

Data-driven pricing

by

Thibault Le Guen

Ingénieur des Arts et Manufactures, Ecole Centrale Paris (2007)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Master of Science in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2008

© Massachusetts Institute of Technology 2008. All rights reserved.

Author
Sloan School of Management
August 7, 2008

Certified by
Professor Georgia Perakis
Associate Professor
Thesis Supervisor

Accepted by
Professor Cynthia Barnhart
Codirector, Operations Research Center

Data-driven pricing

by

Thibault Le Guen

Submitted to the Sloan School of Management
on August 7, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Operations Research

Abstract

In this thesis, we develop a pricing strategy that enables a firm to learn the behavior of its customers as well as optimize its profit in a monopolistic setting. The single product case as well as the multi product case are considered under different parametric forms of demand, whose parameters are unknown to the manager.

For the linear demand case in the single product setting, our main contribution is an algorithm that guarantees almost sure convergence of the estimated demand parameters to the true parameters. Moreover, the pricing strategy is also asymptotically optimal. Simulations are run to study the sensitivity to different parameters.

Using our results on the single product case, we extend the approach to the multi product case with linear demand. The pricing strategy we introduce is easy to implement and guarantees not only learning of the demand parameters but also maximization of the profit. Finally, other parametric forms of the demand are considered. A heuristic that can be used for many parametric forms of the demand is introduced, and is shown to have good performance in practice.

Thesis Supervisor: Professor Georgia Perakis
Title: Associate Professor

Acknowledgments

First and foremost, I would like to express my gratitude to my advisor Georgia Perakis. During these two years, Georgia has been an extremely dedicated advisor, always pushing me to give my best and giving honest feedbacks. Moreover, on a personal level, she has been very supportive and cared of my well-being. Georgia, I am really indebted.

I would like to extend my acknowledgment to Philipp Keller, who got involved in a similar research project. Thanks Phil, for coming up with so many ideas, for being fun and disagreeing with me on every topic possible, from the choice of a movie to the running time of the algorithm! Thanks also for reading this thesis and making very insightful comments.

Moreover, I would like to thank my classmates at the ORC for all the support in the bad times (exams, tough homeworks...) and all the fun in the good times (retreat, socials...). Special thanks to Alex, Dan, Dave, Jonathan, Mallory and Martin. Thanks to Laura, Paulette and Andrews for running the Center so smoothly and being always there to answer my administrative questions.

I am extremely grateful to the Rotary foundation for their “Ambassadorial Scholar Award”, which enabled me not only to come and study in the US, but also to have a wonderful Host Counselor, Dr William Wheeler, who made me discover and appreciate New England.

Finally, I would like to thank my parents for dedicating their lives to their children. Thanks also to my three sisters Anne-Sophie, Aude and Beatrice!

Contents

- 1 Introduction 15**
- 1.1 Motivation 15
- 1.2 Literature review 17
 - 1.2.1 Revenue management and pricing 17
 - 1.2.2 Data-driven approach and learning 18
 - 1.2.3 Least-squares estimates 20
- 1.3 Problem and contributions 20
 - 1.3.1 Practical application 20
 - 1.3.2 Goals of the thesis and contributions 21

- 2 Single product case 23**
- 2.1 Introduction 23
 - 2.1.1 Motivation 23
 - 2.1.2 Notations 24
 - 2.1.3 Formulation of the problem 24
 - 2.1.4 Idea behind the algorithm 25
 - 2.1.5 Structure 26
- 2.2 Least-squares estimator and convergence conditions 27
 - 2.2.1 Least squares 27
 - 2.2.2 Consistency conditions 29
- 2.3 Pricing algorithm 34

2.4	Single product with capacity	42
2.4.1	Model	42
2.4.2	Perturbation pricing	43
2.5	Conclusions	45
3	Simulations: Affine demand, single product case	47
3.1	Different Algorithms	47
3.1.1	Standard algorithm	47
3.1.2	Improved algorithm with transient phase.	51
3.1.3	Capacitated case algorithm	53
3.2	Sensitivity analysis	55
3.2.1	Input: Initial prices	55
3.2.2	Demand parameters: Slope and intercept	57
3.2.3	Noise: Standard deviation and distribution	60
3.2.4	Capacity	61
3.3	Insights	62
4	Multi-product case	63
4.1	Demand model	63
4.1.1	Model	63
4.1.2	Existence and uniqueness of the optimum	65
4.2	Uniform demand	66
4.2.1	Additional assumptions	66
4.2.2	Optimization	67
4.2.3	Convergence algorithm	69
4.3	General symmetric demand case	73
4.3.1	Subroutine focusing on one product	74
4.3.2	Tatonnement algorithm	82
4.3.3	Algorithm in practice	86

4.4	Conclusions	89
5	Simulations: Affine demand, multi-product case.	91
5.1	General algorithm	91
5.2	Sensitivity analysis: Learning phase.	92
5.3	Sensitivity analysis: Tatonnement algorithm.	95
5.3.1	Standard deviation of the noise	95
5.3.2	Number of iterations of the subroutine	97
5.3.3	Degree of diagonal dominance	98
5.3.4	Number of products	101
5.4	Insights	103
6	Non linear parametric models	105
6.1	Demand models	105
6.1.1	Loglinear demand	105
6.1.2	Constant elasticity demand	107
6.2	Single-product case	108
6.2.1	Loglinear demand	108
6.2.2	Constant elasticity demand	113
6.3	Multi-product case	116
6.4	Conclusions	120
7	Simulations: Nonlinear case and heuristic	121
7.1	Single product case	121
7.1.1	Loglinear demand	121
7.1.2	Loglinear demand with capacity	125
7.1.3	Constant elasticity demand	126
7.2	Performance of the heuristic	128
7.2.1	Loglinear case	128
7.2.2	Linear demand case	131

7.3 Insights	138
8 Conclusions	141

List of Figures

3-1	Single product, standard algorithm	48
3-2	Estimates of the slope	49
3-3	Estimates of the intercept	50
3-4	Single product, improved algorithm	52
3-5	Single product with capacity	54
3-6	Sensitivity to the starting prices	56
3-7	Sensitivity to the intercept	58
3-8	Sensitivity to the slope	59
5-1	Output for the multi product case	93
5-2	Sensitivity to the noise standard deviation	96
5-3	Sensitivity to the degree of diagonal dominance	99
5-4	Sensitivity to the number of products	102
6-1	Illustration of the perturbation	117
7-1	Output for the loglinear demand case	122
7-2	Loglinear demand: sensitivity to the slope	124
7-3	Revenue for the loglinear demand case	129
7-4	Heuristic: loglinear demand, 2 products	130
7-5	Conditions by Lai and Wei without a perturbation	132
7-6	Conditions by Lai and Wei with a perturbation	133
7-7	Comparison Heuristic-Tatonnement algorithm	137

7-8 Sensitivity of the heuristic to the number of products 139

List of Tables

2.1	Notations for the single product case	24
3.1	Standard algorithm	50
3.2	Improved algorithm	52
3.3	Single product case with capacity	55
3.4	Sensitivity to the intercept	57
3.5	Sensitivity to the standard deviation of the noise	60
3.6	Sensitivity to the distribution of the noise	61
3.7	Sensitivity to the capacity	61
5.1	Output of the algorithm	94
5.2	Learning phase	95
5.3	Sensitivity to the standard deviation of the noise	97
5.4	Sensitivity to the number of iterations of the subroutine	98
5.5	Output for $r=0.2$	100
5.6	Output for $r=0.6$	100
5.7	Output for $r=0.98$	100
5.8	Sensitivity to the number of products	101
5.9	Sensitivity to the number of products (continued)	102
7.1	Loglinear demand: sensitivity to the standard deviation of the noise	123
7.2	Loglinear demand: sensitivity to the slope	125
7.3	Loglinear demand with capacity	125

7.4	Constant-elasticity demand: sensitivity to the standard deviation of the noise	126
7.5	Constant elasticity demand: sensitivity to the demand parameters	127
7.6	Sensitivity to the value of the perturbation	131
7.7	Accuracy of the tatonnement algorithm	135
7.8	Accuracy of the heuristic	135
7.9	Accuracy of the heuristic with \mathbf{A} known	136
7.10	Sensitivity of the heuristic to the number of products	138

Chapter 1

Introduction

1.1 Motivation

Originally known as yield management, the field of revenue management started in the airline industry in the wake of deregulation. Revenue management is about maximizing profit for a fixed, perishable resource like the seats on an airplane. Robert Crandal, the CEO of American Airlines described revenue management as

“The single most important technical development in the airline industry since we entered deregulation. ”

Although the initial focus of revenue management was opening and closing fares, it later also led to price changes. Hence the border between pricing and revenue management blurred over the last decades.

