Chapter 10
Online Learning

10.1 Introduction

In the models that we have studied so far, we have assumed that the demand model and its parameters are all known. In practice, demand models need to be estimated before dynamic pricing, assortment optimization, and revenue management can be effectively done. In some instances, there is enough data over a long period of time to calibrate different demand models, do model selection, and update parameter estimates. At the other extreme, we may be pricing for products for which we have little or no information. In this case, demand learning needs to be done on the fly. This is particularly true for online retailing of new products. In this chapter, we address the problem of online demand learning. We study the expected loss in revenue of a learning-and-earning policy relative to an optimal clairvoyant policy that knows the expected demand function. We consider both the case of ample and constrained capacity and measure how the regret grows as the length of the sales horizon increases. We present only the strongest available results for both the case of ample and the case of constrained capacity. In Sect. 10.2, we consider the case with ample capacity, whereas in Sect. 10.3, we consider the case with constrained capacity.

10.2 Ample Inventory Model

Let $D_t(p) \in \{0, 1\}$ be the random demand in period t at price p. We will assume that $D_t(p), t \in \{1, \ldots, T\}$ are independent and identically distributed Bernoulli random variables with mean $d(p) = \mathbb{E}[D_t(p)]$ for all $p \in [l, u]$ where l and u are non-negative constants. Without loss of generality we rescale prices if necessary so that $[l, u] = [0, 1]$. We assume that the $d(p)$ is unknown and that the goal of the

G. Gallego, H. Topaloglu, *Revenue Management and Pricing Analytics*,
International Series in Operations Research & Management Science 279,
https://doi.org/10.1007/978-1-4939-9606-3_10

seller is to maximize revenues over the selling horizon via a learning and earning algorithm. We assume that capacity c is ample. In the context of the Bernoulli model just described, this means that $c \geq T$, because there is at most one unit of sale at each period.

We will study both parametric and non-parametric models. In parametric models, the seller knows the form of the expected demand, say $d(p) = \lambda e^{-p/\theta}$, but needs to estimate the unknown parameters (in this case λ and θ). The non-parametric case makes no assumptions about the form of $d(p)$. In the parametric case, the parameters may be updated over time by following several techniques including the Bayesian approach, maximum likelihood, or least squares. For non-parametric models, the exploration does not attempt to estimate the demand function itself as its main concern is to obtain prices that work well empirically.

Let $R(p) = pd(p)$ be the revenue at price p. We assume there exist a unique maximizer of $R(p)$, say $p^* \in [0, 1]$. Over the selling horizon, the expected revenue obtained by the clairvoyant policy is $TR(p^*)$. The objective is to design a non-anticipating pricing policy π_t to maximize the total reward $\sum_{t=1}^{T} \mathbb{E}[R(\pi_t)]$. The information structure of π_t requires that the decision for period t, π_t, only relies on the history of the process until time $t - 1$. This is similar to the multi-armed bandit problem, but here the decision variable is continuous rather than a finite set.

10.2.1 Regret

A standard measure used in the literature for the performance of a policy is the regret incurred compared to the clairvoyant policy. More formally, we define the regret of a policy π_t to be the expected gap in revenue from the clairvoyant policy, given by

$$r_\pi(T) := \sum_{t=1}^{T} \mathbb{E}\left[R(p^*) - R(\pi_t)\right].$$

This is a learning and earning problem where the demand is learned on the fly with a tradeoff between exploration and exploitation, whose goal is to design a policy π_t for which $r_\pi(T)$ scaling gracefully as $T \to \infty$. Since $r_\pi(T)$ depends on the unknown function $d(p)$, we require the designed policy to perform well for a wide class $\mathcal{C}$ of functions, i.e., we seek for optimal policies in terms of the minimax regret

$$\inf_{\pi_t} \sup_{d \in \mathcal{C}} r_\pi(T).$$

Although it is usually impossible to find the exact policy that achieves the minimax regret, most authors focus on policies whose regret is at least comparable to (of the same order as) the minimax regret as $T \to \infty$. To state this more formally,

we will use the big O notation. We say that $f(T) = O(g(T))$ as $T \to \infty$ if there are constants C and t_0 such that $f(t) \leq Cg(t)$ for all $t \geq t_0$. We say that $f(T) = \Omega(g(T))$ as $T \to \infty$ if there are constants C and t_0 such that $f(t) \geq Cg(t)$ for all $t \geq t_0$. We say that $f(T) = O^*(g(T))$ if there constants C and t_0 such that $f(t) \leq Cg(t)p(t)$ for all $t \geq t_0$ for some poly-logarithmic factor $p(t)$ of order lower than $g(t)$, so the big O^* notation neglects multiplicative terms of lower order.

For the ample capacity case, it is possible to show under mild conditions that there is a pricing policy for parametric models based on the maximum likelihood framework that achieves regret $O(\sqrt{T})$. For the non-parametric case, it is possible to show that there is a policy that achieves regret $O(\log(T)^{1/2}\sqrt{T})$, and for both cases, the regret is at least $\Omega(\sqrt{T})$. In summary, under mild assumptions, there are policies that have regret $O^*(\sqrt{T})$ for both the parametric and non-parametric cases. These regret bounds are for models that ignore customer characteristics that are crucial for personalized pricing.

