policy with a pre-determined price ladder, which essentially allows for only a limited number of mark-down prices. Zbaracki et al. (2004) provide empirical results on the cost of price changes.

To the best of our knowledge, the only work that considers price-changing constraints in an unknown demand setting is Broder (2011). The author assumes that the demand function belongs to a known parametric family, e.g. linear family, but has unknown parameters. He shows that in order to achieve the optimal regret, a pricing policy needs at least
\(\Theta(\log T)\) price changes. However, the result only applies to a restricted class of policies where the seller cannot use any knowledge of \(T\).

Our model is different from the model by Broder (2011) in the following aspects. First, we assume a finite number of demand hypotheses, while Broder (2011) assumes a parametric family of demand functions. This is a fundamental difference because the optimal regret in Broder’s case is \(\Theta(\sqrt{T})\), while in our case the regret can be bounded by a constant. Second, we do not assume a restricted class of policies as in Broder (2011), and our results hold for any pricing policies. Last but not least, unlike Broder (2011) where the number of price changes is an output from the model, we design a pricing algorithm that accepts the number of price changes as an input constraint, and achieves the best possible regret bound under that constraint.

3. Problem Formulation

We consider a seller offering a single product with unlimited supply for \(T\) periods. The set of allowable prices is denoted by \(\mathcal{P}\). For example, \(\mathcal{P}\) can be an interval \([p, \overline{p}]\) or a finite set \(\{p_1, \ldots, p_k\}\), although no restriction on \(\mathcal{P}\) is assumed.

In the \(t^{th}\) period \((t = 1, \ldots, T)\), the seller offers a unit price \(P_t \in \mathcal{P}\). Then, she observes the realized customer demand \(X_t\), i.e. the number of units purchased in the \(t^{th}\) period. Given \(P_t = p\), the distribution of \(X_t\) is only determined by price \(p\), and is independent of previous prices and demands \(\{P_1, X_1, \ldots, P_{t-1}, X_{t-1}\}\). We use \(D(p)\) to denote a generic random variable for demand given price \(p\). Thus, given \(P_t = p\), we have \(X_t \sim D(p)\). The corresponding mean demand function \(d : \mathcal{P} \rightarrow \mathbb{R}_+\) is defined as \(d(p) = \mathbb{E}[D(p)]\).

The distribution of \(D(p)\) is unknown to the seller. However, the seller knows that the mean demand function \(d(p)\) is equal to one of the \(K\) given mean demand functions: \(d_1(p), \ldots, d_K(p)\). We call the collection \(\Phi = \{d_1(\cdot), \ldots, d_K(\cdot)\}\) the demand hypothesis set. The seller does not necessarily know the distribution associated with each demand hypothesis \(i\) (\(\forall i = 1, \ldots, K\)) apart from the mean \(d_i(p)\). For each demand function \(d_i \in \Phi\), the expected revenue per period is denoted by \(r_i(p) = pd_i(p)\). We also denote the optimal revenue for demand function \(d_i\) by \(r_i^* = \max_{p \in \mathcal{P}} r_i(p)\) and an optimal price by \(p_i^* \in \text{arg max}_{p \in \mathcal{P}} r_i(p)\).

For all \(p \in \mathcal{P}\), given the mean demand function \(d_i\), the probability distribution of \(D(p)\) is assumed to be light-tailed with parameters \((\sigma, b)\), where \(\sigma, b > 0\). That is, we have
\[\mathbb{E}_i[e^{\lambda(D(p) - d_i(p))}] \leq \exp(\lambda^2\sigma^2/2)\] for all \(|\lambda| < 1/b\). Note that the class of light-tailed distributions includes all sub-Gaussian distributions. Some common light-tailed distributions include normal, Poisson and Gamma distributions, as well as all distributions with bounded support, such as binomial and uniform distributions.

### 3.1. Pricing Policies

We say that \(\pi\) is a non-anticipating pricing policy if the price \(P^\pi_t\) offered by \(\pi\) at period \(t\) is determined by the time horizon \(T\) and the sales history \(H_t\). History \(H_t\) consists of the realized demand \((X_1, \ldots, X_{t-1})\), the previous offered prices \(P^\pi_1, \ldots, P^\pi_{t-1}\), as well as the length of sales horizon \(T\). Importantly, \(P^\pi_t\) is independent of future demand. We express the dependence of \(P^\pi_t\) on the sales history in the following:

\[
P^\pi_t = \pi(X_1, \ldots, X_{t-1}; T).
\]

Note that we make the dependence on the previous offered prices \(P^\pi_1, \ldots, P^\pi_{t-1}\) implicit in the expression, as they are determined by the purchase decision \(X_1, \ldots, X_{t-1}\) and \(T\). Additional explanation is provided in Appendix A. The price offered in period 1, \(P^\pi_1 = \pi(0; T)\), is only determined only by \(\pi\) and \(T\), in the absence of sales history.

For \(i = 1, \ldots, K\), let \(\mathbb{P}^\pi_i[\cdot]\) and \(\mathbb{E}^\pi_i[\cdot]\) be the probability measure and expectation induced by policy \(\pi\) if the underlying demand model is \(i\). In this case, the seller’s expected revenue in \(T\) periods under policy \(\pi\) is given by

\[
R^\pi_i(T) = \mathbb{E}^\pi_i \left[ \sum_{t=1}^T P_t X_t \right] = \mathbb{E}^\pi_i \left[ \sum_{t=1}^T P_t \mathbb{E}^\pi_i [X_t | P_t] \right] = \mathbb{E}^\pi_i \left[ \sum_{t=1}^T r_i(P_t) \right].
\]

As motivated earlier, in many revenue management applications, the seller faces a constraint on the number of price changes. In the model, we assume that the seller can make at most \(m\) changes to the price over the course of the sales event, where \(m\) is a fixed integer. A feasible policy \(\pi\) should therefore satisfy the following condition:

\[
\mathbb{P}^\pi_i \left[ \sum_{t=2}^T I(P_t \neq P_{t-1}) \leq m \right] = 1, \quad \forall i = 1, \ldots, K,
\]

where \(I(\cdot)\) is the indicator function. We refer to a policy with at most \(m\) price changes as an \(m\)-change policy.

The performance of a pricing policy is measured against the optimal policy in the full information case. If the true demand is \(d_i\), then a clairvoyant with full knowledge of the
demand function would offer price $p_i^*$ and obtain expected revenue $r_i^*$ for every period. The *regret* with respect to demand $d_i$ is defined as the gap between the expected revenue achieved by the clairvoyant and the one achieved by policy $\pi$, namely

$$\text{Regret}_i^\pi(T) = T r_i^* - \mathbb{E}_i^\pi \left[ \sum_{t=1}^T (r_i^* - r_i(P_t)) \right].$$

Finally, we define the minimax regret for the demand set, $\Phi = \{d_1, \ldots, d_K\}$, as

$$\text{Regret}_{\Phi}^\pi(T) = \max_{i=1,\ldots,K} \text{Regret}_i^\pi(T).$$

When there is no ambiguity of which policy we are referring to, we suppress the superscript “$\pi$” in the notation for clarity, namely $\mathbb{E}_1 := \mathbb{E}_1^\pi$, $P_1 := P_1^\pi$.

### 3.2. Notations

For two sequences $\{a_n\}$ and $\{b_n\}$ ($n = 1, 2, \ldots$), we write $a_n = O(b_n)$ if there exists a constant $C$ such that $a_n \leq C b_n$ for all $n$; we write $a_n = \Omega(b_n)$ if there exists a constant $c$ such that $a_n \geq c b_n$ for all $n$. We use $\log^{(m)} T$ to represent $m$ iterations of the logarithm, $\log(\log(\ldots \log(T)))$, where $m$ is the number of price changes. For convenience, we let $\log(x) = 0$ for all $0 \leq x < 1$, so the function $\log^{(m)} T$ is defined for all $T \geq 1$. Similarly, we define the notations $e^{(0)}(T) := T$, and $e^{(\ell)}(T) := \exp(e^{(\ell-1)}(T))$ for $\ell \geq 1$. We also define the short hand notation $e^{(\ell)} := e^{(\ell)}(1)$. As mentioned earlier, function $\log^* T$ denotes the smallest nonnegative integer $m$ such that $\log^{(m)} T \leq 1$. For any real number $x$, we denote by $\lceil x \rceil$ the minimum integer greater than or equal to $x$. For any finite set $S$, the cardinality of $S$ is denoted by $|S|$. We occasionally use notations $a \vee b = \max\{a, b\}$, $a \wedge b = \min\{a, b\}$.