Pricing has always been considered as a critical lever for revenue management. It is also considered as the most important of the Ps of marketing-the others being product place and promotion- because it is the only one that generates revenue for a company. The quote below by Mc Kinsey [27] shows that it is a high impact factor on the profits. It concludes that:

“Pricing right is the most effective way for managers to increase profits. Consider the average income statement of an S&P 500 company: a price increase of

1%, if volumes remain stable, would generate an 8% increase in operating profits—an impact nearly 50% greater than that of a 1% fall in variable costs such as materials and direct labor and more than three times greater than the impact of a 1% increase in volume. Unfortunately, the sword of pricing cuts both ways. A decrease of 1% in average prices has the opposite effect, bringing down operating profits by that same 8% if others factors remain steady. ”

The power of pricing

The Mc Kinsey quarterly

Number 1, 2003

However, determining the appropriate price for a product requires a wealth of data. Until recently, data was not easily available and it was difficult for companies to use it to reflect the market. Hence pricing was often static. The recent development of the Internet and information technology, had an impact on the ability of a company to do dynamic pricing. It created tremendous opportunities to use this available data. Thanks to all these new technologies, it became increasingly easy for industry to develop dynamic pricing policies.

Moreover, the wealth of data available can also be used by companies in order to understand what their demand functions are in term of prices. In the literature a specific price-demand relationship with known parameters is often assumed, in practice, nevertheless, this is an important and non trivial step. For example, one can assume a parametric form of the demand but still one needs to know the exact values of the parameters, which is often too much of an assumption to impose. That is why it is better not to separate the demand learning from the profit optimization in order to optimize on the right demand model and to take advantage of the learning incentive. Hence, companies need simultaneously, using the available data, to understand what the demand functions are and to determine how to price. This is particularly decisive in order to set an adequate pricing strategy at the launching of a new product. Mc Kinsey concluded in [28] that “companies habitually charge less than they could for new offerings” and that “price-benefit analysis should start early in

the development circle.”

As a consequence, a lot of firms decide to outsource their entire pricing strategy to external companies and consultants such as Demandtec, SAP, Knowledge support Systems (KSS), Oracle, as well as PROS Revenue Management.

1.2 Literature review

1.2.1 Revenue management and pricing

There exists an extensive literature on the topic of revenue management which became increasingly popular in the last decades. Below, we only list some recent literature review papers.

In their recent paper, Chiang, Chen and Xu [13] provide a comprehensive review of the use of revenue management in different industries and discuss research on different revenue management strategies including pricing, control, capacity control. Talluri and Van Ryzin [37] describe in a book the theory and practice of Revenue Management from the birth of this field. Review papers include also Mc Gill and van Ryzin [26], Bitran and Caldentey [7]. Pak and Piersma [31] focus on revenue management in the airline industry.

In what follows, I will focus on some review papers.

In their paper, Elmaghraby and Keskinocak [17] split the pricing literature into three categories: first the inventory might or not replenish over the time horizon, second the demand can be dependent or independent over time and third the customers can be myopic or strategic.

Most of the papers are making assumptions about the form of the demand function. For example, Bitran, Caldentey and Monschein [8] assume a demand rate which has a known distribution and depends on time. The demand can also be modeled as deterministic [24] or stochastic with a known distribution. Smith and Achabal [36] use a demand that is a function of the price as well as the inventory level. Finally Elmaghraby et al. [16] uses a

stochastic demand with a distribution known a priori.

1.2.2 Data-driven approach and learning

However, in order to price intelligently, we need to understand how prices and demand relate. Assuming a particular price-demand relationship with fixed parameters is not realistic. This is particularly relevant in practice as managers need to manage uncertain demand over a finite time horizon through allocating a fixed capacity.

With the development of information technology over the past years, companies have accumulated a significant number of pricing data. Thus researchers developed data driven pricing methods which are able to estimate the demand functions. We can split these methods into two categories: parametric approaches and non parametric approaches.

The bulk of the literature on parametric approaches uses Bayesian techniques. A parametric form is assumed for the demand function with unknown parameters, which are estimated using data. Aviv and Pazgal ([1],[2]) introduce learning in a model in which customers arrive according to a Poisson process. In [1] they derive a closed form optimal control policy in a model with unknown arrival rate and known reservation price functions while in [2], they use a Markov Decision Process framework to describe a model with a finite number of arrival rate and different reservation prices and parameter scenarios. They show the importance of a trade off between a low price which induces a loss in revenue and a high price which slows the learning. Lin [25] uses a similar approach. Carvalho and Puterman ([11],[12]) address learning with two kinds of demand functions: a loglinear demand model with unknown coefficients in [11] and in [12] a binomial model of demand. Petruzzi and Dada [32] consider learning, pricing and inventory control. In their models, inventory censors demand and the demand function parameters are revealed once they are seen. Lobo and Boyd [10] discuss practical policies for the monopolistic pricing problem with uncertain demand. They introduce price variations to give a better estimate of the elasticity of demand and introduce an approximation of the dynamic programming solution. Rustichini and Wolinsky [34] study

the problem of a monopoly which is uncertain about the demand it faces with a demand that is changing over time in a Markov fashion. They characterize the monopoly's optimal policy and compare it with an informed monopoly's policy. Bertsimas and Perakis [4] present an optimization approach for jointly learning demand as a function of price, using dynamic programming. They consider both the competitive and non competitive case and show experimentally that dynamic programming based approaches outperform myopic policies significantly. Kachani, Perakis and Simon [20] present an approach to dynamic pricing with demand learning in an oligopolistic environment, using ideas from Mathematical Programming with Equilibrium Constraints. Their approach allows for capacitated settings.

The pricing under demand uncertainty can also be addressed in a nonparametric way. The concave adaptive value optimization (CAVE) approach (e.g Godfrey and Powell [19]) can be used to successfully approximate the objective cost function with a sequence of linear function. Cope [15] deals with the case of revenue maximization in an infinite horizon. Assuming that for each price level the demand is observed without noise, he introduces a nonparametric Bayesian approach to help the manager make better pricing decisions. Besbes and Zeevi [5] use a blind nonparametric approach and show asymptotic optimality of a joint pricing and learning method for the single product case. In a recent paper [6], they present an algorithm for the multiple product case, testing every price vector within a multi dimensional grid. Larson, Olson and Sharma in [23] analyze the stochastic inventory control model when the demand is not known. They use a nonparametric approach in which the firm's prior information is characterized by a Dirichlet process. Recently, Farias and Van Roy [18] studied the case of a vendor with limited inventory and who knows the distribution of the demand but is uncertain about the mean and proposed a simple heuristic, for which they derive worst-case bounds.

The learning techniques we use in this paper are encountered in several areas, especially artificial intelligence. Sutton and Barto [3] provide a review of the use of reinforcement

learning and give numerous applications.

1.2.3 Least-squares estimates

One of the components of the algorithm we introduce in this thesis is a computationally tractable least squares algorithm.

In his book, Rice [33] provides a useful iterative formula for the computation of least-squares estimates. Bjorck [9] gives a thorough survey of the latest methods in least-squares estimation and their numerical properties.

Lai and Robbins [21] provide conditions for strong consistency of the parameters in the case where the input regressors are independent under mild assumptions. When the input observations are not independent, the conditions become more messy. They are given in [22]. Lai and Wei in [22] discuss applications of these results to interval estimation of the regression parameters and to applications in linear dynamic systems.

1.3 Problem and contributions

1.3.1 Practical application

In this thesis we develop a data-driven approach to pricing in a monopolistic setting without assuming any inventory control. We consider both the single product and the multi product case. An application domain that motivated this work comes from online song and movie sales. In this setting, capacity or inventory does not play any role and the prices can be updated frequently. More generally, this approach can be applied to any sector where inventory plays a minor role, for example in the “make to order” setting.

Consider the sale of songs on the Itunes.com website. This website is the market leader so we will assume that its market power is so large that we are close to a monopolistic setting. The songs represent products that are substitutes for one another which means that if the

price for one song increases then the demand for similar songs is going to increase.

We propose an efficient pricing method in this setting in order to learn the behavior of the customer as well as optimize the profit in the infinite time horizon setting. Nevertheless, our approach is general and can apply to other industries.