In this section, we describe a non-parametric model that allows for personalized pricing. In this framework, each consumer arrives with a vector x of covariates in a bounded d-dimensional hypercube which we take without loss of generality to be $[0, 1)^d$. The expected demand function $\mathbb{E}[D(p, x)] = d(p, x)$ depends both on the price p and on the covariate vector x. A clairvoyant policy would observe x and return $p(x) = \arg\max R(p, x)$, where $R(p, x) = pd(p, x)$ is the revenue at price p when the covariate vector is x.

The main result of this section is that under mild assumptions there is an algorithm that returns a policy with regret at most $O(\log(T)^2 T^{(2+d)/(4+d)})$ where d is the dimension of the covariate vector. We also show that all policies have regret at least $\Omega(T^{(2+d)/(4+d)})$, so there exist a policy that is $O^*(T^{(2+d)/(4+d}})$. Without covariates, $d = 0$, the regret is $O^*(\sqrt{T})$, matching the performance of earlier algorithms. As d increases the lower bound deteriorates and becomes nearly linear in T. This suggests that only the most salient covariates should be included in personalized pricing, perhaps after applying a dimension-reduction algorithm. Thus, there is a tradeoff between trying to exploit covariate information and minimizing the regret, particularly as d gets large.

10.2.2 Assumptions

For any convex subset $B \subset [0, 1)^d$, let $R_B(p) := \mathbb{E}[r(X, p)|X \in B]$, where the expectation is taken over the distribution of the covariate space.

Assumption 1 $D(p, x)$ is a Bernoulli random variable with mean $d(p, x) := \mathbb{E}[D(p, x)] \in [0, 1]$ for all $p \in [0, 1]$ and all $x \in [0, 1)^d$.

Assumption 2 The expected revenue function $R(p, x) = pd(p, x)$ is Lipschitz continuous, i.e., there exists $M_1 > 0$ such that $|R(x_1, p_1) - R(x_2, p_2)| \leq M_1(\|x_1 - x_2\|_2 + |p_1 - p_2|)$ for all $x_i \in [0, 1)^d$ and $p_i \in [0, 1]$ with $i = 1, 2$.

Assumption 3

3.1 The function $R_B(p)$ has a unique maximizer $p^*(B) \in [0, 1]$. Moreover, there exist uniform constants $M_2, M_3 > 0$ such that for all $p \in [0, 1]$, $M_2(p^*(B) - p)^2 > R_B(p^*(B)) - R_B(p) > M_3(p^*(B) - p)^2$.

3.2 The maximizer $p^*(B)$ of $R_B(p)$ is inside the interval

$$[\inf\{p^*(x) : x \in B\}, \sup\{p^*(x) : x \in B\}].$$

3.3 Let δ_B be the diameter of B. Then there exists a uniform constant $M_4 > 0$ such that $\sup\{p^*(x) : x \in B\} - \inf\{p^*(x) : x \in B\} \leq M_4\delta_B$.

Assumption 1 is very mild. Lipchitz continuity or similar smoothness conditions are common in the literature and are needed for past experiments to be informative. The intuition behind the third assumption is to consider a learning problem associated with B without covariates. Indeed, if we only know that $X \in B$, the learning objective would be $R_B(p)$, so the clairvoyant policy would set price $p^*(B)$ in each period where $X \in B$. Assumption 3 is satisfied by many of the parametric families studied in the literature. For example, in the linear case, we have $d(p, x) = \alpha' x - \beta p$, so $R_B(p) = p(\alpha' \mathbb{E}[X|X \in B] - \beta p)$ and $p^*(B) = \alpha' \mathbb{E}[X|X \in B]/2\beta$.

10.2.3 Preliminary Concepts

We start by defining a bin and its children.

Definition 10.1 A bin is a hyper-rectangle in the covariate space. More precisely, a bin is of the form

$$B = \{x : a_i \leq x_i < b_i,\ i = 1, \ldots, d\}$$

for $0 \leq a_i < b_i \leq 1, i = 1, \ldots, d$.

We can *split* a bin B by bisecting it in all the d dimensions to obtain 2^d *child* bins of B, all of equal size. For a bin B with boundaries a_i and b_i for $i = 1, \ldots, d$, its children are indexed by the 2^d vectors in $\{0, 1\}^d$. Indeed, for any $w \in \{0, 1\}^d$, we have the child

$$B_w = \left\{x : a_i \leq x_i < \frac{a_i + b_i}{2} \text{ if } w_i = 0, \quad \frac{a_i + b_i}{2} \leq x_i < b_i \text{ if } w_i = 1,\ i = 1, \ldots, d\right\}$$

that choses the first half of the range of component i if $w_i = 0$ and the second half if $w_i = 1$ for each $i = 1, \ldots, d$.

Denote the set of all child bins of B by $C(B) = \{B_w : w \in \{0,1\}^d\}$. Notice that $C(B)$ is a mutually exclusive and collectively exhaustive partition of B into 2^d child bins. For any $B' \in C(B)$, we refer to B as the *parent* bin of B', denoted by $P(B') = B$.

