Learning and Pricing with Inventory Constraints (Selected)

Silin DU

Department of Management Science and Engineering
School of Economics and Management
Tsinghua University
dsl21@mails.tsinghua.edu.cn

October 12, 2023

Contents

Background

- We consider learning and pricing problems with inventory constraints: given an *initial inventory* of one or multiple products and *finite* selling season, a seller must choose prices dynamically to maximize revenue over the course of the season.
- Dynamic pricing with inventory constraints: the demand function is *known* to the seller prior to the selling season.
- When the demand function is unknown, the seller faces a trade-off commonly referred to as the *exploration-exploitation* trade-off. With limited inventory, pursuing the exploration objective comes at the cost of not only lowering revenue but also diminishing valuable inventory.

Contents

Settings I

- Consider a monopolist selling a single product in a finite selling season T.
- A fixed inventory x at the beginning and no replenishment.
- Customers arrive according to a Poisson process with an instantaneous demand rate λ_t at time t.
- Assume that λ_t is solely dependent on the price the seller offers at time t, i.e., $\lambda_t = \lambda(p(t))$.
- No salvage value.
- Assume that the set of feasible prices is an interval $[\overline{p}, \underline{p}]$ with an additional cut-off price p_∞ such that $\lambda(p_\infty) = 0$.
- The demand rate function $\lambda(p)$ is assumed to be strictly decreasing in p and has an inverse function $p = \gamma(\lambda)$.

Settings II

- We define a revenue rate function $r(\lambda) = \lambda\gamma(\lambda)$, which captures the expected revenue when the price is chosen such that the demand is λ. We further assume $r(\lambda)$ is concave in λ.

We make the following assumptions on the demand rate function $\lambda(p)$ and the revenue rate function $r(\lambda)$.

Settings III

Assumption 2.1

For some positive constants M, K, m_L and m_U,

1. Boundedness: $|\lambda(p)| \leq M$ *for all* $p \in [\overline{p}, \underline{p}]$.
2. Lipschitz continuity: $\lambda(p)$ *and* $r(\lambda(p))$ *are Lipschitz continuous with respect to* p *with factor* K. *Also, the inverse demand function* $p = \gamma(\lambda)$ *is Lipschitz continuous in* λ *with factor* K.
3. Strict concavity and differentiability: $r''(\lambda)$ *exists and* $-m_L \leq r''(\lambda) \leq -m_U < 0$ *for all* λ *in the range of* $\lambda(p)$ *for* $p \in [\overline{p}, \underline{p}]$.

- We use $\Gamma = \Gamma(M, K, m_L, m_U)$ to denote the set of demand functions satisfying the above assumptions with the corresponding coefficients.

Settings IV

- Assumption 2.1 is satisfied for many commonly used demand function classes including linear, exponential, and logit demand functions.

To evaluate the performance of any pricing algorithm, we adopt the minimax regret objective. We call a pricing policy $\pi = (p(t) : 0 \leq t \leq T)$ *admissible* if

1. it is a non-anticipating price process that is defined on $[\overline{p}, \underline{p}] \cup \{p_\infty\}$
2. it satisfies the inventory constraint, that is, $\int_0^T dN^\pi(s) \leq x$ with probability 1, where $N^\pi(t) = N\left(\int_0^t \lambda(p(s))ds\right)$ denotes the cumulative demand up to time t using policy π.

Settings V

We denote the set of admissible pricing policies by $\mathcal{P}$. Then we define the expected revenue generated by a policy π by

$$J^{\pi}(x, T; \lambda) = E\left[\int_0^T p(s) dN^{\pi}(s)\right] \tag{2.1}$$

Since we do not know λ in advance, we seek $\pi \in \mathcal{P}$ that performs as close to π^* as possible. However, even if the demand function λ is known, computing the expected value of the optimal policy is computationally prohibitive.

Settings VI

- Gallego and van Ryzin (1994) showed that $J^{\pi}(x,T;\lambda) \leq J^{D}(x,T;\lambda)$ for all $\lambda \in \Gamma, \pi \in \mathcal{P}$, where

$$\begin{aligned} J^{D}(x,T;\lambda) = \sup & \int_0^T r(\lambda(p(s)))ds \\ \text{s.t.} & \int_0^T \lambda(p(s))ds \leq x \\ & p(s) \in [\underline{p},\overline{p}] \cup \{p_\infty\}, \quad \forall s \in [0,T]. \end{aligned} \tag{2.2}$$

- We define the regret $R^{\pi}(x,T;\lambda)$ for any given demand function $\lambda \in \Gamma$ and policy $\pi \in \mathcal{P}$ by

$$R^{\pi}(x,T;\lambda) = 1 - \frac{J^{\pi}(x,T;\lambda)}{J^{D}(x,T;\lambda)}. \tag{2.3}$$

Settings VII

- We consider the worst-case regret

$$\inf_{\pi \in \mathcal{P}} \sup_{\lambda \in \Gamma} R^{\pi}(x, T; \lambda). \tag{2.4}$$

  It is hard to evaluate (2.4) for any finite size problem. We adopt a widely used asymptotic performance analysis.