### 4. Main Results: Upper Bounds on Regret

In this section we prove the main results of the paper: an upper bound on the regret as a function of the number of price changes. We design a non-anticipating pricing policy that changes price no more than $m$ times and achieves a regret of $O(\log^{(m)} T)$.

#### 4.1. Upper Bound

We propose a policy $mPC$ (which stands for “$m$-price change”) that achieves a regret of $O(\log^{(m)} T)$ with at most $m$ price changes. An important feature of policy $mPC$ is that it applies a *discriminative* price for every period. A price $p$ is *discriminative* if demands $d_1(p), \ldots, d_K(p)$ are mutually distinct.

We make the following assumption on the set of demand functions $\Phi$:
Assumption 1. For all $d_i \in \Phi = \{d_1, \ldots, d_K\}$, there exists a corresponding revenue-optimal price $p_i^* \in \arg\max_{p \in P} r_i(p)$ such that $p_i^*$ is a discriminative price for $\Phi$, that is, $d_1(p_i^*), \ldots, d_K(p_i^*)$ are distinct. Moreover, such price $p_i^*$ can be efficiently computed.

Assumption 1 ensures that the seller is able to learn the underlying demand curve while maximizing its revenue for any given demand function $d_i \in \Phi$. In fact, we will show in Section 4.3 that this condition is both sufficient and necessary for achieving a regret bound better than $o(\log T)$.

**Algorithm 1** $m$-change policy $mPC$

1: INPUT:
   - A set of demand functions $\Phi = \{d_1, \ldots, d_K\}$.
   - A discriminative price $P_0^*$.
2: (Learning) Set $\tau_0 = 0$.
3: for $\ell = 0, \ldots, m - 1$ do
4:   if $\log^{(m-\ell)}T = 0$ then
5:     Set $\tau_{\ell+1} = 0$ and $P_{\ell+1}^* = P_\ell^*$.
6:   else
7:     From period $\tau_\ell + 1$ to $\tau_{\ell+1} := \tau_\ell + \left[ M_\Phi(P_\ell^*) \log^{(m-\ell)}T \right]$, set the offered price as $P_\ell^*$.
8:     At the end of period $\tau_{\ell+1}$, compute the sample mean $\bar{X}_\ell$ from period $\tau_\ell + 1$ to $\tau_{\ell+1}$:
9:       $\bar{X}_\ell := \frac{\sum_{j=\tau_\ell+1}^{\tau_{\ell+1}} X_j}{\tau_{\ell+1} - \tau_\ell}$, where $X_j =$ Number of items sold in period $j$.
10: Choose an index $i_\ell \in \{1, \ldots, K\}$ which solves
11:     $$\min_{i \in \{1, \ldots, K\}} |\bar{X}_\ell - d_i(P_\ell^*)|.$$ 
12: Set the next offered price as $P_{\ell+1}^* = p_{i_\ell}^*$, where $p_{i_\ell}^*$ is the optimal price for demand $d_{i_\ell}$.
13: end if
14: end for
15: (Earning) From period $\tau_m + 1$ to period $\tau_{m+1} = T$, set the selling price as $P_m$. 
Algorithm 1 describes the mPC policy. The policy partitions the finite time horizon \(1, \cdots, T\) into \(m+1\) phases. For each \(0 \leq \ell \leq m\), a single price \(P^*_\ell\) is offered through Phase \(\ell\), which starts at period \(\tau_\ell + 1\) and ends at \(\tau_{\ell+1}\). Phase 0 to Phase \(m-1\) are called the learning phases, and Phase \(m\) is referred to as the earning phase. Except for a constant factor \(M_\Phi(P^*_0)\), which is to be defined later, the lengths of phases are iterated-exponentially (tetrationally) increasing, which ensures an optimal balance between exploration and exploitation.

At the end of learning phase \(\ell\), policy mPC computes the sample mean \(\bar{X}_\ell\) of the sales under price \(P^*_\ell\) (in line 8 of the algorithm). Since price \(P^*_\ell\) is discriminative, the seller gains new information about the underlying demand in this learning phase. She then updates her belief on the true demand distribution to be \(d_{i,\ell+1}\) (in line 9), and sets the offered price \(P^*_{\ell+1}\) to be \(p^*_{i,\ell+1}\) in the next phase. In going through all the learning phases, the seller progressively refines her estimate on the optimal price, which enables her to establish the choice of optimal price in the earning phase.

The function \(M_\Phi(P)\) in line (7) of the mPC algorithm is defined as follows.

**Definition 1.** Let \(p \in P\) be a discriminative price. We define \(M_\Phi(p)\) as

\[
M_\Phi(p) := \frac{16\sigma^2}{\min_{i \neq j} (d_i(p) - d_j(p))^2} \vee \frac{8b}{\min_{i \neq j} |d_i(p) - d_j(p)|},
\]

where the minimum is taken over distinct pairs of indices \(i, j \in \{1, \cdots, K\}\).

Since we assume \(p\) to be discriminative, \(M_\Phi(p)\) is well defined. The function \(M_\Phi(p)\) measures the distinguishability of the demand functions \(d_1, \cdots, d_k\) under the discriminative price \(p\). We explain the definition of \(M_\Phi(p)\) further in the analysis of mPC.

Define \(M^*_\Phi = \max_{i \in \{1, \cdots, K\}} M_\Phi(p^*_i)\) and \(r^* = \max_{i \in \{1, \cdots, K\}} r^*_i\). The following result shows that the regret of mPC is bounded by \(O(\log^m T)\).

**Theorem 1.** Suppose the demand set \(\Phi\) satisfies Assumption 1. For all \(T \geq 1\), the regret of mPC is bounded by

\[
\text{Regret}^{\text{mPC}}(T) \leq C_\Phi(P^*_0) \max\{\log^m T, 1\} + 4(M^*_\Phi + 1)r^*,
\]

where \(C_\Phi(P^*_0) = \max_{i \in \{1, \cdots, K\}} \{M_\Phi(P^*_0)(r^*_i - r_i(P^*_i))\}\).
Proof Idea of Theorem 1. In the proof, we establish that the regret incurred in Phase 0 is $O(\log^m T)$, and the cumulative regret incurred in the remaining phases is $O(1)$. At the beginning of Phase 0, which is also the beginning of the sale horizon, the seller has no information on the optimal price. Thus, the regret during Phase 0 is proportional to the length of Phase 0. However, in each of the subsequent phases, the seller can choose a price based on the previous sale history. By choosing the lengths of the subsequent phases appropriately, we ensure that the total regret in these phases is $O(1)$. The complete proof can be found in Appendix EC.1. □

Remark 1. In Phase 0, the discriminative price $P_0^*$ is given as a input. One can further reduce the regret bound by choosing a discriminative price $P_0^*$ which minimizes the regret during Phase 0, namely $C_\Phi(P_0^*)$.

Remark 2. In line (9) of Algorithm 1, the test to select a demand function $i_\ell$ is a simple comparison between the sample mean $\bar{X}_\ell$ and the mean demand function value $d_i(P_\ell^*)$. Therefore, the algorithm does not require the seller to know the demand distributions for each demand model. Nevertheless, if the seller does know the demand distribution, line (9) can be replaced by other selection criteria, such as a likelihood ratio test, to improve the efficiency of learning.

4.2. Unbounded but Infrequent Price Experiments

Policy $mPC$ defines $m$ learning phases with iterated-exponentially increasing lengths. This motivates us to consider a modification of $mPC$, which improves the regret bound to a constant. We call this modified policy $uPC$ (which standing for “unbounded price changes”), see Algorithm 2. Although the number of price changes under this policy is not bounded by any finite number as $T$ increases, it grows extremely slowly with order $O(\log^* T)$, where $\log^* T = \min\{m \in \mathbb{Z}^+ : \log^m T \leq 1\}$. For example, for $T \leq 3,000,000$, we have $\log^* T \leq 3$.

Proposition 1. Suppose Assumption 1 holds. For all $T \geq 1$, the pricing policy $uPC$ has regret

$$\text{Regret}_{uPC}(T) \leq C_\Phi(P_0^*) + 2(M_\Phi^* + 1)r^*,$$

where $C_\Phi(P_0^*) = \max_{i \in \{1, \ldots, K\}} \{M_\Phi(P_0^*)(r_i^* - r_i(P_0^*))\}$. 

Algorithm 2 Policy uPC

1: INPUT:
   - A set of demand functions $\Phi = \{d_1, \cdots, d_K\}$.
   - A discriminative price $P_0^*$.