The adaptive binning and exploration (ABE) algorithm given below starts with bin $B_\emptyset = [0,1)^d$ and successively splits it as data is collected. Any bin B produced during the process is the *offspring* of $B_\emptyset$. Therefore, one can use a sequence of vectors in $\{0,1\}^d$, $w^1, w^2, \ldots, w^k$ to represent a bin that is build during the algorithm. The bin $B_{w^1,w^2,\ldots,w^k}$ refers to a bin that is obtained by k split operations of $B_\emptyset$. After the first split, we obtain B_{w^1} from $B_\emptyset$. When B_{w^1} is split, we obtain its child $B_{w^1 w^2}$ and so on. In the last operation, when $B_{w^1 \ldots w^{k-1}}$ is split, we obtain its child $B_{w^1 \ldots w^k}$. For such a bin, we define its *level* to be k, denoted by $l(B) = k$, with $l(B_\emptyset) = 0$.

At the end of the ABE algorithm, there is a partition, say $\mathcal{P}$, of the covariate space, and for each $B \in \mathcal{P}$, the function $R_B(p)$ is estimated from data for values of p in a grid partition of an interval $[p_B^l, p_B^h] \subset [0,1]$ produced by the algorithm. The algorithm then selects the price in this grid that maximizes the approximation of $R_B(p)$, which should be close to $p^*(B)$, and in turn close to $p(x)$ for $x \in B$ given the Lipchitz continuity assumptions. The intuition is that for large T, we should be able to get reliable estimates for fairly small bins, and the approximation should be very accurate. The algorithm tries to do this learning efficiently by judiciously deciding when to split bins. This is done by a set of discrete decisions (referred to as the *decision set* hereafter) for each bin in the partition.

The algorithm keeps a dynamic partition $\mathcal{P}_t$ of the covariate space consisting of offspring of $B_\emptyset$ in each period t, starting with $\mathcal{P}_0 = \{B_\emptyset\}$. Each bin in $\mathcal{P}_{t+1}$ has an ancestor (or itself) in $\mathcal{P}_t$. Each time a bin is partitioned, that bin is removed and replaced by all of its children. The process can also be interpreted as the sequential splitting of a *branching process* and relates to decision trees in statistical learning.

The decision set consists of equally spaced grid points of an interval associated with the bin. When a covariate X_t is generated inside a bin B, a price is chosen successively in a grid and is applied to X_t. The realized reward for this decision is recorded. When sufficient covariates are observed in B, the average reward for each price in the grid is recorded as an estimate of $R_B(p)$. The best price in the grid is the *empirically-optimal* decision and is close to $p^*(B)$ with high confidence.

Adaptive Binning and Exploration (ABE)

Step 1. Initialization

(A) Input: T, d

(B) Constants: M_1, M_2, M_3, M_4, σ

(C) Parameters: K and Δ_k, n_k, N_k for $k = 0, \ldots, K$

(D) Set partition: $\mathcal{P} \leftarrow \{B_\emptyset\}$, $p_l^{B_\emptyset} \leftarrow 0$, $p_u^{B_\emptyset} \leftarrow 1$, $\delta_{B_\emptyset} \leftarrow 1/(N_0 - 1)$, $\bar{Y}_{B,j}, N_{B_\emptyset,j} \leftarrow 0$ for $j = 0, \ldots, N_0 - 1, N(B_0) = 0, l(B_0) = 0$

Step 2. Learning and Earning

(A) For $t = 1$ to T do
(B) Observe X_t
(C) $B \leftarrow \{B \in \mathcal{P} : X_t \in B\}$
(D) $k \leftarrow l(B)$, $N(B) \leftarrow N(B) + 1$
(E) If $k < K$ then

(a) If $N(B) < n_k$ then
(b) $j \leftarrow N(B) - 1 \pmod{N_k}$
(c) $\pi_t \leftarrow p_l^B + j\delta_B$; apply π_t and observe revenue Z_t
(d) $\bar{Y}_{B,j} \leftarrow \frac{1}{N_{B,j}+1}(N_{B,j}\bar{Y}_{B,j} + Z_t)$, $N_{B,j} \leftarrow N_{B,j} + 1$
(e) Else
(f) $j^* \in \arg\max_{j\in\{0,1,\ldots,N_k-1\}}\{\bar{Y}_{B,j}\}$, $p^* \leftarrow p_l^B + j^*\delta_B$
(g) $\mathcal{P} \leftarrow (\mathcal{P} \setminus B) \cup C(B)$
(h) For $B' \in C(B)$

- $N(B') \leftarrow 0$
- $p_l^{B'} \leftarrow \max\{0, p^* - \Delta_{k+1}/2\}$; $p_u^{B'} \leftarrow \min\{1, p^* + \Delta_{k+1}/2\}$
- $\delta_{B'} \leftarrow (p_u^{B'} - p_l^{B'})/(N_{k+1} - 1)$
- $N_{B',j}, \bar{Y}_{B',j} \leftarrow 0$, for $j = 0, \ldots, N_{k+1} - 1$
- End For