- We consider a regime in which both the size of the initial inventory and the demand rate grow proportionally. Specifically, in a problem with size n, the initial inventory and the demand function are given by

$$x_n = nx \text{ and } \lambda_n(\cdot) = n\lambda(\cdot).$$

Settings VIII

- Define $J_n^D(x, T; \lambda) = J^D(nx, T, n\lambda) = nJ^D(x, T, \lambda)$ to be the optimal value to the deterministic problem with size n and $J_n^\pi(x, T; \lambda) = J^\pi(nx, T, n\lambda)$ to be the expected value of a pricing policy π when it is applied to a problem with size n.
- The regret for the size-n problem $R_n^\pi(x, T; \lambda)$ is

$$R_n^\pi(x, T; \lambda) = 1 - \frac{J_n^\pi(x, T; \lambda)}{J_n^D(x, T; \lambda)},$$

and our objective is to study the asymptotic behavior of $R_n^\pi(x, T; \lambda)$ as n grows large and design an algorithm with small asymptotic regret.

Full-information I

We first consider the full-information deterministic problem (2.2). Besbes and Zeevi (2009) showed the optimal solution is given by

$$p(t) = p^D = \max\{p^u, p^c\} \tag{2.5}$$

where

$$p^u = \arg\max_{p\in[\underline{p},\overline{p}]}\{r(\lambda(p))\}, \quad p^c = \arg\min_{p\in[\underline{p},\overline{p}]}\left|\lambda(p) - \frac{x}{T}\right|. \tag{2.6}$$

Lemma 2.1 (Gallego and van Ryzin (1994))

Let p^D be the optimal deterministic price when the underlying demand function is λ. Let π^D be the pricing policy that uses the deterministic optimal price p^D throughout the selling season until there is no inventory left. Then, $R_n^{\pi^D}(x, T, \lambda) = O\left(n^{-1/2}\right)$.

Full-information II

- Lemma 2.1 states that if one knows p^D in advance, then simply applying this price throughout the entire time horizon can achieve asymptotically optimal performance.
- Therefore, the idea of our algorithm is to find an estimate of p^D that is *close* enough to the true one *efficiently*, using empirical observations on hand.

In particular, under Assumption 2.1, we know that if $p^D = p^u > p^c$, then

$$\left|r(p) - r\left(p^D\right)\right| \leq \frac{1}{2} m_L \left(p - p^D\right)^2 \tag{2.7}$$

for p close to p^D, while if $p^D = p^c \geq p^u$, then

$$\left|r(p) - r\left(p^D\right)\right| \leq K \left|p - p^D\right| \tag{2.8}$$

for p close to p^D. In the following discussion, without loss of generality, we assume $p^D \in (\underline{p}, \overline{p})$.

Main Result I

$f(n) = O^*(g(n))$ means there is a constant C and k such that $f(n) \leq C \cdot g(n) \cdot \log^k n$.

Theorem 2.1

Let Assumption 2.1 hold for $\Gamma = \Gamma(M, K, m_L, m_U)$. Then, there exists an admissible policy π generated by Dynamic Pricing Algorithm (DPA) , such that for all $n \geq 1$,

$$\sup_{\lambda \in \Gamma} R_n^{\pi}(x, T; \lambda) = O^*\left(n^{-1/2}\right).$$

- The result in Theorem 2.1 is the best asymptotic regret that one can achieve in this setting.

Main Result II

Corollary 2.1

Assume that Γ is a parameterized demand function family satisfying Assumption 2.1. Then, there exists an admissible policy π generated by Dynamic Pricing Algorithm (DPA) , such that for all $n \geq 1$,

$$\sup_{\lambda \in \Gamma} R_n^{\pi}(x, T; \lambda) = O^*\left(n^{-1/2}\right).$$

- There is *no performance gap* between parametric and nonparametric settings in the asymptotic sense, implying that the value of knowing the parametric form of the demand function is marginal in this problem.

Dynamic Pricing Algorithm I

- We aim to learn p^D through price experimentations.
- The algorithm will be able to distinguish whether p^u or p^c is optimal.
- It keeps a shrinking interval containing the optimal price with high probability until a certain accuracy is achieved.

Ideas

1. We divide the selling season into a *carefully selected set of time periods*.
2. In each time period, we *test a set of prices* within a certain price interval.
3. Based on the empirical observations, we *shrink the price interval* to a smaller subinterval that still contains the optimal price with high probability and enter the next time period with a smaller price range.
4. We repeat the shrinking procedure until the price interval is small enough so that the desired accuracy is achieved.

Dynamic Pricing Algorithm II

4 Steps:

1. Initialization
2. Learn p^u or determine $p^c > p^u$
3. Learn p^c when $p^c > p^u$
4. Apply the learned price

Remarks

- As shown in (2.7) and (2.8), the local behavior of the revenue rate function is quite different around p^u and p^c : the former one resembles a quadratic function, while the latter one resembles a linear function.
- This difference requires us to use different shrinking strategies for the cases when $p^u > p^c$ and $p^c > p^u$. This is why we have two learning steps (Steps 2 and 3) in our algorithm.

Dynamic Pricing Algorithm III

- In Step 2, the algorithm works by shrinking the price interval until either a *transition condition* is triggered or the learning phase is terminated.
- When the transition condition (2.10) is triggered, with high probability, the optimal solution to the deterministic problem is p^c.
- Otherwise, if we terminate learning before the condition is triggered, we know that p^u is either the optimal solution to the deterministic problem or it is close enough so that using p^u can yield a near-optimal revenue.
- When (2.10) is triggered, we switch to Step 3, in which we use a new set of shrinking and price testing parameters.