1.3.2 Goals of the thesis and contributions

The principal goal of this thesis is to study joint pricing and learning in a monopolistic setting. We assume a parametric form for the demand function in the presence of noise, whose parameters are not known. We consider both linear and nonlinear demand functions.

We develop a pricing strategy, using only past price and demand data. Our goal is twofold:

- First, we wish to learn the demand parameters in order to improve our knowledge of the customer behavior. A key issue in this problem comes from the fact that when the price is set at a certain level, the realization of the demand we observe is noisy. Furthermore, we do not assume knowledge of the demand parameters, but only of the parametric form of the demand function.
- Moreover, at the same time that the seller is learning the demand, he also wants to determine the optimal pricing policy. If the seller knew the true underlying demand parameters, he could use them in order to define the optimal pricing strategy. However, these parameters are not known in advance but are learned over time from the data. This gives rise to an online optimization problem.

The interest of such an approach is that it is easy to implement in practice by a manager. The manager just needs to keep track of the past prices and demand observed, and as an output, he will know the next price to use.

The thesis is structured as follows.

- a) In the first two chapters of the thesis, we provide a pricing algorithm for the single product case when the demand is an affine function of the price. We illustrate how the demand parameter estimates converge almost surely to the true demand parameters. Moreover, we show also that the prices converge to the optimal pricing policy asymptotically.
- b) In the next two chapters of the thesis, we extend the approach to the multi product case and show the interest of this approach in theory and through simulations. We illustrate the importance of the degree of substitutability between the products.
- c) In the final two chapters of the thesis, we extend the approach to two nonlinear parametric forms of demand: the loglinear and the constant elasticity demand. We consider both the single product case and the multi product case models.

Chapter 2

Single product case

2.1 Introduction

2.1.1 Motivation

We consider a warehouse selling a single product in a monopolistic environment. The demand is a linear function of the price with an additive noise term. The demand parameters are fixed over time but are unknown. Furthermore, we do not assume that we know the distribution of the noise.

There is no capacity constraint and the time horizon is infinite. Later, we extend the approach to include a capacity constraint. Moreover, we assume that we can neglect the purchase cost of the goods, or, without loss of generality, that the per unit cost is fixed.

The purpose of the pricing strategy is twofold:

- First the seller wants to learn the demand parameters in order to improve his knowledge of the behavior of the customers.
- Moreover, the seller wants to determine the optimal pricing policy. In other words, the seller needs to use the demand parameters in order to define the optimum pricing

p_n	Price for the product at time period n
$d(p)$	Demand for the product priced at p
$d_n(p)$	Estimated demand at time period n for the product priced at p
ϵ	Random noise
$\mathbb{E}(p \cdot d(p))$	Expected profit for the product priced at p
p^*	Optimal price
p_{min}	Lower bound on the prices
p_{max}	Upper bound on the prices

Table 2.1: Notations for the single product case

strategy. Nevertheless, because of the presence of a noise in the demand as well as the fact that the demand parameters are not known in advance but learned over time, the problem becomes more complex.

2.1.2 Notations

We introduce the notations that will be used throughout this part in Table 2.1.

2.1.3 Formulation of the problem

We assume that a warehouse is selling a single product at a price p , where $p \in [p_{min}, p_{max}]$.

We assume that the corresponding demand is of the form

$$d(p) = a - b \cdot p + \epsilon,$$

where $a > 0$, $b > 0$ and ϵ is a random noise with unknown distribution and mean zero.

The firm knows the form of the demand function but does not have knowledge of the actual value of the parameters a and b or the distribution of the noise.

The objective is to find the price p^* that optimizes the expected revenue subject to bounds on the price. Notice that p^* , the optimal price, must be the solution of the following

optimization problem.

$$\max_{p \in [p_{min}, p_{max}]} \mathbb{E}(Revenue) = \max_{p \in [p_{min}, p_{max}]} p \cdot \mathbb{E}(d(p)).$$

In this setting, if the demand parameters a and b were known, the solution would be:

$$p^* = \max \left(\min \left(\frac{b}{2a}, p_{max} \right), p_{min} \right) \quad (2.1)$$

The difficulty in determining p^* comes from the fact that the parameters a and b are unknown and that observations of the demand given a price also include a noise.

The horizon is infinite. At time period n of the horizon, the seller sets a price p_n . The parameters a and b are unknown and are estimated at each time step by a_n and b_n using least squares estimates. Then, the estimated demand is $d_n(p) = a_n - b_n \cdot p$.

2.1.4 Idea behind the algorithm

The ideal objective is to devise an algorithm such that the estimates of the demand parameters as well as the price and the expected revenue computed at each iteration converge respectively to the true demand parameters, the optimal price, and the maximum revenue respectively almost surely.

However, we devise a strategy that is able to achieve this objective only partially. Indeed, we have a policy such that the estimates of the demand parameters as well as a subset of the prices and the expected revenues at each iteration converge almost surely respectively to the true demand parameters, the optimal prices, and the maximum revenue respectively almost surely. Moreover, the average price as well as the average revenue will converge to their optimal values almost surely.

Mathematically, we will have a pricing policy such that almost surely

$$\begin{aligned} \lim_{n \rightarrow \infty} a_n &= a \\ \lim_{n \rightarrow \infty} b_n &= b \\ \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n p_i \cdot d_i &= p^* d^* \\ \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n p_i &= p^* \end{aligned}$$

Moreover, there exists an infinite subset $\mathfrak{A}(\mathbb{N}) \in \mathbb{N}$ such that for all $n \in \mathfrak{A}(\mathbb{N})$ almost surely:

$$\begin{aligned} \lim_{n \rightarrow \infty, n \in \mathfrak{A}(\mathbb{N})} p_n &= p^* \\ \lim_{n \rightarrow \infty, n \in \mathfrak{A}(\mathbb{N})} \mathbb{E}(p_n \cdot d(p_n)) &= \mathbb{E}(p^* \cdot d(p^*)). \end{aligned}$$

The idea behind the algorithm we introduce and study in this thesis is the following. We price optimally given some parameter estimates computed through regressions. We add this price and corresponding demand realization to our data and use least-squares regression on all data seen thus far to obtain new parameter estimates. Nevertheless, these estimates may not converge to the true parameters because the prices are highly correlated as they result from similar optimization problems. We show that providing a discount at some time periods remedies this problem and guarantees convergence.

2.1.5 Structure

In the remainder of this chapter, we present this method and analyze its convergence more formally. Since the method we propose has two components: an estimation and an optimization component, we proceed as follows. In Section 2, we provide some conditions for

the consistency of the least-squares demand parameters estimates using results from the literature. In Section 3, we formally describe a pricing algorithm such that these consistency conditions are satisfied and the prices converge to the optimal price. Finally, in Section 4, we extend the results to the capacitated case.

2.2 Least-squares estimator and convergence conditions

2.2.1 Least squares

At every step the demand parameters will be estimated using least square estimates.

We assume that at time t the firm has observed prices p_1, p_2, \dots, p_{t-1} and demands d_1, d_2, \dots, d_{t-1} . The demand is linear and is such that:

$$d_t = \beta^0 + \beta^1 \cdot p_t + \varepsilon,$$

with ε a noise with mean zero.

We note by $\widehat{\beta}_s = [\widehat{\beta}_s^0, \widehat{\beta}_s^1]$ the vector of parameters estimates at time s . Moreover, if $x_s = [1, p_s]$, then $d_s = x_s' \widehat{\beta}_s + \varepsilon$.

$\widehat{\beta}_t$ is the solution of

$$\widehat{\beta}_t = \arg \min_{r \in \mathbb{R}^2} \sum_{s=1}^{t-1} (d_s - x_s' r)^2.$$

Then given the former prices and demands, the estimates are:

$$\widehat{\beta}_s^1 = \frac{(t-1) \cdot \sum_{s=1}^{t-1} p_s d_s - \sum_{s=1}^{t-1} p_s \sum_{s=1}^{t-1} d_s}{(t-1) \sum_{s=1}^{t-1} p_s^2 - (\sum_{s=1}^{t-1} p_s)^2}$$

$$\widehat{\beta}_s^0 = \frac{\sum_{s=1}^{t-1} d_s}{t-1} - \widehat{\beta}_s^1 \frac{\sum_{s=1}^{t-1} p_s}{t-1}$$

This formula is inefficient to compute least-square estimates. Instead, we will compute these estimates using the recursive formula we discuss in the following proposition.