2: Set $\tau_0 = 0$.

3: for $\ell = 0, 1, \cdots$ do

4: From period $\tau_\ell + 1$ to $\tau_{\ell+1} := \tau_\ell + \left\lceil M_\Phi(P_\ell^*)e(\ell) \right\rceil$, set the offered price as $P_\ell^*$.

5: if $T \leq \tau_{\ell+1}$ then stop the algorithm at period $T$.

6: else

7: At the end of period $\tau_{\ell+1}$, compute the sample mean $\bar{X}_\ell$ from period $\tau_\ell + 1$ to $\tau_{\ell+1}$:

   \[\bar{X}_\ell := \frac{\sum_{j=\tau_\ell+1}^{\tau_{\ell+1}} X_j}{\tau_{\ell+1} - \tau_\ell},\]
   where $X_j =$ Number of items sold in period $j$.

8: Choose an index $i_\ell \in \{1, \cdots, K\}$, which solves

   \[\min_{i \in \{1, \cdots, K\}} |\bar{X}_\ell - d_i(P_\ell^*)| .\]

9: Set the next offered price as $P_{\ell+1}^* = p_{i_{\ell}}^*$, where $p_{i_{\ell}}^*$ is an optimal price for demand $d_{i_{\ell}}$.

10: end if

11: end for

The proof of Proposition 1 is included in Appendix EC.2. Furthermore, uPC is an anytime policy, meaning that the seller can apply uPC algorithm without any knowledge of $T$. Anytime policies can be used for customized pricing. In customized pricing, each customer arrival is modeled as a single time period, so $T$ is the total number of customers arrivals (see Harrison et al. 2012, Broder and Rusmevichientong 2012). Since uPC is an anytime policy, the seller is not required to know the total number of customers arrivals.

4.3. Discussion on the Discriminative Price Assumption

The $O(\log(m) T)$ regret of mPC and the $O(1)$ regret of uPC hold under the assumption that there exists an optimal discriminative price for each demand function (Assumption 1). In fact, one can show that this assumption is necessary for any non-anticipating policy to achieve a regret better than $o(\log T)$. 
**Proposition 2.** If Assumption 1 is violated, then there exists a price set $\mathcal{P}$ and a demand set $\Phi$ such that any non-anticipating pricing policy incurs a regret of $\Omega(\log T)$, even if that policy is allowed to change price for infinitely many times.

The proof of Proposition 2 is included in Appendix EC.3. It implies that the best possible regret bound without Assumption 1 is $O(\log T)$.

The remaining question is whether the lower bound in Proposition 2 is tight given that Assumption 1 does not hold. We show next that for any set of $K$ demand functions, there exists a policy $kPC$ (see Algorithm 3) that achieves regret bound of $O(\log T)$ with at most $K - 1$ price changes, regardless of whether Assumption 1 is satisfied. Before introducing the algorithm, we need the following definition:

**Definition 2.** For any nonempty subset of demand functions $A \subset \{d_1, \ldots, d_K\}$, let

$$\tilde{p}_A \in \arg \max_{p \in \mathcal{P}} |\{d_i(p) : d_i \in A\}|.$$  

In other words, $\tilde{p}_A$ is a price that maximizes the number of distinct values of $d_i(p)$ for all $d_i \in A$.

Furthermore, define

$$\tilde{M}_A(p) := \frac{8\sigma^2}{\min_{(i,j):d_i(p) \neq d_j(p)} (d_i(p) - d_j(p))^2} \vee \frac{4b}{\min_{(i,j):d_i(p) \neq d_j(p)} |d_i(p) - d_j(p)|},$$  

where the minimum is taken over all pairs of demand functions $d_i, d_j \in A$ such that $d_i(p) \neq d_j(p)$.

Note that if $|A| \geq 2$, for any pair of demand functions $d_i$ and $d_j$ in $A$, we can always find a price $p$ such that $d_i(p) \neq d_j(p)$, because otherwise the two demand functions are identical. So the value $\tilde{M}_A(\tilde{p}_A)$ in line 4 of Algorithm 3 is well defined for any $|A| \geq 2$.

**Proposition 3.** For all $T \geq 1$, the regret of $kPC$ is bounded by

$$\text{Regret}^{kPC}(T) \leq (K - 1)(\tilde{M}_\Phi r^* \log T + 3r^*),$$  

where $\tilde{M}_\Phi = \max_{A \subset \{1, \ldots, K\}} \tilde{M}_A(\tilde{p}_A)$.

**Proof Idea of Proposition 3.** In each of the learning phases, the definition of the algorithm (line 6) guarantees that at least one demand function is eliminated. The number of iterations in the while loop is at most $K - 1$, and thus the regret of the learning phases is $O((K - 1) \log T)$. We then show that with high probability, the single demand function remained in the earning phase is the true demand function. The complete proof is in Appendix EC.4. \qed
Algorithm 3 Policy kPC.

1. INPUT: A set of demand functions $\Phi = \{d_1, \cdots, d_K\}$.
2. (Learning) Set $A \leftarrow \Phi$. Set $\ell = 0$, $\tau_0 = 0$.
3. while $|A| \neq 1$ do:
   4. Set the price as $P^*_\ell = \tilde{p}_A$ from period $\tau_\ell + 1$ to $\tau_{\ell+1} := \tau_\ell + \left\lceil \tilde{M}_A(P^*_\ell) \log T \right\rceil$.
   5. At the end of period $\tau_{\ell+1}$, compute the sample mean $\bar{X}^\ell$ from period $\tau_\ell + 1$ to $\tau_{\ell+1}$:
      $$\bar{X}^\ell := \frac{\sum_{j=\tau_\ell+1}^{\tau_{\ell+1}} X_j}{\tau_{\ell+1} - \tau_\ell},$$
      where $X_j = \text{Number of items sold in period } j$.
   6. Update $A$: keep all $d_i$ in set $A$ if it is a minimizer of $\min_{d_i \in A} |\bar{X}^\ell - d_i(P^*_\ell)|$. Eliminate other demand functions from $A$. If there are two minimizers $d_i$ and $d_j$ such that $d_i(P^*_\ell) < \bar{X}^\ell < d_j(P^*_\ell)$, remove $d_j$ and only keep $d_i$.
   7. Set $\ell \leftarrow \ell + 1$.
8. end while
9. (Earning) Suppose $A = \{d_i\}$. From period $\tau_\ell + 1$ to period $\tau_{\ell+1} = T$, set the selling price as $P^*_\ell = p^*_i$.

5. Field Experiment at Groupon

We collaborated with Groupon, a large e-commerce marketplace for daily deals, to implement the pricing algorithm presented in Section 4. Groupon offers discount deals from local merchants to subscribed customers. By the second quarter of 2015, Groupon served more than 500 cities worldwide, had nearly 49 million active customers and featured more than 510,000 active deals.

Prior to our collaboration, Groupon applied a fixed price strategy for each deal. Our initial analysis suggested that Groupon could benefit from the dynamic pricing algorithm proposed in Section 4 for the following reasons:
A majority of Groupon’s deals are offered on its website for the first time, and there is not enough historical data to predict demand before new deals are launched. Thus, there is an opportunity for Groupon to learn from real time sales data after deals are launched so as to improve its demand forecast and pricing strategies.

Deals are offered for a limited time, so there is a time tradeoff between price experimentation and revenue maximization, which is a tradeoff addressed by our pricing algorithm.

Groupon managers prefer using as few price changes as possible for a number of reasons. First, they are concerned that frequent price changes may confuse customers. Second, it is easy to communicate and explain a simple dynamic pricing algorithm with minimal price changes to merchants.

Since Groupon mainly offers coupons instead of physical products, we ignored the inventory constraint in this implementation. Technically, each deal has a cap that specifies the maximum quantity that can be sold, but historical data show that only a small fraction of deals have reached their caps. Moreover, once a deal has reached its cap, Groupon can renegotiate with the merchant to increase the cap. So the unlimited inventory assumption is a reasonable approximation of reality.