Step 1: Initialization

- Consider a sequence of $\tau_i^u, \kappa_i^u, i = 1, 2, \ldots, N^u$ and $\tau_i^c, \kappa_i^c, i = 1, 2, \ldots, N^c$ where τ and κ represent *the length of each learning period* and *the number of different prices to be tested* in each learning period, respectively. Their values along with the values of N^u and N^c are defined in (2.22)-(2.27), (2.17) and (2.21)
- Define $\underline{p}_1^u = \underline{p}_1^c = \underline{p}$ and $\overline{p}_1^u = \overline{p}_1^c = \overline{p}$.
- Define $t_i^u = \sum_{j=1}^{i} \tau_j^u$, for $i = 0$ to N^u and $t_i^c = \sum_{j=1}^{i} \tau_j^c$, for $i = 0$ to N^c.

Step 2: Learn p^u or Determine $p^c > p^u$ I

For $i = 1$ to N^u do

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

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

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

(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;

Step 2: Learn p^u or Determine $p^c > p^u$ II

(d) Compute

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

(e) Compute

$$\hat{p}_i^u = \arg\max_{1 \le j \le \kappa_i^u} \left\{ p_{i,j}^u \hat{d}\left(p_{i,j}^u\right) \right\} \quad \text{and} \quad \hat{p}_i^c = \arg\min_{1 \le j \le \kappa_i^u} \left| \hat{d}\left(p_{i,j}^u\right) - x/T \right|; \tag{2.9}$$

Step 2: Learn p^u or Determine $p^c > p^u$ III

(f) If

$$\hat{p}_i^c > \hat{p}_i^u + 2\sqrt{\log n} \cdot \frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u} \tag{2.10}$$

then break from Step 2, enter Step 3 and set $i_0 = i$; Otherwise, set $\hat{p}_i = \max\left\{\hat{p}_i^c, \hat{p}_i^u\right\}$. Define

$$\underline{p}_{i+1}^u = \hat{p}_i - \frac{\log n}{3} \cdot \frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u} \quad \text{and} \quad \overline{p}_{i+1}^u = \hat{p}_i + \frac{2\log n}{3} \cdot \frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}. \tag{2.11}$$

And define the price range for the next iteration

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

Here we truncate the interval if it does not lie inside the feasible set $[\underline{p}, \overline{p}]$;

(g) if $i = N^u$, then enter Step 4(a).

Step 3: Learn p^c When $p^c > p^u$ I

For $i = 1$ to N^c do

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

(b) Define

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

(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

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

Step 3: Learn p^c When $p^c > p^u$ II

(e) Compute

$$\hat{q}_i = \arg \min_{1 \le j \le k_i^c} \left| \hat{d}\left(p_{i,j}^c\right) - x/T \right|. \tag{2.12}$$

Define

$$\underline{p}_{i+1}^c = \hat{q}_i - \frac{\log n}{2} \cdot \frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c} \quad \text{and} \quad \overline{p}_{i+1}^c = \hat{q}_i + \frac{\log n}{2} \cdot \frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c}. \tag{2.13}$$

And define the price range for the next iteration

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

Here, we truncate the interval if it does not lie inside the feasible set of $[\underline{p}, \overline{p}]$;

(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.

We assume $T = 1$ and $\overline{p} - \underline{p} = 1$. And we use the notation $f \sim g$ to mean that f and g are of the same order in n. The relations that we want $\left(\tau_i^u, \kappa_i^u\right)_{i=1}^{N^u}$ to satisfy are

$$\left(\frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}\right)^2 \sim \sqrt{\frac{\kappa_i^u}{n\tau_i^u}}, \quad \forall i = 2, \ldots, N^u, \tag{2.14}$$

$$\overline{p}_{i+1}^u - \underline{p}_{i+1}^u \sim \log n \cdot \frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}, \quad \forall i = 1, \ldots, N^u - 1, \tag{2.15}$$

$$\tau_{i+1}^u \cdot \left(\frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}\right)^2 \cdot \sqrt{\log n} \sim \tau_1^u, \quad \forall i = 1, \ldots, N^u - 1. \tag{2.16}$$

About τ, κ, N II

Also, we define

$$N^u = \min_l \left\{ l : \left(\frac{\overline{p}_l^u - \underline{p}_l^u}{\kappa_l^u} \right)^2 \sqrt{\log n} < \tau_1^u \right\} \tag{2.17}$$

- To understand the above relations, it is useful to examine the source of revenue losses in our algorithm.
- First, there is an *exploration loss* in each period—the prices tested are not optimal, resulting in suboptimal revenue rate or suboptimal inventory consumption rate.
- Second, there is a *deterministic loss* due to the limited learning capacity—we only test a grid of prices in each period and may never use the exact optimal price.

About τ, κ, N III

- Third, since the demand follows a stochastic process, the observed demand rate may deviate from the true underlying demand rate, resulting in a *stochastic loss*.
- The design of our algorithm tries to balance these losses in each step to achieve the maximum efficiency of learning.
- The first relation (2.14) balances the deterministic loss induced by only considering the grid points (the grid granularity is $\frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u}$) and the stochastic loss induced in the learning period which will be shown to be $\sqrt{\frac{\kappa_i^u}{n\tau_i^u}}$.

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

About τ, κ, N IV

- The second relation (2.15) makes sure that with high probability, the price interval I_{i+1}^u contains the optimal price p^D.