Proposition 2.2.1. *The least squares estimates can be estimated by the following iterative process.*

$$\widehat{\beta}_t = \widehat{\beta}_{t-1} + H_{t-1}^{-1} \cdot x_{t-1} (d_{t-1} - x'_{t-1} \widehat{\beta}_{t-1}), \quad t = 3, \dots, T,$$

where $\widehat{\beta}_2$ is an arbitrary vector and the matrices H_{t-1} are generated by

$$H_{t-1} = H_{t-2} + x'_{t-1} x_{t-1}, \quad t = 3, \dots, T,$$

with:

$$H_1 = \begin{bmatrix} 1 & p_1 \\ p_1 & p_1^2 \end{bmatrix}.$$

Hence:

$$H_{t-1} = \begin{bmatrix} t-1 & \sum_{s=1}^{t-1} p_s \\ \sum_{s=1}^{t-1} p_s & \sum_{s=1}^{t-1} p_s^2 \end{bmatrix}.$$

Proof. We write

$$\widehat{\beta}_t = \widehat{\beta}_{t-1} + c,$$

where c is a vector. By applying the first order conditions for computing $\widehat{\beta}_{t-1}$

$$\sum_{s=1}^{t-2} (d_s - x'_s \widehat{\beta}_t) \cdot x_s = 0.$$

By applying them to $\widehat{\beta}_t = \widehat{\beta}_{t-1} + c$

$$\sum_{s=1}^{t-1} (d_s - x'_s \widehat{\beta}_t - x'_s \cdot c) \cdot x_s = 0.$$

By subtracting the equations, we have:

$$\sum_{s=1}^{t-1} (x'_s \cdot c) \cdot x_s = (d_{t-1} - x'_{t-1} \widehat{\beta}_{t-1}) \cdot x_{t-1}.$$

Finally

$$c = H_{t-1}^{-1} x_{t-1} (d_{t-1} - x'_{t-1} \widehat{\beta}_{t-1})$$

with

$$H_{t-1} = \begin{bmatrix} t-1 & \sum_{s=1}^{t-1} p_s \\ \sum_{s=1}^{t-1} p_s & \sum_{s=1}^{t-1} p_s^2 \end{bmatrix}.$$

□

It can be shown that matrix H_{t-1} is invertible as long as all the prices are not equal (see [4]).

In all our simulations in the next chapter, we will use this recursion to compute the least square estimates.

2.2.2 Consistency conditions

A critical feature is to find conditions that guarantee strong consistency (i.e. almost sure convergence) of the least squares regression parameters.

Sufficient conditions for convergence of stochastic least-squares regression are given by Lai and Wei [22] for the model

$$y_n = \mathbf{x}_n \boldsymbol{\beta} + \boldsymbol{\epsilon}_n, \quad n = 1, 2, \dots,$$

where $\boldsymbol{\beta}$ is the model parameter vector and y_n is the observed response corresponding to the design level $\mathbf{x}_n = (x_{n1}, \dots, x_{np})^\top$. Moreover, $\mathbf{y} = [y_1, y_2, \dots, y_n]^\top$ are the first n observed responses, $\mathbf{X}_n = (\mathbf{x}_{ij})_{1 \leq i \leq n, 1 \leq j \leq p}$ the matrix of the n first input vectors, and $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \dots, \epsilon_n]^\top$ the first n random errors. The error ϵ_n is assumed to be \mathcal{F}_n -measurable with $\mathbb{E}(\epsilon_n | \mathcal{F}_{n-1}) = 0$ for an increasing sequence of σ -fields $\{\mathcal{F}_n\}$.

For a linear demand model with parameters $\boldsymbol{\beta} = [\beta_0, \beta_1]$, set $\mathbf{x}_i = [1, p_i]$ and let the responses be the sampled demands $y_i = d_i$. The assumption that the errors ϵ_i are i.i.d.

zero-mean is a special case of the condition above.

The conditions are related to the eigenvalues of the inverse covariance matrix as the following theorem by Lai and Wei [22] shows.

Theorem 2.2.2. *Suppose that in the regression model above, ϵ_n is a martingale difference sequence with respect to an increasing sequence of σ -fields $\{\mathcal{F}_{n-1}\}$ such that*

$$\sup_n \mathbb{E}(|\epsilon_n|^\alpha | \{\mathcal{F}_{n-1}\}) \leq \infty \text{ a.s. for some } \alpha > 2.$$

Let λ_{\min} and λ_{\max} be respectively the minimum and maximum eigenvalues of matrix $\mathbf{X}_n^\top \mathbf{X}_n$. Moreover assume that the design levels x_{n1}, \dots, x_{np} at stage n are $\{\mathcal{F}_n\}$ measurable random variable such that

$$\lambda_{\min} \longrightarrow \infty \text{ almost surely, and, } \frac{\log(\lambda_{\max})}{\lambda_{\min}} \longrightarrow 0 \text{ almost surely.}$$

Then the least square estimates b_n converge almost surely to β ; in fact,

$$\max_j |b_{nj} - \beta_j| = O\left(\left[\frac{\log(\lambda_{\max}(n))}{\lambda_{\min}(n)}\right]^{1/2}\right) \text{ a.s.}$$

Proof. Lai Wei 1982 (part 2) [22]. □

In what follows, we provide some bounds on these eigenvalues in order to obtain consistency conditions easier to check in practice.

Denote the unnormalized covariance matrix

$$\mathbf{H}_n^{-1} = \mathbf{V}_n = (\mathbf{X}_n^\top \mathbf{X}_n)^{-1} = \left(\sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top\right)^{-1} = \left(\sum_{i=1}^n \begin{bmatrix} 1 & p_i \\ p_i & p_i^2 \end{bmatrix}\right)^{-1} \triangleq \begin{bmatrix} n & n\bar{p}_n \\ n\bar{p}_n & nt_n^2 \end{bmatrix}^{-1},$$

where \bar{p}_n , s_n^2 and t_n^2 are the sample mean, sample variance and the sample second moment of p_1, \dots, p_n respectively. Assume that it exists, i.e. that $\mathbf{X}_n^\top \mathbf{X}_n$ is invertible. (This is satisfied if $p_i \neq p_j$ for some $1 \leq i < j \leq n$ since the matrices being summed are positive semi-definite.) The matrix $\mathbf{X}_n^\top \mathbf{X}_n$ is assumed to be positive definite. The eigenvalues of $\mathbf{V}_n^{-1} = \mathbf{X}_n^\top \mathbf{X}_n$ are

the roots of its characteristic polynomial. In what follows, we drop the n subscript for the sake of simplicity.

$$(n - \lambda)(nt^2 - \lambda) - n^2\bar{p}^2 = \lambda^2 - (n + nt^2)\lambda + (n^2t^2 - n^2\bar{p}^2)$$

given by

$$\lambda = \frac{(n + nt^2) \pm \sqrt{(n + nt^2)^2 - 4(n^2t^2 - n^2\bar{p}^2)}}{2}$$

and therefore,

$$\lambda_{max} = \lambda_{min} + \sqrt{(n + nt^2)^2 - 4(n^2t^2 - n^2\bar{p}^2)},$$

where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of $\mathbf{V}_n^{-1} = \mathbf{X}_n^\top \mathbf{X}_n$ respectively.

Alternatively, the trace and determinant of $\mathbf{X}_n^\top \mathbf{X}_n$ are

$$\text{Tr}(\mathbf{X}_n^\top \mathbf{X}_n) = \lambda_{max} + \lambda_{min} = n(1 + t_n^2) \quad \text{and} \quad \det(\mathbf{X}_n^\top \mathbf{X}_n) = \lambda_{max}\lambda_{min} = n^2(t^2 - \bar{p}^2).$$

Solving for the eigenvalues similarly yields:

$$\begin{aligned} \lambda &= \frac{n}{2} \left(1 + t^2 \pm \sqrt{(1 + t^2)^2 - 4s^2} \right) \\ &= \frac{n}{2} \left(1 + \bar{p}^2 + s^2 \pm \sqrt{(1 + \bar{p}^2 + s^2)^2 - 4s^2} \right), \end{aligned}$$

Therefore,

$$\begin{aligned} \lambda_{min} &= \frac{n}{2} \left(1 + \bar{p}^2 + s^2 - \sqrt{(1 + \bar{p}^2 + s^2)^2 - 4s^2} \right) \\ &= \frac{n(1 + \bar{p}^2 + s^2)}{2} \left(1 - \sqrt{1 - \frac{4s^2}{(1 + \bar{p}^2 + s^2)^2}} \right) \\ &\geq \frac{ns^2}{1 + \bar{p}^2 + s^2}. \end{aligned}$$

To show the last inequality, we use the following lemma

Lemma 2.2.3. $1 - \sqrt{1 - x} \geq \frac{x}{2}$.