(i) End If

(F) Else $\pi_t \leftarrow (p_l^B + p_u^B)/2$
(G) End If
(H) End For

The parameters for the algorithm include K, the maximal level of the bins, Δ_k the length of the interval for level-k bins, n_k the maximum number of covariates observed in a level-k bin, N_k the number of decisions to explore in the decision set of level-k bins (consisting in N_k evenly spaced points in the interval $[p_l^B, p_u^B]$ specified by the algorithm). We initialize with the root bin $\mathcal{P}_0 = \{B_\emptyset\}$. Its decision set spans the whole interval [0, 1] with N_0 equally spaced grid points. That is, the j-th decision is $j\delta_{B_\emptyset} := j/(N_0-1)$ for $j = 0, \ldots, N_0-1$. The initial average reward and the number of explorations already applied to the j-th decision are set to $\bar{Y}_{B_\emptyset,j} = N_{B_\emptyset,j} = 0$. We set $K = \lfloor \frac{\log(T)}{(d+4)\log(2)} \rfloor$, $\Delta_k = 2^{-k}\log(T)$, $N_k = \lceil \log(T) \rceil$, and

$$n_k = \max\left\{0, \left\lceil \frac{2^{4k+18}\sigma}{M_2^2 \log^3(T)}(\log(T) + \log(\log(T)) - (d+2)k\log(2)) \right\rceil\right\}.$$

To give a sense of their magnitudes, the maximal level of bins is $K \approx \log(T)/(d+4)$. The range of the decision set (Δ_k) is proportional to the edge length of the bin (2^{-k}). The number of decisions in a decision set is approximately $\log(T)$. Therefore, the grid size $\delta_B \approx 2^{-k}$ for a level-k bin B. The number of covariates to collect in a level-k bin B is roughly $n_k \approx 2^{4k}/\log(T)^2$. When k

is small, n_k can be zero according to the expression. In this case, the algorithm immediately splits the bin without collecting any covariate in it.

Suppose the partition is $\mathcal{P}_t$ at t and a covariate X_t is generated (Step B). The algorithm determines the bin $B \in \mathcal{P}_t$ which the covariate falls into. The counter $N(B)$ records the number of covariates already observed in B up to t when B is in the partition (Step C). If the level of B is $l(B) = k < K$ (i.e., B is not at the maximal level) and the number of covariates observed in B is not sufficient, then the algorithm further explores and test prices in the decision set $\{p_l^B + j\delta_B\}$ for $j = 0, \ldots, N_k - 1$. There are N_k decisions in the set and they are equally spaced in the interval $[p_l^B, p_u^B]$. They are explored sequentially as new covariates are observed in B. The algorithm applies decision $\pi_t = p_l^B + j\delta_B$ where $j = N(B) - 1 \pmod{N_k}$ to the $N(B)$-th covariate observed in B (Step b). Step d updates the average reward and the number of explorations for the j-th decision.

If the level of B is $l(B) = k < K$ and we have observed sufficient covariates in B (Step e), then the algorithm splits B and replaces it by its 2^d child bins in the partition (Step g). For each child bin, Step h initializes the counter, the interval that encloses the decision set, the grid size of the decision set, and the average reward/number of explorations that have been conducted for each decision in the decision set. In particular, to construct the decision set of a child bin, the algorithm first computes the empirically optimal decision in the decision set of the parent bin B; that is, $j^* \in \arg\max_{j \in \{0,1,\ldots,N_k-1\}} \{\bar{Y}_{B,j}\}$ in Step f. Then, the algorithm creates an interval centered at this optimal decision with width Δ_{k+1}, properly cut off by the boundaries [0, 1]. The decision set is then an equally spaced grid of the above interval. If the level of B is already K, then the algorithm simply applies a single decision inherited from its parent (Step F) repeatedly without further exploration. For such a bin, its size is sufficiently small and the algorithm has narrowed the range of the decision set K times. The applied decision, which is the middle point of the interval, is close enough to all $p^*(x)$, $x \in B$, with high probability.

The following result provides upper and lower bounds on the regret.

Theorem 10.2 *For any function $R(p, x)$ satisfying Assumptions 1–3, the regret incurred by the ABE algorithm is bounded by*

$$r_{\pi_{ABE}}(T) \leq K T^{\frac{2+d}{4+d}} \log(T)^2$$

for a constant $K > 0$ that is independent of T. For all non-anticipating polices, we have

$$\inf_{\pi} \sup_{f \in \mathcal{C}} R_\pi \geq k T^{\frac{2+d}{4+d}}$$

for a constant $k > 0$ that is independent of T.

We illustrate the key steps of the algorithm by an example with $d = 2$. Figure 10.1 illustrates a possible outcome of the algorithm in periods $t_1 < t_2 < t_3$ (top panel, mid panel, and bottom panel, respectively). Up until period t_1, there is