The pricing algorithm proposed in Section 4 requires a set of possible demand functions as an input. In the rest of this section, we propose a method to generate demand function sets based on clustering. We then describe implementation details and state additional business constraints specified by Groupon. Finally, we present the implementation results and the analysis.
5.1. Generating the Demand Function Set

Recall that we assume a finite demand function set, \( \Phi \), in the model assumption. In reality, it is unlikely that the true demand function will belong to the finite set that we estimated. However, our goal is to find a set \( \Phi \), such that the true demand function can be well approximated by at least one function in the set. To this end, we propose the following three step process to generate a finite set of linear demand functions.\(^1\)

– Step 1: We first collect data on historical deals that have been tested for dynamic pricing. Given a new deal, we select a subset of historical deals with similar features (e.g. deal category, price range, discount rate). Since deals in this subset were tested for dynamic pricing, they have been offered under at least two different prices, so we can fit a linear demand function for each historical deal.

– Step 2: The linear demand functions are then mapped into points on a 2-dimensional plane according the following rule: the \( y \)-coordinate of the plane represents the negative slope of linear functions, and the \( x \)-coordinate represents the mean demand valued at the initial price of the new deal. For example, suppose we fit a demand function \( d_i(p) = a - bp \) for a historical deal, and the new deal has an initial price \( P_1 \), then this demand function is mapped to the point \( (a - bP_1, b) \). Using this mapping, each deal in the subset can be represented by a point on the plane, shown as an ‘x’ in Figure 2. Moreover, the mapping is bijective, so there is a one-to-one correspondence between any point on the plane and a linear demand function.

– Step 3: We apply \( K \)-means clustering to group the points into \( K \) clusters. For example, Figure 2 shows the clustering result for \( K = 3 \). Note that the centroids of the clusters are points on the plane, so according to the bijective mapping defined in Step 2, they also represent linear demand functions. In particular, if a centroid is located at \( (x, y) \), it corresponds to the linear function \( d(p) = x + (P_1 - p)y \). Collectively, the centroids of \( K \) clusters represent \( K \) linear demand functions, which form the demand function set.

Typically, Step 1 would produce hundreds to thousands of historical deals. If we omit Step 3 and simply use linear functions generated in Step 2 as input to our dynamic pricing algorithm, it would be difficult for the learning algorithm to correctly identify the true demand function among thousands of functions within a short period of time. Therefore, we use clustering in Step 3 to limit the number of demand functions to be learned.
To minimize the demand prediction error, we need to select an appropriate number $K$ to balance the bias-variance tradeoff. When $K$ is large, we have a large set of demand functions that can better approximate the true demand function (small bias), but learning about the correct demand function is hard (large variance), since the constant $M^*_K$ in the regret bound of Theorem 1 is large. When $K$ is small, the demand function set has a large approximation error relative to the true demand function (large bias), but it is easier to identify the best function in this set (small variance).

To determine the best value of $K$, we apply cross-validation: The historical deals are randomly split into training and testing sets. Each deal in the testing set had been offered under two prices ($p_1, p_2$) and is treated as a new deal with initial price ($p_1$). We then generate demand functions from the training set following the three-step process described above. After that, we select one function among the $K$ functions whose value at $p_1$ is the closest to the actual demand of the new deal at $p_1$, since this is the function that would have been chosen by our learning algorithm. Next, we compare the realized demand under price $p_2$ to the mean demand predicted by the selected function at $p_2$. The difference between these two values can be interpreted as the prediction error of our learning algorithm. We repeat this process for different values of $K$, and choose $K$ that minimizes prediction error.

In Figure 3, we plot the mean squared error (MSE) of demand prediction for different values of $K$. The dark curve is the MSE of purchase quantity per impression (i.e., per customer visit), and the gray curve is the MSE of booking per impression. The figure shows that the prediction error is large for small values of $K$, and then decreases as $K$ increases. This implies that for small values of $K$, none of the demand functions in the demand set is close to the true demand function (i.e., large bias), so the prediction error is large. For very few demand functions ($2 \leq K \leq 10$), the bias is so large that the prediction error...
is even worse than that of fixed pricing ($K = 1$). Therefore, it is important to choose a large enough $K$ so that at least one of the demand function in the demand set is close to the true demand function. We select $K$ to be around 100 in our final implementation at Groupon. Once $K$ is chosen, the demand functions are generated according to Step 3 in the aforementioned process. Notice that the prediction error will eventually go up due to over-fitting when we select a $K$ that is sufficiently large (i.e., large variance). This trend is not shown in Figure 3, because we didn’t have enough historical deals in this example to demonstrate over-fitting.

5.2. Implementation Details

Compared with the model defined in Section 3, we face some additional constraints in practice due to Groupon’s business rules, so our pricing algorithm must be adjusted to include these implementation details.

For each deal, we suppose that the allowable price set is a continuous interval $P = [p, \bar{p}]$. Each time period is defined as one day, and prices of deals are changed at midnight local time. Since customer traffic for Groupon is not necessarily time homogeneous, in the data pre-processing step, the sales data have been normalized to remove the time effect. More specifically, the normalized sales quantity is the original sales quantity multiplied by a normalization factor, which only depends on time (e.g., number of days after launch, holiday/weekend, etc.) and is specific for each deal category. After this normalization step, we can treat demand in different time periods as stationary.
The main result in Section 4 shows that the first price change captures most of the benefit of dynamic pricing, which reduces regret from $\Theta(T)$ to $\Theta(\log T)$. So we decided to apply the single price change policy at Groupon, i.e., the mPC policy defined in Section 4.1 with $m = 1$.

In the Groupon implementation, the initial price of a deal is negotiated by the local merchant and Groupon, so we treat the initial price as a fixed input — this is the same price that Groupon would have used under its existing fixed price policy. Since a finite set of linear demand functions has only finite non-discriminative prices in the price interval $[p, \bar{p}]$, it is unlikely that the initial price is non-discriminative. In fact, in all the examples that we tested, the initial price $P_1$ is discriminative with respect to the demand set.

When changing price, Groupon decided to allow for price decrease only. The main reason for this decision is that many merchants use Groupon as a marketing channel to attract new customers, so Groupon is unwilling to reduce sales quantity by increasing price, even if it may potentially increase revenue. Groupon further imposes a constraint that price can only be decreased between 5% and 30%. Therefore, if the pricing algorithm recommends either a price decrease of less than 5% or a price increase, then no price change is made. If the algorithm recommends a price decrease of more than 30%, then price is decreased by only 30%.

Moreover, the agreement between Groupon and the local merchant specifies that the merchant’s share of revenue is not affected after price decrease. For example, suppose a deal has an initial price of $20, and both Groupon and the merchant receive $10 from each deal purchased under the initial price. If the pricing algorithm reduces the price to, say, $15, then Groupon would receive $5 and the merchant would still receive $10. This agreement guarantees that merchants always benefit from price decrease, and hopefully this would make merchants more willing to accept the dynamic pricing policy, which is designed to be a win-win strategy for both Groupon and local merchants. In practice, Groupon always give the merchants two options: they can either use the existing fixed price policy, or they can choose to use the dynamic pricing algorithm.

5.3. Field Experiment Results

The field experiment at Groupon consists of two stages. In the first stage we focused on fine-tuning the pricing algorithm, while the second stage was used as a final evaluation of the algorithm performance.
Table 1  Deals selected in the field experiment.

|               | Beauty/Health | Food/Drink | Leisure/Activities | Services | Shopping |
|---------------|---------------|------------|--------------------|----------|----------|
| Number of deals selected | 591           | 111        | 259                | 274      | 60       |
| Mean bookings per day        | 35.1          | 88.0       | 37.6               | 27.2     | 9.3      |

Figure 4  Bookings and revenue increase by deal category.

In the first stage, we tested different ways to generate demand function sets ($\Phi$). The method presented in the previous subsection was the approach that we finally selected. We also used the first stage experiment to test different price switching time. In the definition of algorithm $mPC$ for $m = 1$, the price is switched at period $\lceil M_\Phi(P_0) \log T \rceil$, where the constant $M_\Phi(P_0)$ is given by Eq (4). However, this constant is mainly designed for proving the theoretical regret bound. In practice, we tested several price switching times (between 1 to 7 days) through live experiment, and the switching time with the best performance was chosen.

The second (evaluation) stage of the field experiment lasted for several weeks. During this testing period, 1,295 deals were selected by our dynamic pricing algorithm for price decrease. These deals span five product categories: Beauty/Healthcare, Food/Drink, Leisure/Activities, Services, and Shopping. Table 1 provides information on the number of deals and the average daily bookings per category. We note that if a deal was tested for dynamic pricing but was not selected for price decrease, that deal is not included in this dataset.