Proof. Notice that $1 - \sqrt{1 - x}$ is a convex function. Hence

$$f(x) - f(0) \geq f'(0) \cdot (x - 0).$$

□

Furthermore, we have that

$$\begin{aligned} \frac{\log(\lambda_{max})}{\lambda_{min}} &= \frac{\log\left(\frac{n}{2}\left(1 + \bar{p}^2 + s^2 + \sqrt{(1 + \bar{p}^2 + s^2)^2 - 4s^2}\right)\right)}{\frac{n}{2}\left(1 + \bar{p}^2 + s^2 - \sqrt{(1 + \bar{p}^2 + s^2)^2 - 4s^2}\right)} \\ &\leq \frac{2 \log(n(1 + \bar{p}^2 + s^2))}{n\left(1 + \bar{p}^2 + s^2 - \sqrt{(1 + \bar{p}^2 + s^2)^2 - 4s^2}\right)} \quad \text{by discarding the } -4s^2 \text{ term} \\ &= \frac{2\left(1 + \bar{p}^2 + s^2 + \sqrt{(1 + \bar{p}^2 + s^2)^2 - 4s^2}\right) \log(n(1 + \bar{p}^2 + s^2))}{4ns^2} \\ &\leq \frac{4(1 + \bar{p}^2 + s^2) \log(n(1 + \bar{p}^2 + s^2))}{4ns^2} \\ &= \frac{(1 + \bar{p}^2 + s^2) (\log n + \log(1 + \bar{p}^2 + s^2))}{ns^2}. \end{aligned}$$

Then the conditions of the theorem by Lai and Wei [22] can be replaced by a simpler one.

Lemma 2.2.4. *The two necessary conditions by Lai and Wei [22]*

$$\lambda_{min} \longrightarrow \infty \text{ almost surely and } \frac{\log(\lambda_{max})}{\lambda_{min}} \longrightarrow 0 \text{ almost surely}$$

are satisfied if:

$$\frac{n \cdot s^2}{\ln(n)} \longrightarrow \infty \text{ almost surely}$$

Proof. We use the fact that the price is bounded. Because we know that $p \in [p_{min}, p_{max}]$ then both the standard deviation and the average price are bounded. Indeed, we have that:

$$\bar{p} \in [p_{min}, p_{max}] \text{ and } s^2 \in [0, (p_{max} - p_{min})^2].$$

Then we have that

$$(1 + \bar{p}^2 + s^2) \leq 1 + p_{max}^2 + (p_{max} - p_{min})^2.$$

Hence a sufficient condition for: $\lambda_{min} \rightarrow \infty$ almost surely is:

$$n.s^2 \rightarrow \infty \text{ almost surely.}$$

Moreover, in order to have: $\frac{\log(\lambda_{max})}{\lambda_{min}} \rightarrow 0$ almost surely, we need:

$$\frac{n.s^2}{\ln(n)} \rightarrow \infty \text{ almost surely.}$$

Since the second condition is more restrictive than the first one, it is enough to have:

$$\frac{n.s^2}{\ln(n)} \rightarrow \infty \text{ almost surely.}$$

□

Lai and Wei [22] mention in their paper that the first condition is sufficient for the case when the inputs are independent. Hence, as expected, the second condition is more restrictive. The power of this theorem is that it gives rise to a simple condition for the cases where the inputs are correlated to each other, which is what applies in our case because we wish to price optimally after observing the past prices.

2.3 Pricing algorithm

Assuming that demand is of the form $d(p) = a - b.p + \epsilon$ and that at every step we compute the estimated parameters a_n and b_n of a and b , we develop a pricing policy that satisfies the previous conditions in order to ensure consistency of the estimates.

The basic idea of this algorithm is to optimize the revenue over the interval $[p_{min}, p_{max}]$, where p_{min} and p_{max} are such that the price is not below a threshold and that the demand is typically nonnegative (at least with high probability). Moreover, we assume that we know based on historical demand data that the optimum should lie within M and N such that:

$$p_{min} \leq M \leq N \leq p_{max}.$$

Intuitively, we price most of the time between M and N , but occasionally we offer a discount. The idea is that providing this discount will improve the learning of the demand parameters and satisfy the condition for the consistency of these parameters. Below, we describe the algorithm formally.

Algorithm:

1. Let a_n and b_n denote the parameters estimated through regression on the prices and corresponding observed demands thus far.
2. Set:

$$p_n = \max\left(\min\left(\frac{b_n}{2a_n}, N\right), M\right)$$

if there does not exist i such that $n = \lfloor 2^{\sqrt{i}} \rfloor$. Otherwise, set

$$p_n = \max\left(\min\left(\frac{b_n}{2a_n}, N\right), M\right) - \gamma.$$

3. Update the parameter estimates and repeat the procedure.

Note that the choice of prices is coherent with Equation (2.1). The intuition is that when

there is no discount, we choose the price that maximizes the expected revenue with the estimated demand parameters, ensuring at the same time that it lies within M and N . We give a discount γ at some time periods but this discount is offered less and less often as time goes by since learning has improved.

Throughout the remainder of this section we denote $p_{est}(n) = \max(\min(\frac{b_n}{2a_n}, N), M)$. This corresponds to the estimate of the optimal price.

We provide a condition on the discount γ to guarantee convergence of the parameters and satisfy the constraint that the price should never lie below the threshold p_{min} .

We have $M \leq p_n \leq N$ when no discount is offered. We define $\bar{p}_n = \frac{1}{n} \cdot \sum_{i=1}^n p_i$. It is easy to show that for all $\eta > 0$, for n big enough, we have: $M - \eta \leq \bar{p}_n \leq N$. This is due to the fact that we discount a fixed amount less and less often hence the average discount goes to zero.

With this pricing algorithm, we prove in the next lemma that the previous consistency condition is satisfied.

Lemma 2.3.1. *Assume that we have a demand function of the form $d(p) = a - b.p + \epsilon$ and that at every step we compute the estimated parameters a_n and b_n using linear regression. We pick the next price as:*

$$p_n = \max \left(\min \left(\frac{b_n}{2a_n}, N \right), M \right) + \gamma_n,$$

where

$$\gamma(n) = \begin{cases} -\gamma & \text{if there exist } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

where $[N, M]$ is a price range within we want the price to lie most of the time and γ constant such that:

$$2(N - M) < \gamma \leq M - p_{min}.$$

We also assume that the noise is such that:

$$\mathbb{E}(|\epsilon|^\alpha) \leq \infty \text{ a.s for some } \alpha > 2.$$

Then we have:

$$\frac{n \cdot s_n^2}{\ln(n)} \longrightarrow \infty \text{ a. s.}$$

Remark: This is the condition we wanted from Lemma 2.2.4.

Proof.

$$s_n^2 = \frac{1}{n} \cdot \sum_{i=1}^n [p_i - \bar{p}_n]^2$$

We can rewrite

$$n \cdot s_n^2 = \frac{1}{n} \cdot \sum_{i=1}^n (p_i - p_{est}(i) + p_{est}(i) - \bar{p}_n)^2.$$

Then by discarding the last square term:

$$n \cdot s_n^2 \geq \sum_{i=1}^n (p_i - p_{est}(i))^2 + 2 \cdot (p_{est}(i) - p_i)(\bar{p}_n - p_{est}(i)).$$

For all n , there exist a k_n such that:

$$\lfloor 2^{\sqrt{k_n}} \rfloor \leq n \leq \lfloor 2^{\sqrt{k_n+1}} \rfloor.$$

We fix ϵ . Then there exists n_0 such that for all $n \geq n_0$, for all i , we have $\bar{p}_n - p_{est}(i) \geq M - N - \epsilon$.