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

- The third relation (2.16) bounds the exploration loss for each learning period. This is done by considering the multiplication of the revenue rate deviation (also demand rate deviation) and the length of the learning period, which in our case can be upper bounded by $\tau_{i+1}^u \sqrt{\log n} \cdot \left(\frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u} \right)^2$. We want this loss to *be of the same order* for each

learning period (and all equal to the loss in the first learning period, which is τ_1) to achieve the maximum efficiency of learning.

$$\tau_{i+1}^u \cdot \left(\frac{\overline{p}_i^u - \underline{p}_i^u}{\kappa_i^u} \right)^2 \cdot \sqrt{\log n} \sim \tau_1^u, \quad \forall i = 1, \ldots, N^u - 1$$

- Formula (2.17) determines when the price we obtain is close enough to optimal such that we can apply this price in the remaining selling season. We show that $\sqrt{\log n} \cdot \left(\frac{\overline{p}_l^u - \underline{p}_l^u}{\kappa_l^u} \right)^2$ is *an upper bound of the revenue rate and demand rate deviations* of price $\hat{p}_l$. When this is less than τ_1, we can simply apply $\hat{p}_l$ and the loss will not exceed the loss of the first learning period.

$$N^u = \min_l \left\{ l : \left(\frac{\overline{p}_l^u - \underline{p}_l^u}{\kappa_l^u} \right)^2 \sqrt{\log n} < \tau_1^u \right\}$$

Similarly, we state the set of relations we want $\left(\tau_i^c, \kappa_i^c\right)_{i=1}^{N^c}$ to satisfy

$$\frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c} \sim \sqrt{\frac{\kappa_i^c}{n\tau_i^c}}, \quad \forall i = 2, \ldots, N^c, \tag{2.18}$$

$$\overline{p}_{i+1}^c - \underline{p}_{i+1}^c \sim \log n \cdot \frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c}, \quad \forall i = 1, \ldots, N^c - 1, \tag{2.19}$$

$$\tau_{i+1}^c \cdot \frac{\overline{p}_i^c - \underline{p}_i^c}{\kappa_i^c} \cdot \sqrt{\log n} \sim \tau_1^c, \quad \forall i = 1, \ldots, N^c - 1. \tag{2.20}$$

Also, we define

$$N^c = \min_l \left\{ l : \frac{\overline{p}_l^c - \underline{p}_l^c}{\kappa_l^c} \sqrt{\log n} < \tau_1^c \right\}. \tag{2.21}$$

About τ, κ, N VII

Now, we solve the relations (2.14)-(2.16) and obtain a set of parameters that satisfy them:

$$\tau_1^u = n^{-\frac{1}{2}} \cdot (\log n)^{3.5} \text{ and } \tau_i^u = n^{-\frac{1}{2}\cdot\left(\frac{3}{5}\right)^{i-1}} \cdot (\log n)^5, \forall i = 2, \ldots, N^u, \tag{2.22}$$

$$\kappa_i^u = n^{\frac{1}{10}\cdot\left(\frac{3}{5}\right)^{i-1}} \cdot \log n, \quad \forall i = 1, 2, \ldots, N^u \tag{2.23}$$

As a by-product, we have

$$\overline{p}_i^u - \underline{p}_i^u = n^{-\frac{1}{4}\left(1-\left(\frac{3}{5}\right)^{i-1}\right)}, \quad \forall i = 1, 2, \ldots, N^u. \tag{2.24}$$

Similarly, we solve (2.18)-(2.21) to obtain

$$\tau_1^c = n^{-\frac{1}{2}} \cdot (\log n)^{2.5} \text{ and } \tau_i^c = n^{-\frac{1}{2}\cdot\left(\frac{2}{3}\right)^{i-1}} \cdot (\log n)^3, \forall i = 2, \ldots, N^c \tag{2.25}$$

$$\kappa_i^c = n^{\frac{1}{6}\cdot\left(\frac{2}{3}\right)^{i-1}} \cdot \log n, \quad \forall i = 1, 2, \ldots, N^c \tag{2.26}$$

and

$$\overline{p}_i^c - \underline{p}_i^c = n^{-\frac{1}{2}\left(1-\left(\frac{2}{3}\right)^{i-1}\right)}, \quad \forall i = 1, \ldots, N^c. \tag{2.27}$$

Note that by (2.24) and (2.27), the price intervals defined in our algorithm indeed shrink in each iteration.

Contents

Settings I

- The multi-armed bandit (MAB) problem is often used to model the exploration-exploitation trade-off in the dynamic learning and pricing model without inventory constraints.
- We extend Thompson sampling to address the challenge of balancing the exploration-exploitation trade-off under the presence of inventory constraints.
- The first algorithm adapts Thompson sampling by *adding a linear programming (LP) subroutine* to incorporate inventory constraints.
- The second algorithm builds upon our first, which in each period, modifies the LP subroutine to further *account for the purchases made to date*.
- Both of the algorithms contain two simple steps in each iteration: *sampling* from a posterior distribution and *solving* a linear program.

Settings II

- Consider a retailer who sells N products, indexed by $i \in [N] = \{1, 2, ..., N\}$, over a finite selling season. The selling season is divided into T periods.
- These products consume M resources, indexed by $j \in [M]$.
- There are I_j units of initial inventory for each resource $j \in [M]$, and there is no replenishment during the selling season. We define $I_j(t)$ as the inventory at the end of period t, and we denote $I_j(0) = I_j$.
- We assume that one unit of product i consumes a_{ij} units of resource j, where a_{ij} is a fixed constant.