In the field experiment we focused on two performance metrics. One is the total amount of money paid by customers to Groupon, referred to as bookings, and it is directly related to Groupon’s market share. The other is the part of the revenue that Groupon keeps
after paying local merchants, referred to simply as \textit{revenue}. For each product category, we compare the average bookings and revenue per day pre and post price change. Since the initial price is determined by Groupon’s current fixed price policy, comparing the average daily bookings and daily revenue before and after the price change measures the revenue lift of our dynamic pricing policy over Groupon’s existing fixed price policy. As we did in the data preprocessing step, the lift has been normalized to control for time-varying demand effect. Specifically, when generating Figure 4, we multiply the booking and revenue quantities per day by a normalization factor, which only depends on time (e.g., number of days after launch, holiday/weekend, etc.) and deal category.

Figure 4 shows the average increase in daily bookings and revenue after price decrease. Overall, daily bookings are increased by 116%, and daily revenue is increased by 21.7%. Among the five categories, Beauty/Healthcare, Food/Drink, and Shopping have significant increase in both revenue and bookings. Services category has almost no revenue change but significant bookings increase. Leisure/Activities category has a negative gain in revenue. For Groupon, Beauty/Healthcare and Food/Drink are the two leading categories in terms of revenue generation. Our pricing algorithm has solid revenue gain in these two categories.

Further analysis of the field experiment result shows that reducing price has a much bigger impact on deals that have fewer bookings per day, which holds across all categories. Overall, the average increase in daily revenue is 116% for deals with bookings per day below the median, while the increase is only 14% for deals with bookings per day above the median. This effect also explains the big increase in bookings and revenue for the Shopping category, the category with the smallest mean daily bookings among all five categories, see Table 1. Therefore, the increases on the Shopping deals are also the most significant.

Lastly, our pricing algorithm has a poor performance for Leisure/Activities category, despite the fact that this categories has almost the same level of average daily bookings as the Beauty category. We suspect the reason is that some features of customer demand for Leisure/Activities deals are not captured by our demand model. For example, it might be that the weekend/holiday effect is stronger for this category than we estimated. The data preprocessing step does include time normalization for weekend/holiday effect, but it is likely that customers purchase Leisure/Activities deals a few days before holidays, instead of during holidays. Therefore, further work is needed to improve the demand prediction method for the Leisure/Activities category.
6. Conclusion

We consider a dynamic pricing problem where the latent demand model is unknown but belongs to a finite set of demand functions. The seller faces a constraint that price can be changed at most \( m \) times. We propose a pricing policy that incurs a regret of \( O(\log^m T) \), where \( T \) is the length of the sales horizon and \( \log^m T \) is \( m \) iterations of logarithm.

We then implement the pricing algorithm at Groupon, a website that sells deals from local merchants. We design a process to generate linear demand function sets from historical data, and use it as an input to our pricing algorithm. The algorithm incorporates Groupons’s business rules, which allow at most one price decrease per deal. The field experiment shows that the algorithm has significantly improved both daily revenue and bookings per deal.

Endnotes

1. We use linear demand functions to approximate local price elasticity, but our method can also be adapted for other forms of parametric demand families such as Cobb-Douglas functions, as long as the demand functions can be specified by finitely many parameters.

2. In our pricing algorithm, we can easily include merchant’s revenue share by redefining the optimal price as \( p^*_t \in \arg\max_{p \in P} (p - c)d_i(p) \), where \( c \) is the merchant’s fixed revenue split. All the results in Section 4 go through with this modification.

Appendix A: A note on the definition of a non-anticipatory policy

Conventionally, a pricing policy \( \pi \) is said to be non-anticipatory if

\[
P^\pi_t = f(P^\pi_1, X_1, P^\pi_2, ..., P^\pi_{t-1}, X_{t-1}; T)
\]

for some function \( f \). The last argument \( T \) is the length of sales horizon. We claim that \( P^\pi_t \) can be alternatively expressed as

\[
P^\pi_t = g(X_1, X_2, ..., X_{t-1}; T)
\]

by another suitably defined function \( g \).

The claim is shown by induction on \( t \). For \( t = 1 \), we have \( P^\pi_1 = f(\emptyset; T) \), for which we can express in terms of another function \( g \) by simply defining \( g(\emptyset; T) = f(\emptyset; T) \). Suppose the claim is true for \( t - 1 \); that is, we have constructed a function \( g \) such that \( P^\pi_s = g(X_1, X_2, ..., X_s; T) \) for every \( s \in \{1, ..., t-1\} \). Then we can extend the definition \( g \) to the \( t^{th} \) period offered price \( P^\pi_t \) by defining

\[
g(X_1, X_2, ..., X_{t-1}; T) = f(g(\emptyset; T), X_1, g(X_1; T), ..., g(X_1, ..., X_{t-2}; T), X_{t-1}; T),
\]

we see that \( P^\pi_t = g(X_1, X_2, ..., X_{t-1}; T) \). This completes the proof of the claim, hence justifying the definition by (1).
References

Aviv, Y. and Vulcano, G. (2012). Dynamic list pricing. In Özer, Ö. and Phillips, R., editors, The Oxford Handbook of Pricing Management, pages 522–584. Oxford University Press, Oxford.

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.

Bitran, G. R. and Mondschein, S. V. (1997). Periodic pricing of seasonal products in retailing. Management Science, 43(1):64–79.

Boyaci, T. and Özer, Ö. (2010). Information acquisition for capacity planning via pricing and advance selling: When to stop and act? Operations Research, 58(5):1328–1349.

Broder, J. (2011). Online Algorithms For Revenue Management. PhD thesis, Cornell University.

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

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

Chen, Q., Jasin, S., and Duenyas, I. (2015). Real-time dynamic pricing with minimal and flexible price adjustment. Management Science, 62(8):2437–2455.

den Boer, A. and Zwart, B. (2014). Simultaneously learning and optimizing using controlled variance pricing. Management Science, 60(3):770–783.

den Boer, A. V. (2014). Dynamic pricing with multiple products and partially specified demand distribution. Mathematics of operations research, 39(3):863–888.

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.

Feng, Y. and Gallego, G. (1995). Optimal starting times for end-of-season sales and optimal stopping times for promotional fares. Management Science, 41(8):1371–1391.

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

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, R. and Leighton, T. (2003). The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In Proceedings of 44th Annual IEEE Symposium on Foundations of Computer Science, pages 594–605.
Mersereau, A. J., Rusmevichientong, P., and Tsitsiklis, J. N. (2009). A structured multiarmed bandit problem and the greedy policy. *IEEE Transactions on Automatic Control, 54*(12):2787–2802.

Netessine, S. (2006). Dynamic pricing of inventory/capacity with infrequent price changes. *European Journal of Operational Research, 174*(1):553–580.

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

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

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.

Zbaracki, M. J., Ritson, M., Levy, D., Dutta, S., and Bergen, M. (2004). Managerial and customer costs of price adjustment: direct evidence from industrial markets. *Review of Economics and Statistics, 86*(2):514–533.
E-companion: Additional Proofs for Section 4

Appendix EC.1: Proof of Theorem 1

Proof of Theorem 1. Suppose $d_1$ is the underlying demand function. The regret under demand $d_1$ can be decomposed as

$$\text{Regret}^{\text{mPC}}_1(T) = \mathbb{E}_1 \left[ \sum_{t=1}^{T} (r_1^* - r_1(P_t)) \right] = \sum_{\ell=0}^{m} \mathbb{E}_1 \left[ \sum_{t=\tau_{\ell}+1}^{\tau_{\ell+1}} (r_1^* - r_1(P_t)) \right].$$

We first consider the case where $\log(m) T > 0$. By definition, $\tau_1 = \left\lceil M_\Phi(P_0^*) \log(m) T \right\rceil$, so the regret during Phase 0 is equal to

$$\mathbb{E}_1 \left[ \sum_{t=1}^{\tau_1} (r_1^* - r_1(P_t)) \right] = \left\lceil M_\Phi(P_0^*) \log(m) T \right\rceil (r_1^* - r_1(P_0^*)). \quad (\text{EC.1})$$

Next, we show that for each $1 \leq \ell \leq m$, the regret during Phase $\ell$ is bounded by

$$\mathbb{E}_1 \left[ \sum_{t=\tau_{\ell}+1}^{\tau_{\ell+1}} (r_1^* - r_1(P_t)) \right] \leq \frac{2M_\Phi^* r_1^*}{\log^{(m-\ell)}T} + \frac{2r_1^*}{(\log^{(m-\ell)}T)^2}, \quad (\text{EC.2})$$

where $M_\Phi^* = \max_{i \in \{1, \ldots, K\}} M_\Phi(p_i^*)$.