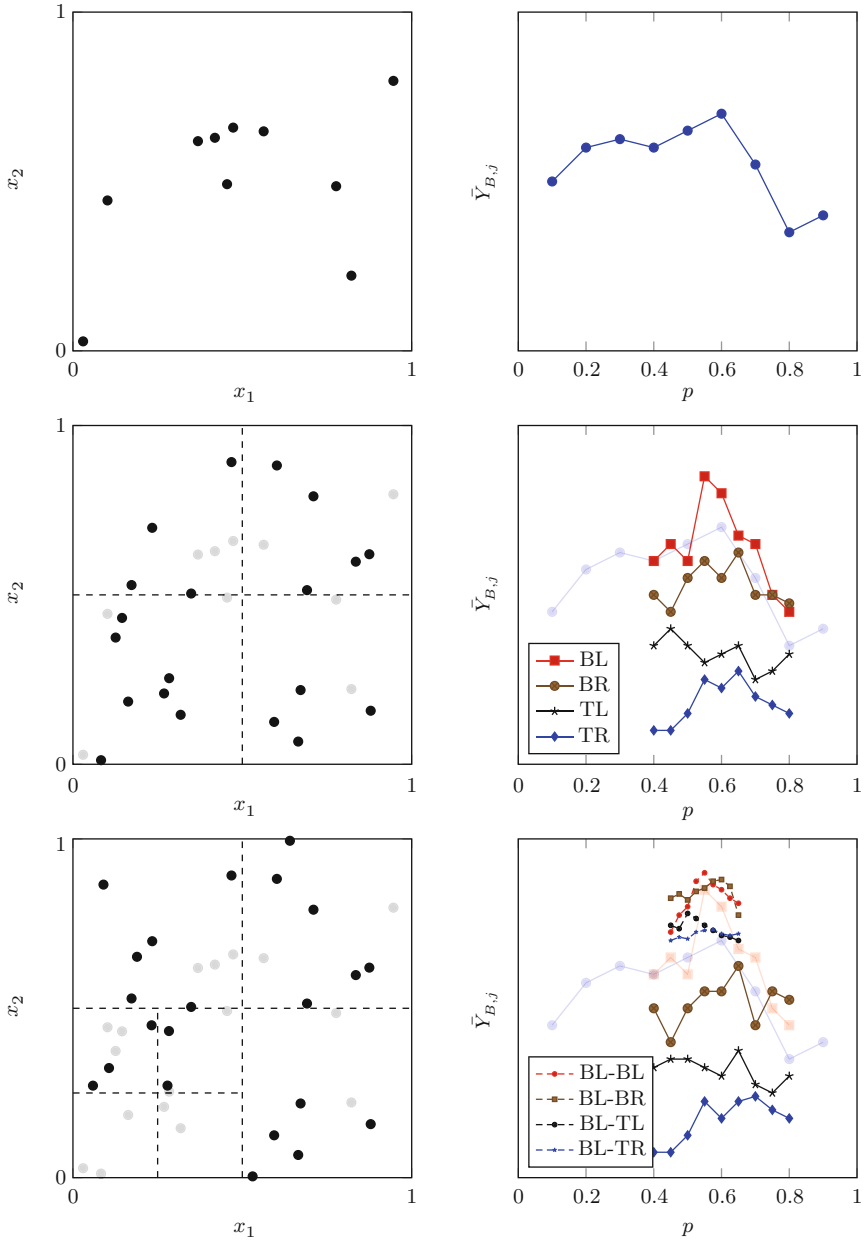

Fig. 10.1 A schematic illustration of the ABE algorithm

a single bin and the observed values X_t for $t \leq t_1$ are illustrated in the top left panel. The algorithm has explored the objective in the decision set, in this case $p \in \{0.1, 0.2, \ldots, 0.9\}$, and recorded the average reward $\bar{Y}_{B,j}$, illustrated by the top right panel. At $t_1 + 1$, sufficient observations are collected and Step e is triggered in the algorithm. Therefore, the bin is split into four child bins.

From period $t_1 + 1$ to t_2, new covariates are observed in each child bin (mid left panel). Note that the covariates generated before t_1 in the parent bin are no longer used and colored in gray. For each child bin (the bottom-left bin is abbreviated as BL and similarly for others), the average reward for the decision set is demonstrated in the mid right panel. The decision sets are centered at the empirically optimal decision of their parent bin, in this case $p^* = 0.6$ from the top right panel. They have narrower ranges and finer grids than that of the parent bin. At $t_2 + 1$, sufficient observations are collected for BL, and it is split into four child bins.

From period $t_2 + 1$ to t_3, the partition consists of seven bins, as shown in the bottom left panel. The BR, TL, and TR bins keep collecting covariates and updating the average reward, because they have not collected sufficient data. Their status at t_3 is shown in the bottom panels. In the four newly created child bins of BL (the bottom-left bin of BL is abbreviated as BL-BL and similarly for others), the decisions in the decision sets are applied successively and their average rewards are illustrated in the bottom right panel.

10.3 Constrained Inventory Model

Constrained inventory models require a slightly more careful analysis. Most of the models assume that the demand is either a Bernoulli or a Poisson process. In the Bernoulli process case, the inventory is ample whenever $c \geq T$ since in this case capacity exceeds potential demand. In the Poisson case, things are more subtle as the demand over the sales horizon at price is $Td(p)$ where $d(p)$ is the sales rate at price p, and $d(p)$ may be larger than one. If $d(p)$ is bounded above, a change of variables $T \leftarrow aT$ and $d(p) \leftarrow d(p)/a$ for a sufficiently large a results in $d(p) \ll 1$ for all p. In this case, the Poisson process can be closely approximated by a Bernoulli process. With this scaling, the ample inventory model corresponds to the case $\rho := c/T \geq 1$, and the constrained inventory model to the case $\rho < 1$. For a fixed ρ, we are interested in the learning and earning problem as $T \to \infty$, which means that initial inventory $c = \rho T < T$ also grows at the same rate. If we want to keep c integer (and this is important for the formulation of the dynamic pricing problem with finite capacity), we can restrict T to be of the form $T = nc/\rho$, where n is an integer so $\rho T = nc$. In the literature, we often see a different but equivalent scaling mechanism for the Poisson case, where T is held fixed (often normalized to one) and c and $d(p)$ are scaled up by a factor n, so the initial inventory is nc and the demand rate is $nd(p)$. Most authors use the second scaling method $(nc, nd(p))$, but it should be clear to the reader that an algorithm with regret $O^*(\sqrt{n})$ has regret $O(\sqrt{T})$ under the first scaling method. In this section, we follow the literature and report the regret relative to the scaling $(n, nd(p))$ and report the regret in terms of n.