In each period $t \in [T]$, the following sequence of events occurs:

1. The retailer offers a price for each product from a finite set of admissible price vectors.

Settings III

- We denote this set by $\{p_1, p_2, \ldots, p_K\}$where $p_k(\forall k \in [K])$ is a vector of length N specifying the price of each product.
- More specifically, we have $p_k = (p_{1k}, \ldots, p_{Nk})$, where p_{ik} is the price of product i, for all $i \in [N]$.
- Following the tradition in dynamic pricing literature, we also assume that there is a "shut-off" price p_∞ such that the demand for any product under this price is zero with probability one.
- We denote by $P(t) = (P_1(t), \ldots, P_N(t))$ the prices chosen by the retailer in this period, and require that $P(t) \in \{p_1, p_2, \ldots, p_K, p_\infty\}$.

2. Customers then observe the prices chosen by the retailer and make purchase decisions.
   - We denote by $D(t) = (D_1(t), \ldots, D_N(t))$ the demand of each product at period t.

Settings IV

- We assume that given $P(t) = p_k$, the demand $D(t)$ is sampled from a probability distribution on $\mathbb{R}_+^N$ with joint cumulative distribution function (CDF) $F(x_1, \ldots, x_N; p_k, \theta)$, indexed by a parameter (or a vector of parameters) θ that takes values in the parameter space $\Theta \subset \mathbb{R}^l$.
- The distribution is assumed to be *subexponential*; note that many commonly used demand distributions such as normal, Poisson, exponential and all bounded distributions belong to the family of subexponential distributions.
- We also assume that $D(t)$ is *independent* of the history $\mathcal{H}_{t-1} = (P(1), D(1), \ldots, P(t-1), D(t-1))$ given $P(t)$

Settings V

Depending on whether there is sufficient inventory, one of the following events happens:

(a) If there is enough inventory to satisfy all demand, the retailer receives an amount of revenue equal to $\sum_{i=1}^{N} D_i(t)P_i(t)$, and the inventory level of each resource $j \in [M]$ diminishes by the amount of each resource used such that $I_j(t) = I_j(t-1) - \sum_{i=1}^{N} D_i(t)a_{ij}$.

(b) If there is not enough inventory to satisfy all demand, the demand is partially satisfied and the rest of demand is lost. Let $\tilde{D}_i(t)$ be the demand satisfied for product i. We require $\tilde{D}_i(t)$ to satisfy three conditions:

- $0 \leq \tilde{D}_i(t) \leq D_i(t), \forall i \in [N]$;
- the inventory level for each resource at the end of this period is nonnegative: $I_j(t) = I_j(t-1) - \sum_{i=1}^{N} \tilde{D}_i(t)a_{ij} \geq 0, \forall j \in [M]$;

Settings VI

- there exists at least one resource $j' \in [M]$ whose inventory level is zero at the end of this period, i.e. $I_{j'}(t) = 0$.

Besides these natural conditions, we do not require any additional assumption on how demand is specifically fulfilled. The retailer then receives an amount of revenue equal to $\sum_{i=1}^{N} \tilde{D}_i(t) P_i(t)$ in this period.

- Assume that the demand parameter θ is fixed but *unknown* to the retailer at the beginning of the season, and the retailer must learn the true value of θ from demand data.
- In each period, the price vector $P(t)$ can only be chosen based on the observed history $\mathcal{H}_{t-1}$, but cannot depend on the unknown value θ or any event in the future.
- We use a parametric Bayesian approach in our model, where the retailer has a *known* prior distribution of $\theta \in \Theta$ at the beginning of the selling season.

Thompson Sampling with Fixed Inventory Constraints I

- For each resource $j \in [M]$, we define a *fixed* constant $c_j := I_j/T$.
- Given any demand parameter $\rho \in \Theta$, we define the mean demand under ρ as the expectation associated with CDF $F(x_1, ..., x_N; p_k, \rho)$ for each product $i \in [N]$ and price vector $k \in [K]$.
- We denote by $d = \{d_{ik}\}_{i\in[N],k\in[K]}$ the mean demand under the *true* model parameter θ.
- "Fixed" refers to the fact that we use fixed constants c_j for all time periods as opposed to updating c_j over the selling season as inventory is depleted.

Thompson Sampling with Fixed Inventory Constraints II

Repeat the following steps for all periods $t = 1, ..., T$:

1. *Sample Demand*: Sample a random parameter $\theta(t) \in \Theta$ according to the posterior distribution of θ given history $\mathcal{H}_{t-1}$. Let the mean demand under $\theta(t)$ be $d(t) = \{d_{ik}(t)\}_{i\in[N],k\in[K]}$.
2. *Optimize Prices given Sampled Demand*: Solve the following linear program, denoted by $\mathbf{LP}(d(t))$:

$$\begin{aligned}\mathbf{LP}(d(t)): \quad \max_{x} & \sum_{k=1}^{K}\left(\sum_{i=1}^{N} p_{ik} d_{ik}(t)\right) x_k \\ \text{subject to} & \sum_{k=1}^{K}\left(\sum_{i=1}^{N} a_{ij} d_{ik}(t)\right) x_k \le c_j, \forall j \in [M] \\ & \sum_{k=1}^{K} x_k \le 1 \text{ and } x_k \ge 0, k \in [K].\end{aligned}$$

Thompson Sampling with Fixed Inventory Constraints III

Let $x(t) = (x_1(t), \ldots, x_K(t))$ be the optimal solution to $\mathbf{LP}(d(t))$.