For $1 \leq \ell \leq m$, the regret during Phase $\ell$ satisfies the following bound:

$$\mathbb{E}_1 \left[ \sum_{t=\tau_{\ell}+1}^{\tau_{\ell+1}} (r_1^* - r_1(P_t)) \right] = \mathbb{E}_1 \left[ (\tau_{\ell+1} - \tau_{\ell}) \times (r_1^* - r_1(P_{\ell}^*)) \right] \
\leq \mathbb{E}_1 \left[ (M_\Phi(P_{\ell}^*) \log^{(m-\ell)}T + 1) \times (r_1^* - r_1(P_{\ell}^*)) \right] \
\leq \left( M_\Phi^* \log^{(m-\ell)}T + 1 \right) \sum_{i=1}^{K} (r_i^* - r_1(p_i^*)) \times \mathbb{P}_1[P_{\ell}^* = p_i^*] \
\leq \left( M_\Phi^* \log^{(m-\ell)}T + 1 \right) r_1^* \times \sum_{i=2}^{K} \mathbb{P}_1[P_{\ell}^* = p_i^*]. \quad (\text{EC.3})$$

In the expression above, the expectation is taken on the price offered in Phase $\ell$, $P_{\ell}^*$, which is a random variable depending on the realized demand in phases $0, \ldots, \ell - 1$. In Eq (EC.3), we use the fact that for all $\ell = 1, \ldots, m$, the offered price $P_{\ell}^* \in \{p_1^*, \ldots, p_K^*\}$ (see line 10 of mPC).

To complete the proof of inequality (EC.2), we prove the following inequality:

$$\sum_{i=2}^{K} \mathbb{P}_1[P_{\ell}^* = p_i^*] \leq \frac{2}{(\log^{(m-\ell)}T)^2}. \quad (\text{EC.4})$$
By the definition of mPC, the choice of price $P^*_\ell$ is determined by the sample mean $\bar{X}_{\ell-1}$ in Phase $\ell-1$, so we have

$$\sum_{i=2}^{K} \mathbb{P}_1[P^*_\ell = p^*_i] = \mathbb{P}_1[|\bar{X}^\ell - d_1(P^*_\ell-1)| \geq |\bar{X}^\ell - d_1(P^*_\ell-1)| \text{ for some } i \neq 1].$$

Now, if $|\bar{X}^\ell - d_1(P^*_\ell-1)| \geq |\bar{X}^\ell - d_1(P^*_\ell-1)|$ for some $i \neq 1$, we have

$$|\bar{X}^\ell - d_1(P^*_\ell-1)| \geq \frac{1}{2} (|\bar{X}^\ell - d_1(P^*_\ell-1)| + |\bar{X}^\ell - d_1(P^*_\ell-1)|) \geq \frac{1}{2} |d_1(P^*_\ell-1) - d_1(P^*_\ell-1)|,$$

where the last step uses the triangle inequality. This leads to the following bound:

$$\sum_{i=2}^{K} \mathbb{P}_1[P^*_\ell = p^*_i] \leq \mathbb{P}_1 \left[ |\bar{X}^\ell - d_1(P^*_\ell-1)| \geq \frac{1}{2} \min_{i \neq 1} |d_1(P^*_\ell-1) - d_1(P^*_\ell-1)| \right]. \quad (EC.5)$$

Given price $P^*_\ell-1$, sample mean $\bar{X}_{\ell-1}$ is the average of i.i.d. random variables with mean $d_1(P^*_\ell-1)$. Because demand in each period is light-tailed with parameters $(\sigma, b)$, we can apply the Chernoff inequality: conditioning on $P^*_\ell-1$, for any $\epsilon > 0$, it holds that

$$\mathbb{P}_1 \left[ |\bar{X}^\ell - d_1(P^*_\ell-1)| \geq \epsilon |P^*_\ell-1| \right] \leq 2 \exp \left( -\left( \tau_\epsilon - \tau_{\ell-1} \right) \left( \frac{\epsilon^2}{2\sigma^2} \wedge \frac{\epsilon}{2b} \right) \right).$$

Let $\epsilon = \frac{1}{2} \min_{i \neq 1} |d_1(P^*_\ell-1) - d_i(P^*_\ell-1)|$. Because $\tau_\epsilon - \tau_{\ell-1} = \left[ M_\Phi(P^*_\ell-1) \log^{(m-\ell+1)} T \right]$, we have

$$\mathbb{P}_1 \left[ |\bar{X}^\ell - d_1(P^*_\ell-1)| \geq \frac{1}{2} \min_{i \neq 1} |d_1(P^*_\ell-1) - d_i(P^*_\ell-1)| \right] \leq 2 \mathbb{E}_1 \left[ \exp \left( -\left[ M_\Phi(P^*_\ell-1) \log^{(m-\ell+1)} T \right] \left( \frac{\epsilon^2}{2\sigma^2} \wedge \frac{\epsilon}{2b} \right) \right) \right] \leq 2 \exp \left( -\left[ M_\Phi(P^*_\ell-1) \log^{(m-\ell+1)} T \left( \frac{\epsilon^2}{2\sigma^2} \wedge \frac{\epsilon}{2b} \right) \right] \right) \leq 2 \exp \left( -2 \log^{(m-\ell+1)} T \right) \leq \frac{2}{\left( \log^{(m-\ell)} T \right)^2}, \quad (EC.6)$$

where step (EC.6) uses the definition

$$M_\Phi(P^*_\ell-1) = 2 \times \left( \frac{2\sigma^2}{\frac{1}{2} \min_{i \neq j} (d_i(P^*_\ell-1) - d_j(P^*_\ell-1))^2} \wedge \frac{2b}{\frac{1}{2} \min_{i \neq j} |d_i(P^*_\ell-1) - d_j(P^*_\ell-1)|} \right).$$

By integrating over the realizations of $P^*_\ell-1$ in the above bound, we have established inequality (EC.4), which in turn proves (EC.2).
Combining Eqs (EC.1) and (EC.2), we can prove the regret bound on mPC under demand \( d_i \) as follows:

\[
\text{Regret}^\text{mPC}_i(T) = \mathbb{E}_1 \left[ \sum_{t=1}^{\tau_1} (r_1^* - r_1(P_t)) \right] + \sum_{\ell=1}^m \mathbb{E}_1 \left[ \sum_{t=\tau_{\ell+1}}^{\tau_{\ell+1}} (r_1^* - r_1(P_t)) \right] \\
\leq \left( M_\Phi(P_0^*) \log^{(m)} T + 1 \right) (r_1^* - r_1(P_0^*)) + \sum_{\ell=1}^m \left( \frac{2M_\Phi^* r_1^*}{\log^{(m-\ell)} T} + \frac{2r_1^*}{(\log^{(m-\ell)} T)^2} \right).
\]

Since \( \log^{(m)} T > 0 \), it is easily verified that \( \log^{(m-\ell)} T \geq e^{\ell-1} \) for all \( \ell \geq 1 \), so

\[
\sum_{\ell=1}^m \frac{1}{\log^{(m-\ell)} T} \leq \sum_{\ell=1}^\infty \frac{1}{e^{\ell-1}} \leq 2, \quad \sum_{\ell=1}^m \frac{1}{(\log^{(m-\ell)} T)^2} \leq \sum_{\ell=1}^\infty \frac{1}{e^{2\ell-2}} \leq \frac{3}{2}.
\]

Therefore,

\[
\text{Regret}^\text{mPC}_i(T) \leq \left( M_\Phi(P_0^*) \log^{(m)} T + 1 \right) (r_1^* - r_1(P_0^*)) + 4M_\Phi^* r_1^* + 3r_1^* \\
\leq M_\Phi(P_0^*)(r_1^* - r_1(P_0^*)) \log^{(m)} T + 4M_\Phi^* r_1^* + 4r_1^*.
\]

The minimax regret of demand set \( \Phi \) is bounded by

\[
\text{Regret}^\Phi(T) = \max_{i=1,\ldots,K} \text{Regret}^\text{mPC}_i(T) \leq C_\Phi(P_0^*) \log^{(m)} T + 4M_\Phi^* r^* + 4r^*,
\]

where \( C_\Phi(P_0^*) = \max_{i \in \{1, \ldots, K\}} \{ M_\Phi(P_0^*)(r_1^* - r_i(P_0^*)) \} \) and \( r^* = \max_{i \in \{1, \ldots, K\}} r_i^* \).

If \( \log^{(m)} T = 0 \), let \( m' \leq m \) be the largest integer such that \( \log^{(m')} T > 0 \). Clearly, \( \log^{(m')} T \leq 1 \).