Hence we have for $n \geq n_0$

$$\begin{aligned}
n \cdot s_n^2 &= \sum_{i=1}^n [p_i - \bar{p}_n]^2 \\
&\geq \sum_{i=1}^n (p_i - p_{est}(i))^2 + 2((p_{est}(i) - p(i))(\bar{p}_n - p_{est}(i))) \\
&\geq (k_n - 3) \cdot [\gamma^2 + 2 \cdot \gamma \cdot (M - N - \epsilon)] \text{ (counting when the discount is offered)}
\end{aligned}$$

Note that we have $k_n - 3$ and not k_n in the equation above. It is due to the fact that $\lfloor 2\sqrt{n} \rfloor$ may have the same value for different n when n is small, so we need to make sure not to overcount. Then we have that there exists n_0 , such that for all $n \geq n_0$:

$$n \cdot s_n^2 \geq (k_n - 3) \cdot [\gamma^2 + 2 \cdot \gamma \cdot (M - N - \epsilon)].$$

Nevertheless, we want to make sure that the right hand side actually goes to infinity. Then we need: $\gamma^2 + 2 \cdot \gamma \cdot (M - N - \epsilon) > 0$ hence the condition is $\gamma > 2(N - M)$, by choosing ϵ accurately.

The price has to be greater than the minimum price so we also want $M - \gamma \geq p_{min}$.

In summary, the conditions we need are:

$$2(N - M) < \gamma \leq M - p_{min}.$$

This implies that N and M need to be close enough such that

$$2(N - M) < M - p_{min}.$$

Therefore:

$$n \cdot s_n^2 \geq (k_n - 3) \cdot C$$

with $C = \gamma^2 + 2\gamma \cdot (M - N - \epsilon) > 0$.

Then we have:

$$\begin{aligned} \frac{n \cdot s_n^2}{\ln(n)} &\geq \frac{(k_n - 3) \cdot C}{\ln(\lfloor 2^{\sqrt{k_n+1}} \rfloor)} \\ &\geq \frac{(k_n - 3)C}{\ln(2^{\sqrt{k_n+1}})} \\ &= \frac{(k_n - 3)C}{\ln(2) \cdot \sqrt{k_n + 1}} \\ &\sim \frac{\ln(n) \cdot C}{[\ln(2)]^2}. \end{aligned}$$

We can conclude that the condition is satisfied because k_n goes to infinity. □

Then the following theorem applies. This is the main result of the thesis for the single product case.

Theorem 2.3.2. *Assume that we have a demand function of the form $d(p) = a - b \cdot p + \epsilon$, and that at every step we compute the estimated parameters a_n and b_n using linear regression. Moreover, we pick the next price such that:*

$$p_n = \max \left(\min \left(\frac{b_n}{2a_n}, N \right), M \right) + \gamma_n$$

where

$$\gamma_n = \begin{cases} -\gamma & \text{if there exists } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

where N and M is a price range within which we want to price most of the time. Furthermore, γ is such that:

$$2(N - M) < \gamma \leq M - p_{\min}.$$

We also assume that the noise is such that

$$\mathbb{E}(|\epsilon|^\alpha) \leq \infty \text{ a. s. for some } \alpha > 2.$$

Then the following holds:

1. The estimated demand parameters a_n and b_n converge almost surely to the true parameters a and b . Moreover the convergence rate is $\frac{1}{\ln(n)^{0.5}}$ which means that

$$\max|a_n - a, b_n - b| = O\left(\frac{1}{\ln(n)^{0.5}}\right) \text{ a.s.}$$

2. If the bounds are adequate, i.e. $2(N - M) < M - p_{\min}$, the price converges in the Cesaro sense to the optimal price which means that the average price converges to the optimal price.
3. A subset $\mathfrak{A}(\mathbb{N}) = \{n \mid \text{there exists no } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor\}$ of the prices converges to the optimal price almost surely. Moreover, on this subset, the expected revenue also converges to the optimal revenue.
4. The average revenue converges also almost surely to the optimal revenue.

Proof. 1. It results directly from Theorem 2.2.2 since the conditions of Lai and Wei [22] hold. [22] also gives us the convergence rate of the parameters. Note that the convergence rate is slow.

2. We know that $a_n \rightarrow a$ and that $b_n \rightarrow b$ almost surely.

As a result, $\frac{a_n}{b_n}$ converges to $\frac{a}{b}$ almost surely. We assume that the upper and lower bounds have been fixed accordingly so that $M \leq \frac{a}{b} \leq N$. Then we write:

$$p_n = \max\left(\min\left(\frac{b_n}{2a_n}, N\right), M\right) + \gamma_n,$$

where

$$\gamma(n) = \begin{cases} -\gamma & \text{if there exist } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

We are going to show that $\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n p_i = p^*$ a.s.

We have that:

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n p_i &= \frac{1}{n} \sum_{i=1}^n \max(\min(\frac{b_i}{2a_i}, N), M) + \gamma_i \\ &= \frac{1}{n} \sum_{i=1}^n \max(\min(\frac{b_i}{2a_i}, N), M) + \frac{1}{n} \sum_{i=1}^n \gamma_i \end{aligned}$$

The first part of the sum converges almost surely to $\frac{b}{2a}$ and the second part goes to zero, hence the result.

3. This is a direct consequence of Part 1.
4. The average revenue converges to the optimal revenue also almost surely

It follows that:

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n p_i d_i &= \frac{1}{n} \sum_{i=1}^n p_i \cdot (a - b \cdot p_i + \epsilon_i) \\ &= a \frac{1}{n} \sum_{i=1}^n p_i - b \frac{1}{n} \sum_{i=1}^n p_i^2 + \frac{1}{n} \sum_{i=1}^n \epsilon_i p_i. \end{aligned}$$

Moreover,:

$$\frac{1}{n} \sum_{i=1}^n p_i \longrightarrow \frac{a}{2b} \text{ a.s.}$$

and:

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n p_i^2 &= \sum_{i=1}^n \frac{1}{n} (\max(\min(\frac{b_i}{2a_i}, N), M) + \gamma_i)^2 \\ &= \frac{1}{n} \left[\sum_{i=1}^n (\max(\min(\frac{b_i}{2a_i}, N), M))^2 + 2 \sum_{i=1}^n (\max(\min(\frac{b_i}{2a_i}, N), M) \cdot \gamma_i + \sum_{i=1}^n \gamma_i^2) \right]. \end{aligned}$$

The first part of the sum converges to $\frac{a^2}{4b^2}$ and the two other parts go to zero because $\max(\min(\frac{b_i}{2a_i}, N), M)$ is bounded and the sum goes to zero. Then we can conclude that

$$\frac{1}{n} \sum_{i=1}^n p_i^2 \longrightarrow \frac{a^2}{4b^2} \text{ a.s.}$$

Moreover we know that all the prices are bounded, for example by $[p_{min}, p_{max}]$.

Then we use Theorem 5.1.2 in [14].

$$\sum_{i=1}^n \frac{1}{n} p_i \epsilon_i \longrightarrow 0 \text{ a.s.}$$

Then we are able to conclude the proof since:

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n p_i d_i &= \frac{1}{n} \sum_{i=1}^n p_i \cdot (a - b \cdot p_i + \epsilon_i) \\ &= a \frac{1}{n} \sum_{i=1}^n p_i - b \frac{1}{n} \sum_{i=1}^n p_i^2 + \frac{1}{n} \sum_{i=1}^n \epsilon_i p_i \\ &\xrightarrow{n \rightarrow \infty} a \cdot \frac{a}{2b} - b \frac{a^2}{4b^2} \\ &= \frac{a^2}{4b}. \end{aligned}$$

□

Note that in practice, on an infinite horizon, the notion of an average revenue may not be relevant and a discount may need to enter into consideration.

So far in this chapter, we proposed a pricing strategy for the linear demand case that enables both the average revenue and the average price to converge almost surely. Moreover, if we restrict $n \in \mathbb{N}$ to a subsequence, then p_n converges almost surely.

We note that the results hold when, instead of providing a discount, a premium is charged as long as the premium is still big enough.

2.4 Single product with capacity

2.4.1 Model

In this part of the thesis, we consider the single product case with an extra capacity constraint. That is, we also require the expected demand to be restricted by some capacity. Moreover, as before the parameters are unknown.