Early work on the constrained inventory model divided the horizon into a learning phase and an earning phase. Under these regime, it was possible to show that all policies have regret at least $\Omega(\sqrt{n})$, and that there exist a policy with regret $O(\sqrt{\log(n)}n^{2/3})$ for the parametric case, and regret $O(\sqrt{\log(n)}n^{3/4})$ for the non-parametric case. In this section, we review recent work that intertwines learning and earning and improves the regret to $O(\log(n)^{4.5}\sqrt{n})$ for both the parametric and non-parametric case, thus also achieving an $O^*(\sqrt{n})$ policy, which is equivalent to the results for the ample inventory model stated in terms of T.

The analysis is based on slightly different assumptions from the case with ample capacity. The most salient difference is that demand is assumed to be a Poisson process with rate $\lambda_t = d(p_t)$, where $d(p)$ is the expected demand rate that is strictly decreasing in p. This in contrast to the ample capacity case where the analysis is based on a binomial approximation to the Poisson. There is an inverse demand function given by $p = \gamma(\lambda)$, and a corresponding revenue rate $R(\lambda) = \lambda\gamma(\lambda)$ function written in terms of the sales rate instead of the price.

The analysis allows for both parametric and non-parametric models with constrained capacity and requires the following assumptions on the class $\mathcal{C}$ of admissible demand functions.

Assumption 1 Boundedness: $|d(p)| < M$ for all $p \in [0, 1] \cup \{p_\infty\}$ with $d(p_\infty) = 0$.

Assumption 2 Lipschitz continuity: $d(p)$ and $R(d(p)) = pd(p)$ are Lipschitz continuous functions with respect to p with constant K. The inverse demand function $\gamma(\lambda)$ is also Lipschitz continuous with constant K.

Assumption 3 Strict concavity and differentiability: The function $R(\lambda) = \lambda\gamma(\lambda)$ has a second derivative for all λ, and there are positive constants such that $-m_L \leq R''(\lambda) \leq -m_U < 0$ for all $p \in [0, 1]$.

The assumptions are all reasonable in light of our previous discussion of the case of ample capacity. The most significant difference here is the existence of a cutoff price p_∞ such that $d(p_\infty) = 0$. This is a modeling artifact that provides us a price to use when the system runs out of inventory.

For any d satisfying Assumptions 1–3, let $V_\pi(T, c; d)$ denote the expected revenue that can be obtained from c units of inventory over the selling horizon $[0, T]$ by applying an non-anticipating policy π. From our analysis of dynamic pricing, we know that for every demand function d,

$$V_\pi(T, c; d) \leq V(T, c; d) \leq \bar{V}(T, c : d),$$

where $V(T, c : d)$ is the maximum expected revenue under any non-anticipating policy and $\bar{V}(T, c; d)$ is the upper bound based on replacing demand by its expectations. Rather than measuring the regret of a policy π by $V(T, c; d) - V_\pi(T, c; d)$, in this section we measure the regret relative to the deterministic upper bound, resulting in

$$r_\pi(T, c; d) = \bar{V}(T, c; d) - V_\pi(T, c; d).$$

Normalizing $T = 1$, the regret is

$$r_\pi(c; d) = \bar{V}(c; d) - V_\pi(c; d),$$

where $T = 1$ is omitted for convenience. The goal is to minimize the worst-case regret, which is given by

$$\inf_\pi \sup_{d \in \mathcal{C}} r_\pi(nc; nd)$$

as n increases, where the infimum is taken over any non-anticipating policy and any demand function d satisfying Assumptions 1–3.

Learning and Dynamic Pricing (LDP)

Step 1. Initialization

(a) Consider a sequence of $\tau_i^u, \kappa_i^u, i = 1, 2, \ldots, N^u$ and $\tau_i^c, \kappa_i^c, i = 1, 2, \ldots, N^c$ Define $\underline{p}_1^u = \underline{p}_1^c = 0$ and $\overline{p}_1^u = \overline{p}_1^c = 1$. Define $t_i^u = \sum_{j=1}^i \tau_j^u$, for $i = 1$ to N^u and $t_i^c = \sum_{j=1}^i \tau_j^c$, for $i = 1$ to N^c;

Step 2. Learn p^u or Determine $p^c > p^u$

For $i = 1$ to N^u do

(a) Divide $[\underline{p}_i^u, \overline{p}_i^u]$ into κ_i^u equally spaced intervals and let $\{p_{i,j}^u, j = 1, 2, \ldots, \kappa_i^u\}$ be the left endpoints of these intervals;

(b) Divide the time interval $[t_{i-1}^u, t_i^u]$ into κ_i^u equal parts and define

$$\Delta_i^u = \frac{\tau_i^u}{\kappa_i^u}, \qquad t_{i,j}^u = t_{i-1}^u + j\Delta_i^u, \qquad j = 0, 1, \ldots, \kappa_i^u;$$

(c) For j from 1 to κ_i^u, apply $p_{i,j}^u$ from time $t_{i,j-1}^u$ to $t_{i,j}^u$. If inventory runs out, then apply p_∞ until time T and STOP;