3. *Offer Price*: Offer price vector $P(t) = p_k$ with probability $x_k(t)$, and choose $P(t) = p_\infty$ with probability $1 - \sum_{k=1}^{K} x_k(t)$.
4. *Update Estimate of Parameter*: Observe demand $D(t)$. Update the history $\mathcal{H}_t = \mathcal{H}_{t-1} \cup \{P(t), D(t)\}$ and the posterior distribution of θ given $\mathcal{H}_t$.

Remarks

- The key idea of the Thompson sampling algorithm is to use random sampling from the posterior distribution to balance the exploration-exploitation trade-off. (Step 1)
- In Step 2, the retailer solves a linear program, $\mathbf{LP}(d(t))$, which identifies the optimal mixed price strategy that maximizes expected revenue given the sampled parameters.
- In Step 3, the retailer randomly offers one of the K price vectors (or p_∞) according to probabilities specified by the optimal solution of $\mathbf{LP}(d(t))$.

Bernoulli Demand with Independent Uniform Prior I

- Assume that for all prices, the demand for each product is Bernoulli distributed. Then the unknown parameter θ is just the mean demand of each product.
- We use a beta posterior distribution for each θ.
- We assume that the prior distribution of mean demand d_{ik} is Beta(1,1) and is *independent* for all $i \in [N]$ and $k \in [K]$.
- Let $N_k(t-1)$ be the number of time periods that the retailer has offered price p_k in the first $t-1$ periods, and let $W_{ik}(t-1)$ be the number of periods that product i is purchased under price p_k during these periods.
- In Step 1, the posterior distribution of d_{ik} is

$$\text{Beta}\left(w_{ik}(t-1)+1, N_k(t-1)-W_{ik}(t-1)+1\right)$$

Bernoulli Demand with Independent Uniform Prior II

- In Step 2 and 3, $\mathbf{LP}(d(t))$ is solved and a price vector $p_{k'}$ is chosen; then, the customer demand $D_i(t)$ is revealed to the retailer.
- In Step 4, we then update $N_{k'}(t) \leftarrow N_{k'}(t-1)+1$, $W_{ik'}(t) \leftarrow W_{ik'}(t-1)+D_i(t)$ for all $i \in [N]$. The posterior distributions associated with the $K-1$ unchosen price vectors ($k \neq k'$) are not changed.

Thompson Sampling with Inventory Constraint Updating I

- We can update c_j over the selling season as inventory is depleted, thereby incorporating real-time inventory information into the algorithm.
- Define $c_j(t) = I_j(t-1)/(T-t+1)$ as the average inventory for resource j available from period t to T.
- We then replace constants c_j with $c_j(t)$ in $\mathbf{LP}(d(t))$ in step 2 of TS-fixed, which gives us the Thompson sampling with Inventory Constraint Updating algorithm (*TS-update* for short).

Thompson Sampling with Inventory Constraint Updating II

$$\begin{aligned}
\mathbf{LP}(d(t), c(t)): \quad & \max_{x} \sum_{k=1}^{K} \left(\sum_{i=1}^{N} p_{ik} d_{ik}(t) \right) x_k \\
& \text{subject to} \sum_{k=1}^{K} \left(\sum_{i=1}^{N} a_{ij} d_{ik}(t) \right) x_k \leq c_j(t), \forall j \in [M] \\
& \sum_{k=1}^{K} x_k \leq 1 \\
& x_k \geq 0, k \in [K].
\end{aligned}$$

- In fact there are situations where TS-update outperforms TS-fixed and vice versa.

Performance Analysis I

- The retail's regret over the selling horizon

$$\text{Regret}(T, \theta) = \text{E}\left[\text{Rev}^*(T) \mid \theta\right] - \text{E}[\text{Rev}(T) \mid \theta],$$

where $\text{Rev}^*(T)$ is is the revenue achieved by the optimal policy if the demand parameter θ is known a priori, and $\text{Rev}(T)$ is the revenue achieved by an algorithm that may not know θ.

- Bayesian regret (risk)

$$\text{BayesRegret(T)} = E[\text{Regret}(T, \theta)]$$

where the expectation is taken over the prior distribution of θ.

Performance Analysis II

- We assume that for each product $i \in [N]$, the demand $D_i(t)$ is bounded by $D_i(t) \in [0, \overline{d}_i]$ under any price vector $p_k, \forall k \in [K]$.
- We define the constants

$$p_{\max} := \max_{k \in [K]} \sum_{i=1}^{N} p_{ik} \overline{d}_i, \quad p^j_{\max} := \max_{i \in [N]: a_{ij} \neq 0, k \in [K]} \frac{p_{ik}}{a_{ij}}, \forall j \in [M]$$

where $p_{\max}$ is the maximum revenue that can possibly be achieved in one period, and $p^j_{\max}$ is the maximum revenue that can possibly be achieved by adding one unit of resource $j, \forall j \in [M]$.