In this case, policy mPC applied to \( T \) periods uses only \( m' \) price changes, so

\[
\text{Regret}^\text{mPC}_i(T) \leq C_\Phi(P_0^*) \log^{(m')} T + 4M_\Phi^* r^* + 4r^* \leq C_\Phi(P_0^*) + 4M_\Phi^* r^* + 4r^*.
\]

Combining both cases for \( \log^{(m)} T > 0 \) and \( \log^{(m)} T = 0 \), we have

\[
\text{Regret}^\text{mPC}_i(T) \leq C_\Phi(P_0^*) \max\{\log^{(m)} T, 1\} + 4M_\Phi^* r^* + 4r^*.
\]

\[\square\]

**Appendix EC.2: Proof of Proposition 1**

Proof of Proposition 1. Let \( m \) be the integer such that \( \tau_m < T \leq \tau_{m+1} \). Suppose \( d_i \) is the underlying demand function. The regret under demand \( d_i \) can be composed as

\[
\text{Regret}^\text{mPC}_i(T) = \mathbb{E}_1 \left[ \sum_{t=1}^T (r_1^* - r_1(P_t)) \right] \leq \sum_{\ell=0}^m \mathbb{E}_1 \left[ \sum_{t=\tau_{\ell+1}}^{\tau_{\ell+1}} (r_1^* - r_1(P_t)) \right].
\]

The regret during Phase 0 is equal to

\[
\mathbb{E}_1 \left[ \sum_{t=1}^{\tau_1} (r_1^* - r_1(P_t)) \right] = [M_\Phi(P_0^*)](r_1^* - r_1(P_0^*)).
\]
For $1 \leq \ell \leq m$, the offered price $P^*_\ell \in \{p^*_1, \ldots, p^*_K\}$, so the regret during Phase $\ell$ is bounded by:

$$
\mathbb{E}_1 \left[ \sum_{t=\tau_{\ell}+1}^{\tau_{\ell+1}} (r_i^* - r_1(P_t)) \right] \\
= \mathbb{E}_1 \left[ (r_{\ell+1} - \tau_\ell) \times (r_i^* - r_1(P^*_\ell)) \right] \\
\leq \mathbb{E}_1 \left[ (M_\Phi(P^*_\ell)e^{(\ell)} + 1) \times (r_i^* - r_1(P^*_\ell)) \right] \\
\leq (M_\Phi e^{(\ell)} + 1) \sum_{i=1}^K (r_i^* - r_1(p^*_i)) \times \mathbb{P}_1[P^*_\ell = p^*_i] \\
\leq (M_\Phi e^{(\ell)} + 1) r_i^* \times \sum_{i=2}^K \mathbb{P}_1[P^*_\ell = p^*_i].
$$

By the definition of the uPC policy, the choice of price $P^*_\ell$ is determined by the sample mean $\bar{X}_{\ell-1}$ in Phase $\ell - 1$. Similar to the proof of Theorem $1$, letting $\varepsilon = \frac{1}{2} \min_{i \neq 1} |d_1(P^*_{\ell-1}) - d_i(P^*_{\ell-1})|$, we have

$$
\sum_{i=2}^K \mathbb{P}_1[P^*_\ell = p^*_i] \\
\leq \mathbb{P}_1 \left[ |\bar{X}_{\ell-1} - d_1(P^*_{\ell-1})| \geq \varepsilon \right] \\
= \mathbb{E}_1 \left[ \mathbb{P}_1 \left[ |\bar{X}_{\ell-1} - d_1(P^*_{\ell-1})| \geq \varepsilon |P^*_{\ell-1} \right] \right] \\
\leq \mathbb{E}_1 \left[ 2\mathbb{E}_1 \left[ \exp \left( -\left(\bar{X}_{\ell-1} - (\ell^2/2\sigma^2 \wedge \varepsilon/2b) \right) \right) \left\lvert P^*_{\ell-1} \right\rvert \right] \right] \\
\leq \mathbb{E}_1 \left[ 2\mathbb{E}_1 \left[ \exp \left( -M_\Phi(P^*_{\ell-1})e^{(\ell-1)}(\ell^2/2\sigma^2 \wedge \varepsilon/2b) \right) \right] \right] \\
\leq \mathbb{E}_1 \left[ 2\mathbb{E}_1 \left[ \exp \left( -2e^{(\ell-1)} \right) \right] \right] \\
= 2/(e^{(\ell)})^2.
$$

So

$$
\mathbb{E}_1 \left[ \sum_{t=\tau_{\ell}+1}^{\tau_{\ell+1}} (r_i^* - r_1(P_t)) \right] \leq (M_\Phi e^{(\ell)} + 1) r_i^* \cdot \frac{2}{(e^{(\ell)})^2} = \frac{2M_\Phi r_i^*}{e^{(\ell)}} + \frac{2r_i^*}{(e^{(\ell)})^2}.
$$

In sum, the regret of uPC under demand $d_1$ is bounded by

$$
\text{Regret}^{\text{uPC}}_1(T) = \mathbb{E}_1 \left[ \sum_{i=1}^{\tau_1} (r_i^* - r_1(P_t)) \right] + \sum_{\ell=1}^m \mathbb{E}_1 \left[ \sum_{t=\tau_{\ell}+1}^{\tau_{\ell+1}} (r_i^* - r_1(P_t)) \right] \\
\leq (M_\Phi(P^*_0) + 1)(r_i^* - r_1(P^*_0)) + \sum_{i=1}^K \left( \frac{2M_\Phi r_i^*}{e^{(\ell)}} + \frac{2r_i^*}{(e^{(\ell)})^2} \right) \\
\leq M_\Phi(P^*_0)(r_i^* - r_1(P^*_0)) + r_i^* + (2M_\Phi r_i^* + r_i^*) \\
= M_\Phi(P^*_0)(r_i^* - r_1(P^*_0)) + 2M_\Phi r_i^* + 2r_i^*.
$$

The minimax regret of demand set $\Phi$ is given by

$$
\text{Regret}^{\text{uPC}}_\Phi(T) = \max_{i=1,\ldots,K} \text{Regret}^{\text{uPC}}_i(T) \leq C_\Phi(P^*_0) + 2M_\Phi r^* + 2r^*,
$$

where $C_\Phi(P^*_0) = \max_{i \in \{1,\ldots,K\}} \{M_\Phi(P^*_0)(r_i^* - r_i(P^*_0))\}$ and $r^* = \max_{i \in \{1,\ldots,K\}} r_i^*$. \hfill \Box
Appendix EC.3: Proof of Proposition 2

Proof of Proposition 2. Consider a price set \( \mathcal{P} = \{1, 2\} \) and two demand functions \( d_1(1) = 0.6, d_1(2) = 0.25; d_2(1) = 0.4, d_2(2) = 0.25 \). Demand per period has a Bernoulli distribution. It is clear that the optimal prices are \( p_1^* = 1, p_2^* = 2 \). This demand model violates Assumption 1, because \( p_2^* = 2 \) is not a discriminative price. We show that for this model, any non-anticipating policy must have a regret of \( \Omega(\log T) \).

The one period regret for not using the optimal price is \( a = 0.1 \) under either demand function. For any policy, we let \( T_1 \) be the number of the times that \( p = 1 \) is used.

We prove the result by contradiction. Suppose \( \text{Regret}_2(T) = a \cdot E_2[T_1] = o(1) \cdot \log T \) and \( \text{Regret}_1(T) = a(E_1[T - T_1]) = o(1) \cdot \log T \). The change-of-measure inequality (see Lemma EC.1) implies that for any event \( A \),
\[
P_2[A] \leq E_1[1_A \exp(bT_1)].
\]
where \( b = \log(0.6/0.4) \).

Consider the event: \( A = \{T_1 \leq \log T/(2b)\} \), then we have
\[
P_2[A] \leq P_1[A] \exp(b \cdot \log T/(2b)) = P_1[A] \sqrt{T}.
\]

By Markov’s inequality,
\[
P_1[A] = P_1[T - T_1 \geq T - \log T/(2b)] \leq \frac{E_1[T - T_1]}{T - \log T/(2b)} = \frac{o(1) \log T}{T - \log T/(2b)}.
\]
Thus, we have
\[
P_2[A] \leq \frac{o(1) \sqrt{T} \log T}{T - \log T/(2b)} = o(1).
\]

Using Markov’s inequality again, we get
\[
E_2[T_1] \geq \frac{\log T}{2b} P_2[T_1 \geq \log T/(2b)] = \frac{\log T}{2b} (1 - P_2[A]) = \frac{\log T}{2b} (1 - o(1)).
\]
This contradicts the assumption that \( E_2[T_1] = o(1) \cdot \log T \). \( \square \)

Appendix EC.4: Proof of Proposition 3

Proof of Proposition 3. Suppose \( d_1 \) is the underlying demand function. Let \( k \leq K - 1 \) be the number of iterations in the while loop.