Mathematically, the objective is to maximize the expected revenue subject to this extra constraint. We want to solve:

$$\begin{aligned}
 & \max p \cdot \mathbb{E}d(p) \\
 & \text{s.t } p_{min} \leq p \leq p_{max} \\
 & \mathbb{E}d(p) = a - b \cdot p \leq d_{max}
 \end{aligned} \tag{2.2}$$

This is equivalent to:

$$\begin{aligned} & \max p \cdot (a - b \cdot p) \\ & \text{s. t } \max(p_{min}, \frac{a - d_{max}}{b}) \leq p \leq p_{max}. \end{aligned}$$

Lemma 2.4.1. *The solution of the problem above is:*

$$p^* = \begin{cases} p_{max} & \text{if } \frac{a}{2b} \geq p_{max} \\ \frac{a}{2b} & \text{if } \max(p_{min}, \frac{a - d_{max}}{b}) \leq \frac{a}{2b} \leq p_{max} \\ \max(p_{min}, \frac{a - d_{max}}{b}) & \text{if } \frac{a}{2b} \leq \max(p_{min}, \frac{a - d_{max}}{b}) \end{cases}$$

Proof. This is an easy application of the KKT Conditions. □

As before, the principal difficulty is that the demand parameters are unknown. Moreover, an additional difficulty arises from the fact that the capacity constraint can be binding.

2.4.2 Perturbation pricing

To solve the problem with a capacity constraint, we use a very similar method as in the problem without capacity (see Section 2.3). At every step, a_n and b_n are estimated using linear regression. We denote p_n^* as the solution of the optimization problem using the estimated parameters. As in the single product case without any capacity, we assume that we know that the optimal price lies within M and N . We use the algorithm described below.

Algorithm:

1. Let a_n and b_n denote the parameters estimated through regression on the prices and corresponding observed demands thus far.
2. Set:

$$p_n = \max(\min(p_n^*, N), M)$$

if there does not exist i such that $n = \lfloor 2^{\sqrt{i}} \rfloor$. Otherwise, set

$$p_n = \max(\min(p_n^*, N), M) - \gamma.$$

3. Update the parameter estimates and repeat the procedure.

Then we have the following theorem:

Theorem 2.4.2. *Assume that we have a demand function of the form $d(p) = a - b.p + \epsilon$, and that at every step we compute the estimated parameters a_n and b_n using linear regression.*

We set the next price such as:

$$p_n = \max(\min(p_n^*, N), M) + \gamma_n$$

where

$$\gamma_n = \begin{cases} \gamma & \text{if there exist } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

where N and M is a price range within we want to be most of the time. Furthermore, γ is such that:

$$2(N - M) < \gamma \leq p_{max} - N.$$

We also assume that the noise is such that

$$\mathbb{E}(|\epsilon|^\alpha) \leq \infty \text{ a.s for some } \alpha > 2.$$

Then the following holds:

1. The demand parameters a_n and b_n converge almost surely to true real parameters a and b .
2. If the bounds M and N are adequate, i.e. $2(N - M) < M - p_{min}$ the price converges in the Cesaro sense to the optimal price which means that the average price converges to the optimal price.

3. A subset $\mathfrak{A}(\mathbb{N}) = \{n \mid \text{there exists no } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor\}$ of the prices converges to the optimal price almost surely. Moreover on this subset the revenue also converges to the optimal revenue.

4. The average revenue converges also almost surely to the optimal revenue .

Proof. The proof is similar to the single product case without capacity. The only difference is that the optimal price is different. We first show that the demand parameters converge almost surely and everything follows. Note that we charge a premium instead of providing a discount. This is in order to make sure that the capacity constraint is not violated. \square

Note that we still need the condition

$$\gamma > 2(N - M).$$

Hence M and N must be known with enough precision in order to avoid too big discounts.

2.5 Conclusions

This pricing method we proposed is useful because it guarantees joint learning and pricing. In Chapter 4 we extend this method to a multi product setting.

This policy makes sense in practice. On a new product we provide frequent discounts when it is launched in order to enable customers to get familiar with the product. As time goes by, the discounts on the product are less and less frequent because the consumers are really familiar with the product and the suppliers do not need to learn the demand function anymore.

The setting with no capacity constraints is realistic for “make to order” firms assuming that they can serve any demand without any backlogging as well as for firms who do not sell “physical goods” such as buying songs on Itunes.com or downloading movies on Netflix.com.

The capacitated case applies when there is a limitation on the number of goods that can be supplied.

In the next chapter, we will see through simulations that this algorithm performs well for a variety of noise distributions. We also illustrate the correlation between accuracy of the solution and length of the learning phase as well as variance of the demand noise. One caveat is that we need precise bounds N and M on the prices. Nevertheless, in practice we can enhance the method and get rid of this requirement.

Chapter 3

Simulations: Affine demand, single product case

In this chapter, we run simulations to see how well the pricing algorithm we propose works in practice.

3.1 Different Algorithms

3.1.1 Standard algorithm

First we test our algorithm for the following demand function

$$d(p) = 300 - p + \varepsilon.$$

The noise ε has a Normal distribution with mean zero and standard deviation 10. Moreover, we use as starting prices $P_1 = 130$ and $P_2 = 140$. We assume that we know that the optimal price lies within the following bounds: $M = 130$ and $N = 170$. The expected optimal revenue is $\mathbb{E}(\Pi^*) = 150 \cdot 150 = 22,500$.

On Figure 3-1, we can see a typical output for the algorithm and in particular at step n

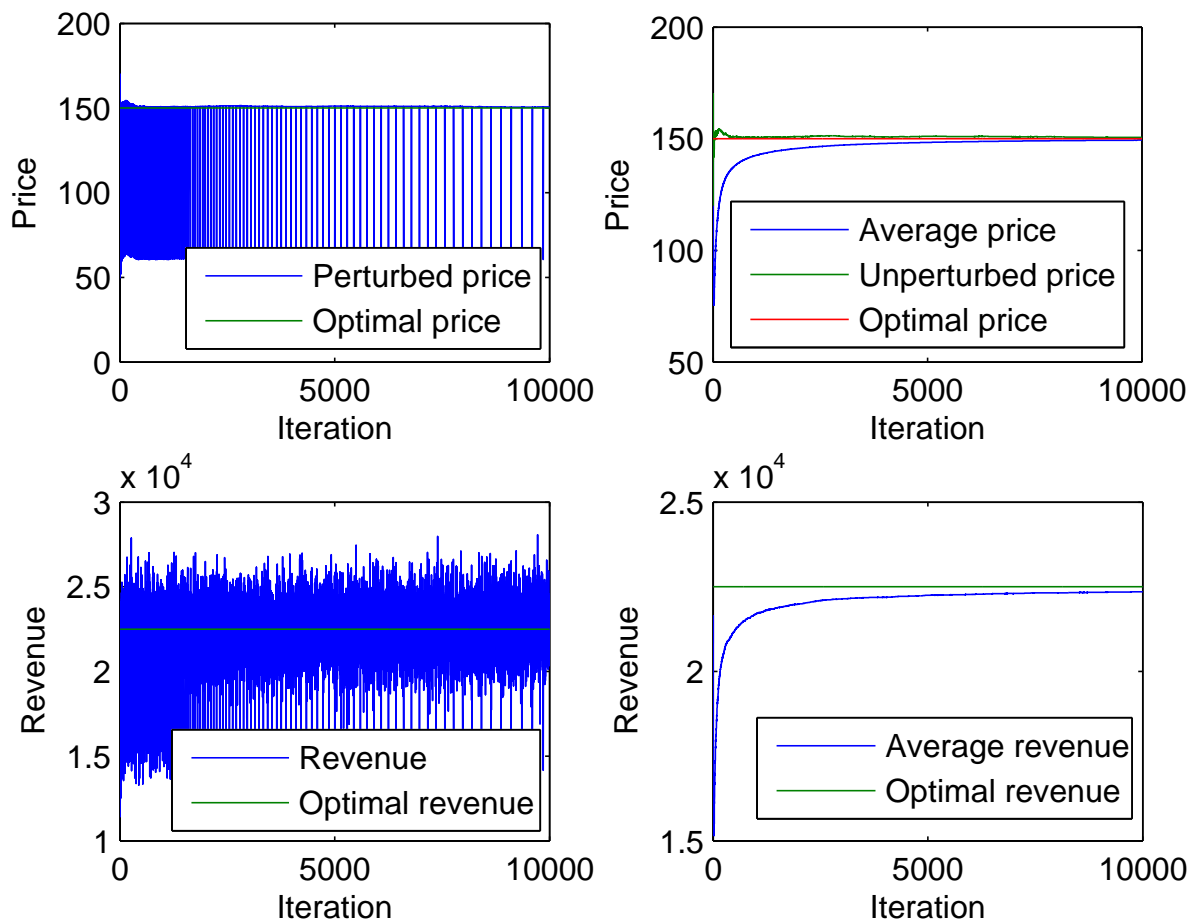


Figure 3-1: Single product, standard algorithm

the behavior of

- the price set, the corresponding unperturbed price, and the average price $\frac{1}{n} \sum_{i=1..n} p_i$.
- the revenue as well as the average revenue $\frac{1}{n} \sum_{i=1..n} p_i \cdot d_i$.