Performance Analysis III

Theorem 3.1

The Bayesian regret of TS-fixed is bounded by

$$\text{BayesRegret}(T) \leq \left(18p_{\max} + 37\sum_{i=1}^{N}\sum_{j=1}^{M} p_{\max}^{j} a_{ij}\overline{d}_i\right)\sqrt{TK\log K}.$$

Theorem 3.2

The Bayesian regret of TS-update is bounded by

$$\text{BayesRegret}(T) \leq \left(18p_{\max} + 40\sum_{i=1}^{N}\sum_{j=1}^{M} p_{\max}^{j} a_{ij}\overline{d}_i\right)\sqrt{TK\log K} + p_{\max}M$$

Performance Analysis IV

- The regret bound of TS-update is slightly worse than the regret bound of TS-fixed.
- The results above state that the Bayesian regrets of both TS-fixed and TS-update are bounded by $O(\sqrt{TK \log K})$.
- The regret bounds are *prior-free* as they do not depend on the prior distribution of parameter θ;
- For a multi-armed bandit problem with reward in $[0, 1]$ – a special case of our model with no inventory constraints – no algorithm can achieve a prior-free Bayesian regret smaller than $\Omega\left(\sqrt{KT}\right)$.
- The above regret bounds are optimal with respect to T and cannot be improved by any other algorithm by more than $\sqrt{\log K}$.

Contents

Settings I

- A seller sells to incoming customers n types of products, , each of which is made up from a subset of m types of resources, during a finite selling season which consists of T decision periods.
- $A = [A_{ij}] \in \mathbb{R}_+^{n \times n}$ is the *resource consumption matrix*, which indicates that a single unit of product j requires A_{ij} units of resource i.
- Denote by $C \in \mathbb{R}_+^m$ the vector of *initial capacity levels* of all resources at the beginning of the selling season which cannot be replenished and have zero salvage value at the end of the selling season.
- At the beginning of period $t \in [T]$, the seller first decides the price $p_t = (p_{t,1}; \ldots; p_{t,n})$ for his products, where p_t is chosen from a convex and compact set $\mathcal{P} = \otimes_{l=1}^n \left[\underline{p}_l, \overline{p}_l\right] \subseteq \mathbb{R}^n$ of feasible price vectors.

Settings II

- Let $D_t(p_t) = (D_{t,1}(p_t); \ldots; D_{t,n}(p_t)) \in \mathcal{D} := \{(d_1; \ldots; d_n) \in \{0,1\}^n : \sum_{i=1}^n d_i \leq 1\}$ denote the vector of *realized demand* in period t under price p_t. For simplicity, we assume that at most one sale for one of the products occurs in each period. We assume that the purchase probability vector for all products under any price p_t, i.e., $\lambda^*(p_t) := \mathbf{E}[D_t(p_t)]$ is *unknown* to the seller, and this relationship $\lambda^*(.)$, also known as the demand function, needs to be estimated from the data the seller observes during the finite selling season.
- Define the *revenue function* $r^*(p) := p \cdot \lambda^*(p)$ to be the one-period expected revenue that the seller can earn under price p. It is typically assumed in the literature that $\lambda^*(.)$ is invertible. By abuse of notation, we can then write $r^*(p) = p \cdot \lambda^*(p) = \lambda \cdot p^*(\lambda) = r^*(\lambda)$ to emphasize the dependency of revenue on demand rate instead of on price.

Settings III

Assumption 4.1 (Regularity Assumptions)

R1. *$\lambda^*(.)$ is twice continuously differentiable and it has an inverse function $p^*(.)$ which is also twice continuously differentiable.*

R2. *There exists a set of turnoff prices $p_j^\infty \in \mathbb{R}_+ \cap \{\infty\}$ for $j = 1, \ldots, n$ such that for any $p = (p_1; \ldots; p_n)$, $p_j = p_j^\infty$ implies that $\lambda_j^*(p) = 0$.*

R3. *$r^*(.)$ is bounded and strongly concave in λ.*

Remarks:

- The nice solution structure for single product setting breaks down in the presence of multiple types of products and resources.

Settings IV

- The approach of estimating the deterministic optimal prices and then applying this learned price *may not be sufficient to get tight regret bound*. Because ensuring the same estimation error of the deterministic optimal prices in multidimensional setting requires significantly more observations which in turn affects the best achievable regret bound of this approach.

Parametric Case I

- Let Θ be a compact subset of $\mathbb{R}^q$, where $q \in \mathbb{Z}_{++}$is the number of unknown parameters. The seller knows that the underlying demand function $\lambda^*(.)$ equals $\lambda(.;\theta)$ for some $\theta \in \Theta$.
- The one-period expected revenue function is given by $r(p;\theta) := p \cdot \lambda(p;\theta)$. To leverage the parametric structure of the unknown function, we will focus primarily on *Maximum Likelihood (ML) estimation*. To guarantee the regular behavior of ML estimator, certain statistical conditions need to be satisfied.

Parametric Case II

- To formalize these conditions in our context, it is convenient to first consider the distribution of a sequence of demand realizations when a sequence of $\tilde{q} \in \mathbb{Z}_{++}$ *fixed* price vectors $\tilde{p} = \left(\tilde{p}^{(1)}, \tilde{p}^{(2)}, \ldots, \tilde{p}^{(\tilde{q})}\right) \in \mathcal{P}^{\tilde{q}}$ have been applied. For all $d_{1:\tilde{q}} \in \mathcal{D}^{\tilde{q}}$, we define

$$\mathbf{P}^{\tilde{p},\theta}\left(d_{1:\tilde{q}}\right) := \prod_{s=1}^{\tilde{q}} \left[\left(1 - \sum_{j=1}^{n} \lambda_j\left(\tilde{p}^{(s)};\theta\right)\right)^{\left(1-\sum_{j=1}^{n} d_{s,j}\right)} \prod_{j=1}^{n} \lambda_j\left(\tilde{p}^{(s)};\theta\right)^{d_{s,j}} \right]$$

and denote by $\mathbf{E}_{\theta}^{\tilde{p}}$ the expectation with respect to $\mathbf{P}^{\tilde{p},\theta}$.