(d) Compute

$$\hat{d}(p_{i,j}^u) = \frac{\text{total demand over } [t_{i,j-1}^u, t_{i,j}^u)}{\Delta_i^u}, \quad j = 1, \ldots, \kappa_i^u;$$

(e) Compute

$$\hat{p}_i^u = \arg\max_{1 \le j \le \kappa_i^u} \{p_{i,j}^u \hat{d}(p_{i,j}^u)\} \quad \text{and} \quad \hat{p}_i^c = \arg\min_{1 \le j \le \kappa_i^u} \left| \hat{d}(p_{i,j}^u) - x/T \right|;$$

(f) If

$$\hat{p}_i^c > \hat{p}_i^u + 2\sqrt{\log n} \cdot \frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}$$

then break from Step 2, enter Step 3 and set $i_0 = i$;

Otherwise, set $\hat{p}_i = \max\{\hat{p}_i^c, \hat{p}_i^u\}$. The price range for the next iteration is given by

$$I_{i+1}^u = [\underline{p}_{i+1}^u, \overline{p}_{i+1}^u],$$

where

$$\underline{p}_{i+1}^u = \hat{p}_i - \frac{\log n}{3} \cdot \frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u} \text{ and } \overline{p}_{i+1}^u = \hat{p}_i + \frac{2\log n}{3} \cdot \frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}.$$

We truncate the interval if it does not lie inside the feasible set $[0, 1]$;

(g) If $i = N^u$, then enter Step 4(a);

Step 3. Learn p^c When $p^c > p^u$
For $i = i_0$ to N^c do

(a) Divide $[\underline{p}_i^c, \overline{p}_i^c]$ into κ_i^c equally spaced intervals and let $\{p_{i,j}^c, j = 1, 2, \ldots, \kappa_i^c\}$ be the left endpoints of these intervals;

(b) Define

$$\Delta_i^c = \frac{\tau_i^c}{\kappa_i^c}, \qquad t_{i,j}^c = t_{i-1}^c + j\Delta_i^c + t_{i_0}^u, \qquad j = 0, 1, \ldots, \kappa_i^c;$$

(c) For j from 1 to κ_i^c, apply $p_{i,j}^c$ from time $t_{i,j-1}^c$ to $t_{i,j}^c$. If inventory runs out, then apply p_∞ until time T and STOP;

(d) Compute

$$\hat{d}(p_{i,j}^c) = \frac{\text{total demand over } [t_{i,j-1}^c, t_{i,j}^c)}{\Delta_i^c}, \quad j = 1, \ldots, \kappa_i^c;$$

(e) Compute

$$\hat{q}_i = \arg \min_{1 \le j \le \kappa_i^c} \left| \hat{d}(p_{i,j}^c) - x/T \right|.$$

The price range for the next iteration is given by

$$I_{i+1}^c = [\underline{p}_{i+1}^c, \overline{p}_{i+1}^c],$$

where

$$\underline{p}_{i+1}^c = \hat{q}_i - \frac{\log n}{2} \cdot \frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c} \text{ and } \overline{p}_{i+1}^c = \hat{q}_i + \frac{\log n}{2} \cdot \frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c},$$

and we truncate the interval if it doesn't lie inside the feasible set of $[0, 1]$;

(f) If $i = N^c$, then enter Step 4(b);

Step 4. Apply the Learned Price

(a) Define $\tilde{p} = \hat{p}_{N^u} + 2\sqrt{\log n} \cdot \frac{\overline{p}^u_{N^u} - \underline{p}^u_{N^u}}{\kappa^u_{N^u}}$. Use $\tilde{p}$ for the rest of the selling season until the inventory runs out;

(b) Define $\tilde{q} = \hat{q}_{N^c}$. Use $\tilde{q}$ for the rest of the selling season until the inventory runs out.

A few comments about the LDP algorithm are in order. The selling season is divided into a set of time periods. In each time period, a set of a grid prices is tested within the current price interval. The intervals are then updated based on empirical observations at the end of each time interval, so the price intervals contain the optimal price with high probability. The process is repeated until the price interval is small enough so that the desired accuracy is achieved.

The optimal price is the largest between the unconstrained optimal price, say p^*, and the market clearing price, say p_{mc}. Finding these two prices require different shrinking strategies for the cases when $p^* > p_{mc}$ (Step 2) and $p_{mc} > p^*$ (Step 3). At the end of the algorithm, a fixed price is used for the remaining selling season (Step 4) until the inventory runs out.

The definitions of τ^u_i, κ^u_i, N^u, τ^c_i, κ^c_i, and N^c now follow, where $T = 1$ without loss of generality:

$$\left(\frac{\overline{p}^u_i - \underline{p}^u_i}{\kappa^u_i}\right)^2 \sim \sqrt{\frac{\kappa^u_i}{n\tau^u_i}}, \quad \forall i = 2, \ldots, N^u,$$

$$\overline{p}^u_{i+1} - \underline{p}^u_{i+1} \sim \log n \cdot \frac{\overline{p}^u_i - \underline{p}^u_i}{\kappa^u_i}, \quad \forall i = 1, \ldots, N^u - 1,$$

$$\tau^u_{i+1} \cdot \left(\frac{\overline{p}^u_i - \underline{p}^u_i}{\kappa^u_i}\right)^2 \cdot \sqrt{\log n} \sim \tau^u_1, \quad \forall i = 1, \ldots, N^u - 1,$$

$$N^u = \min_l \left\{ l : \left(\frac{\overline{p}^u_l - \underline{p}^u_l}{\kappa^u_l}\right)^2 \sqrt{\log n} < \tau^u_1 \right\}.$$

The main results for this section is the following.