The regret under demand \( d_1 \) can be composed as
\[
\text{Regret}_1^{\text{PC}}(T) = E_1 \left[ \sum_{t=1}^{T} (r_1^* - r_1(P_t)) \right] \leq E_1 \left[ \sum_{t=0}^{k} \sum_{\ell=1}^{T_{\ell+1}} (r_1^* - r_1(P_t)) \right] .
\]
Let \( \epsilon = \frac{1}{2} \min_{i:d_1(p_{\ell-1}^*) \neq d_i(p_{\ell-1}^*)} |d_1(p_{\ell-1}^*) - d_i(p_{\ell-1}^*)| \). The probability that demand \( d_1 \) is eliminated in phase \( \ell < k \) is bounded by

\[
\mathbb{P}_1 \left( |\bar{X}^{\ell-1} - d_1(p_{\ell-1}^*)| \geq |\bar{X}^{\ell-1} - d_i(p_{\ell-1}^*)| \right) \text{ for some } i \neq 1
\]

Inequality (EC.7) is proved in Theorem 1, and (EC.8) uses the Chernoff bound. Since \( k \leq K - 1 \), we have

\[
\mathbb{P}_1 \left( |\bar{X}^{\ell-1} - d_1(p_{\ell-1}^*)| \geq |\bar{X}^{\ell-1} - d_i(p_{\ell-1}^*)| \right) \text{ for some } i \neq 1, 0 \leq \ell < k \leq \frac{2(K-1)}{T}.
\]

For each of the learning phase (\( 0 \leq \ell \leq k - 1 \)), the regret is bounded by

\[
\mathbb{E}_1 \left[ \sum_{\ell=\tau_\ell+1}^{\tau_{\ell+1}} (r_{\ell}^* - r_1(p_{\ell})) \right] = \mathbb{E}_1 \left[ \tilde{M}_\Phi(p_{\ell}) \log T (r_{\ell}^* - r_1(p_{\ell})) \right] \leq \tilde{M}_\Phi r_1^* \log T + r_1^*.
\]

The regret in the earning phase (\( \ell = k \)) is bounded by

\[
\mathbb{E}_1 \left[ \sum_{\ell=\tau_k+1}^{T} (r_{\ell}^* - r_1(p_{\ell})) \right] \leq Tr_1^* \mathbb{P}_1[p_k \neq p_1^*].
\]

So the regret of kPC under demand \( d_1 \) is bounded by

\[
\text{Regret}_{1}^{kPC}(T) = \mathbb{E}_1 \left[ \sum_{\ell=0}^{k-1} \sum_{t=\tau_\ell+1}^{\tau_{\ell+1}} (r_{\ell}^* - r_1(p_{\ell})) \right] + \mathbb{E}_1 \left[ \sum_{\ell=\tau_k+1}^{T} (r_{\ell}^* - r_1(p_{\ell})) \right]
\]

\[
\leq (K-1)(\tilde{M}_\Phi r_1^* \log T + r_1^*) + Tr_1^* \frac{2(K-1)}{T}
\]

\[
=(K-1)\tilde{M}_\Phi r_1^* \log T + 3(K-1)r_1^*.
\]

The minimax regret of demand set \( \Phi \) is given by

\[
\text{Regret}_{\Phi}^{kPC}(T) = \max_{i=1,\ldots,K} \text{Regret}_{i}^{kPC}(T) \leq (K-1)\tilde{M}_\Phi r^* \log T + 3(K-1)r^*.
\]

\[\square\]
EC.4.1. Proof of Lemma EC.1

Consider a problem instance \((\Gamma)\) that satisfies the following conditions:

1. There exists a constant \(Q_\Gamma > 0\), such that \(\sum_{i=1}^K (r_i^* - r_i(p)) \geq Q_\Gamma\) for all \(p \in \mathcal{P}\).
2. The demand \(D(p) \in \mathbb{N}\) for any price \(p \in \mathcal{P}\).
3. Given \(p \in \mathcal{P}\), there exists a subset \(\mathcal{B}_p \subset \mathbb{N}\), such that for all \(i\), \(\mathbb{P}_i[D(p) = d] > 0\) if and only if \(d \in \mathcal{B}_p\).
4. There exists a constant \(0 < \kappa_\Gamma < 1\), such that \(\mathbb{P}_i[D(p) = d]/\mathbb{P}_j[D(p) = d] \geq \kappa_\Gamma\) for all \(i, j \in \{1, \ldots, K\}, p \in \mathcal{P}, d \in \mathcal{B}_p\).

The first condition states that there is no price \(p \in \mathcal{P}\) that simultaneously maximizes the revenue of all demand functions in \(\Phi\). This ensures that the problem instance is nontrivial and a learning process is necessary for maximizing the revenue when the demand function is unknown. The second condition is that demand must be integers. The third condition states that all demand functions have the same support for a given price. The fourth condition states that the ratios of probability mass functions of different demand models are bounded. Based on these conditions, we prove the following change-of-measure lemma.

**Lemma EC.1.** Let \(H_t = (P_1, X_1, \ldots, P_t, X_t)\) be the history observed by the end of period \(t\), and let \(h_t\) be a realization of \(H_t\). For any non-anticipating pricing policy \(\pi\), we have

\[
\mathbb{P}_i^\pi[H_t = h_t] \geq \kappa_1^i \mathbb{P}_i^\pi[H_t = h_t],
\]

for all \(i, i' \in \{1, \ldots, K\}\). The constant \(\kappa_1\) is defined in the condition \((\Gamma)\).

**Proof of Lemma EC.1.** Let \(h_t = (p_1, x_1, \ldots, p_t, x_t)\) be a realization of \(H_t = (P_1, X_1, \ldots, P_t, X_t)\). We first assume \(\mathbb{P}_i^\pi[H_t = h_t] > 0\), so we have

\[
\begin{align*}
\mathbb{P}_i^\pi[H_t = h_t] &= \prod_{s=1}^t \mathbb{P}_i^\pi[D(p_s) = x_s] \prod_{s=1}^{t-1} \mathbb{P}_i^\pi[P_{s+1} = p_{s+1} \mid H_s = h_s] \\
&= \prod_{s=1}^t \left( \mathbb{P}_i^\pi[D(p_s) = x_s] \cdot \frac{\mathbb{P}_i^\pi[D(p_s) = x_s]}{\mathbb{P}_{i'}^\pi[D(p_s) = x_s]} \right) \prod_{s=1}^{t-1} \mathbb{P}_i^\pi[P_{s+1} = p_{s+1} \mid H_s = h_s] \\
&\geq \prod_{s=1}^t (\mathbb{P}_i^\pi[D(p_s) = x_s] \cdot \kappa_1^s) \prod_{s=1}^{t-1} \mathbb{P}_i^\pi[P_{s+1} = p_{s+1} \mid H_s = h_s] \\
&= \kappa_1^i \prod_{s=1}^t \mathbb{P}_i^\pi[D(p_s) = x_s] \prod_{s=1}^{t-1} \mathbb{P}_i^\pi[P_{s+1} = p_{s+1} \mid H_s = h_s] \\
&= \kappa_1^i \mathbb{P}_i^\pi[H_t = h_t] \quad \text{(EC.11)}
\end{align*}
\]

\(\kappa_1^i\) is defined in the condition \((\Gamma)\).
Step (EC.9) uses the third condition of (Γ), which states that all demand functions have the same support under a given price, so $P_\pi^s[D(p_s) = x_s] \neq 0$. Step (EC.10) uses the fourth condition of (Γ). Step (EC.11) holds because price $P_{s+1}$ is determined by policy $\pi$ and realized history $h_s$, and is independent of the underlying demand model. Note that if $\pi$ is a deterministic policy, we always have $P_\pi^s[p_{s+1} | H_s = h_s] = 1$ for all $i$.

Finally, if $P_\pi^s[H_t = h_t] = 0$, we have $P_\pi^s[H_t = h_t] = 0$, too. This is again due to the third condition of (Γ), which states that all demand functions have the same support under a given price. $\square$