Assumption 4.2 (Parametric Family Assumptions)

A1 $\lambda(p;\theta)$ *and* $\frac{\partial \lambda_j}{\partial p_i}(p;\theta)$ *for all* $i, j \in [n]$ *and* $i \neq j$ *are continuously differentiable in* θ.

A2 *R1 and R3 hold for all* $\theta \in \Theta$.

A3 *There exists* $\tilde{p} = \left(\tilde{p}^{(1)}, \tilde{p}^{(2)}, \ldots, \tilde{p}^{(\tilde{q})}\right) \in \mathcal{P}^{\tilde{q}}$ *such as for all* $\theta \in \Theta$,

i $\mathbf{P}^{\tilde{p},\theta}(.) \neq \mathbf{P}^{\tilde{p},\theta'}(.)$ *for all* $\theta' \in \Theta$ *and* $\theta' \neq \theta$.

ii *For all* $k \in [\tilde{q}]$ *and* $j \in [n]$, $\lambda_j\left(\tilde{p}^{(k)};\theta\right) > 0$ *and* $\sum_{j=1}^{n} \lambda_j\left(\tilde{p}^{(k)};\theta\right) < 1$.

iii *The minimum eigenvalue of the matrix* $\mathcal{I}(\tilde{p},\theta) := \left[\mathcal{I}_{i,j}(\tilde{p},\theta)\right] \in \mathbb{R}^{q\times q}$ *where*

$$\mathcal{I}_{i,j}(\tilde{p},\theta) = \mathbf{E}_{\theta}^{\tilde{p}}\left[-\frac{\partial^2}{\partial\theta_i \partial\theta_j}\log \mathbf{P}^{\tilde{p},\theta}\left(D_{1:\tilde{q}}\right)\right]$$

is bounded from below by a positive number.

Parametric Case IV

- We call $\tilde{p}$ in Assumption 4.2 A3 *exploration prices*. A3 ensures that there exists a set of price vectors (e.g., $\tilde{p}$), which, when used repeatedly, would allow the seller to use *ML estimator to statistically identify the true demand parameter*.
- A3-iii requires the *Fisher matrix* to be strongly positive definite; this is needed to guarantee that the seller's information about the underlying parameter vector strictly increases as he observes more demand realizations under $\tilde{p}$.

Parametric Self-adjusting (PSC) I

- Tuning Parameter: L
- Define $\hat{\Delta}_t\left(p_t;\hat{\theta}_L\right) = D_t - \lambda(p_t;\hat{\theta}_L)$ and let C_t denote the remaining capacity at the *end* of period t.

Stage 1 (Exploration)

a. Determine the exploration prices $\left\{\tilde{p}^{(1)},\tilde{p}^{(2)},\ldots,\tilde{p}^{(\tilde{q})}\right\}$.

b. For $t = 1$ to L, do:
   - If $C_{t-1} \succeq A_j$ for all j, apply price $p_t = \tilde{p}^{(\lfloor (t-1)\tilde{q}/L \rfloor + 1)}$ in period t.
   - Otherwise, apply price $p_{t',j} = p_j^{\infty}$ for all j and $t' \geq t$; then terminate PSC.

c. At the end of period L :
   - Compute the ML estimate $\hat{\theta}_L$ based on $p_{1:L}$ and $D_{1:L}$
   - Solve $\mathrm{P}_\lambda\left(\hat{\theta}_L\right)$ for $\lambda^D\left(\hat{\theta}_L\right)$.

Parametric Self-adjusting (PSC) II

Stage 2 (Exploitation)

For $t = L + 1$ to T, compute:

$$\hat{p}_t = p\left(\lambda^D\left(\hat{\theta}_L\right) - \sum_{s=L+1}^{t-1} \frac{\hat{\Delta}_s\left(p_s; \hat{\theta}_L\right)}{T - s}; \hat{\theta}_L\right). \tag{4.1}$$

- If $C_{t-1} \succeq A_j$, and $\hat{p}_t \in \mathcal{P}$, apply price $p_t = \hat{p}_t$ in period t
- Otherwise, for product $j = 1$ to n, do:
  - If $C_{t-1} \prec A_j$, apply price $p_{t,j} = p_j^\infty$.
  - Otherwise, apply price $p_{t,j} = p_{t-1,j}$

Remarks:

- PSC uses the adaptive price adjustment rule in (4.1) for exploitation.

Parametric Self-adjusting (PSC) III

- Suppose the estimate of the parameter vector is accurate. $\hat{\Delta} = \Delta := D_t - \lambda(p_t; \theta^*)$, and the pricing rule in (4.1) reduces to adjusting the prices in each period t to achieve a *target demand rate*, i.e., $\lambda^D(\theta^*) - \sum_{s=L+1}^{t-1} \frac{\Delta_s}{T-s}$.
- $\lambda^D(\theta^*)$ is the optimal demand rate if there were no stochastic variability, and we use it as a *base rate*;
- The second part of the expression, on the other hand, works as a fine adjustment to the base rate in order to mitigate the observed stochastic variability.

Theorem 4.1

Suppose that $R1 - R4$ *and* $A1 - A3$ *hold. Set* $L = \lceil \sqrt{kT} \rceil$. *Then, there exists a constant* $M_1 > 0$ *independent of* $k \geq 1$ *such that* $\rho^{PSC}(k) \leq M_1 \sqrt{k}$ *for all* $k \geq 1$.