Theorem 10.3 *For any function demand $d(p)$ satisfying Assumptions 1–3, the regret incurred by the LDP algorithm is bounded by*

$$\sup_{d \in \mathcal{C}} r_{\pi_{DPA}}(nc, nd) \leq K\sqrt{n}\log(n)^{4.5}$$

for a constant $K > 0$ that is independent of n for both the parametric and the non-parametric cases. For all non-anticipating polices, we have

$$\inf_{\pi} \sup_{f \in \mathcal{C}} r_{\pi}(nc, nd) \geq k\sqrt{n}$$

for a constant $k > 0$ that is independent of n.

Under mild assumptions, the results in this section can be extended to multiple market segments $d_m(p), m = 1, \ldots, M$ using a primal-dual approach.

10.4 Bibliographical Remarks

There is a large and growing literature for parametric models that follows a dynamic programming formulation with Bayesian updating. Some examples in this stream of literature include the work by Aviv and Pazgal (2005), Bertsimas and Mersereau (2007), Araman and Caldentey (2009), Sen and Zhang (2009), Farias and Van Roy (2010), and Harrison et al. (2012). Bayesian methods require the specification of a prior distribution that belongs to a conjugate family, and the method is mostly use for the case of a single unknown parameter with a few notable exceptions. Alternatives to the Bayesian approach that are capable of dealing with a large number of parameters involve maximum likelihood methods and least squares as in Bertsimas and Perakis (2006) and Bertsimas and Misic (2019). Araman and Caldentey (2011) go over both Bayesian and non-parametric models. The survey paper by den Boer (2015) provides a comprehensive overview of this area.

There is an alternative stream of literature, closer to the results described in this chapter, that focus on the learning and earning problem to minimize the worst-case regret. This literature can be divided into the case of ample or constrained inventory. Some of the contributors to this literature come from the computer science literature, including Kleinberg and Leighton (2003), who provide analysis for an online posted-price auction for the case of ample capacity obtaining a regret of $O(\log T\sqrt{T})$ for a non-parametric model. The paper by Broder and Rusmevichientong (2012) provides an algorithm with regret $O(\log T\sqrt{T})$ for a parametric model based on maximum likelihood. They also show that the regret can be improved to $O(\log T)$ for situations where the demand functions can be separated. Cheung and Simchi-Levi (2017) also look into an infinite inventory model but restrict the class of demand functions to a finite set and allow a maximum of m price changes. They achieve a regret of $O((\log T)^m)$ under the assumption that exploration is done with informative prices. Besbes and Zeevi (2015) show that even if the demand is misspecified as linear, a regret of $O((\log T)^2\sqrt{T})$ can be surprisingly achieved under mild restrictions. None of the models mentioned allow for covariates.

The results presented in this chapter for the ample capacity case are due to Chen and Gallego (2018a). For the constrained capacity case, an important early reference is Besbes and Zeevi (2009), where learning and earning are separated into two phases. The section on constrained inventory is based on Wang et al. (2014), where more refined results are obtained by intertwining learning and earning. The extension of the constrained model to multiple market segments is due to Chen and Gallego (2018b). A related model that is applicable to multi-resource revenue management problems is given in Agrawal et al. (2014). Chen et al. (2019b, 2016a) discuss models for joint inventory and pricing decisions, when the price-demand

relationship is unknown. There is also work on learning the customer preferences in assortment optimization problems, as exemplified by Saure and Zeevi (2013), Agrawal et al. (2018), Chen and Wang (2018), and Chen et al. (2018b,c).

Other works that focus on learning the price-demand relationship while making pricing decisions and earning revenues include Levina et al. (2009b), Besbes and Zeevi (2011), Kwon et al. (2012), Besbes and Saure (2014), Keskin and Zeevi (2014), and Nambiar et al. (2019). Ciocan and Farias (2012a) give bounds on the performance of a policy that is based on re-solving a mathematical program and updating the demand forecast. Ciocan and Farias (2014), Ban and Keskin (2017), Javanmard and Nazerzadeh (2018), and Cohen et al. (2018a) learn parameterized relationships for the demand for a product that are based on features. Keskin and Zeevi (2017) consider learning the price-demand relationship when this relationship is changing over time. A related model also appears in Besbes et al. (2015). The paper by den Boer and Keskin (2017a) focus on learning a discontinuous price-demand relationship, whereas den Boer and Keskin (2017b) study the case where there is an observable kink in the price-demand relationship. Keskin and Birge (2019) study a model where the firm learns the quality sensitivity of its customers and demonstrate that myopic policies can display near-optimal performance. A Bayesian approach based on Thompson sampling is given in Ferreira et al. (2018). Chen et al. (2019a) characterize the revenue loss of a policy that learns the multi-product demand function while making decisions. Afeche and Ata (2013) study a Bayesian learning model for a pricing problem in the queueing setting, where the proportion of patient customers needs to be learned. Lastly, Acemoglu et al. (2011), Crapis et al. (2017), and Ifrach et al. (2018) focus on learning problems within social networks.