Now we repeat the algorithm 10 times. We plot the estimates of the demand parameters on Figures 3-2 and 3-3. Table 3.1 represents the demand estimates as well as prices and revenue at different steps of the algorithm. We can see that the learning of the param-

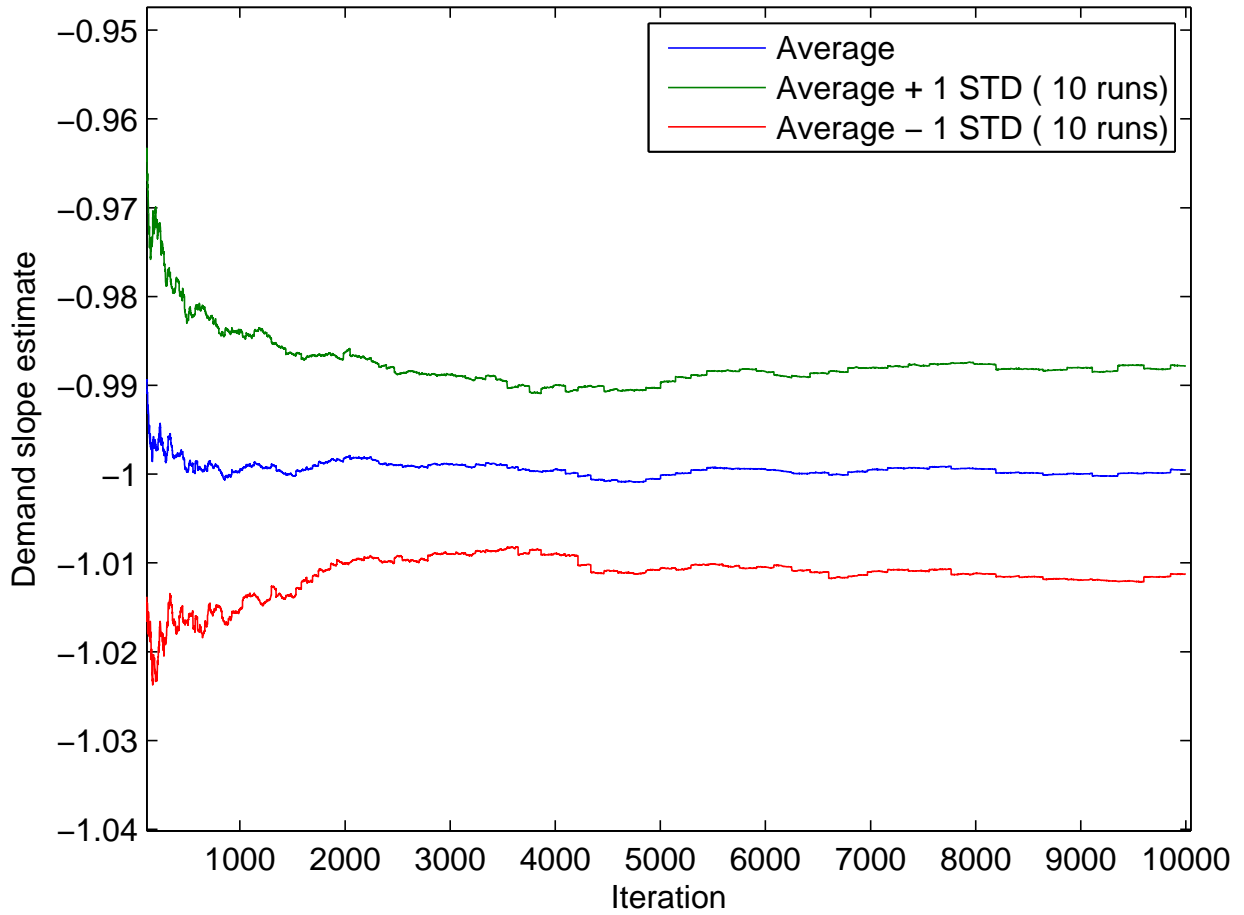


Figure 3-2: Estimates of the slope

ters, as well as of the optimal price, improves. However, this algorithm might seem hard to implement in practice because precise bounds on the optimal price are required and the perturbation added on the price is significant.

We now present an algorithm, that does not need bounds on the optimal price.

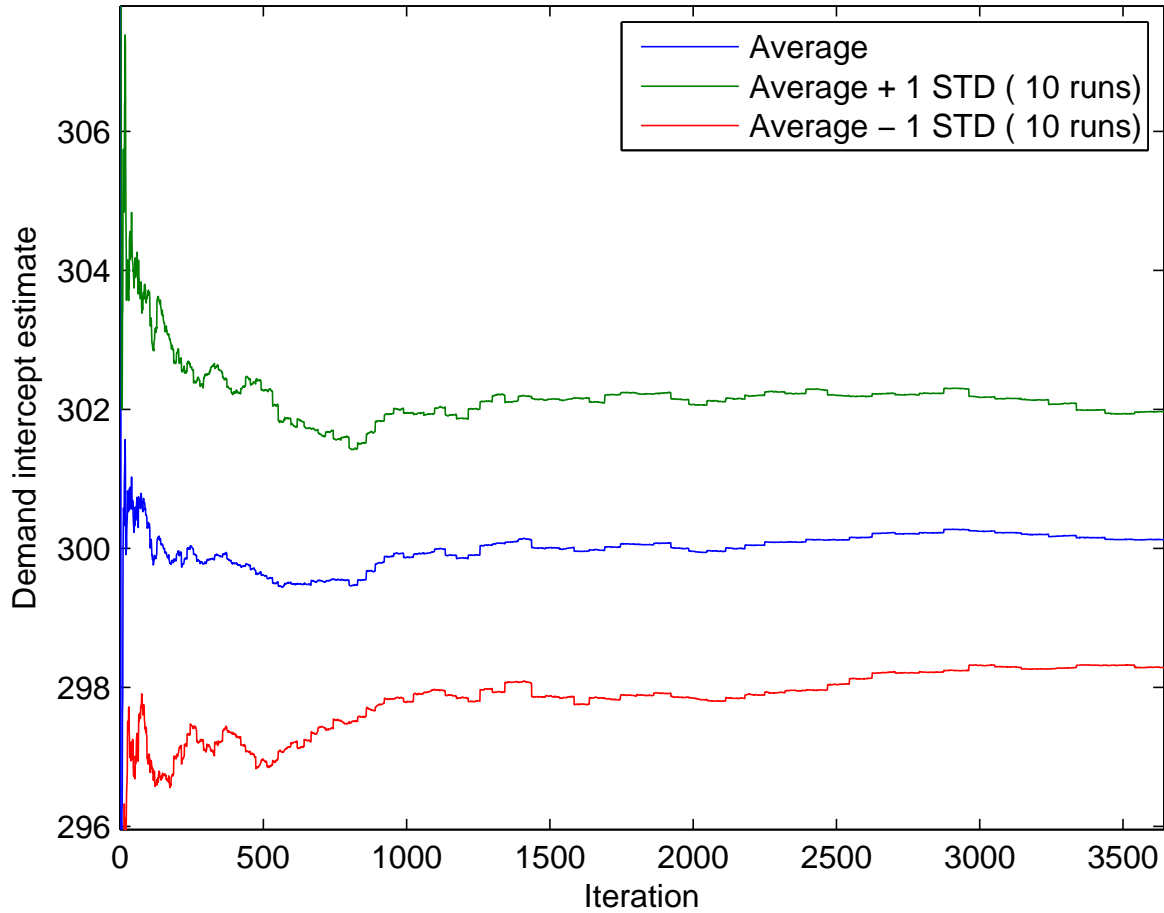


Figure 3-3: Estimates of the intercept

	Iteration 100	Iteration 1000	Iteration 10000
Mean(Intercept)	298.4258819	299.7966774	299.8311603
STD(Intercept)	2.952332878	2.365947719	1.828554929
Mean(Slope)	-0.986503323	-0.998507603	-0.998871602
STD(Slope)	0.023383146	0.017198974	0.012401486
Mean(Price)	151.3437163	150.1286527	150.0951805
STD(Price)	2.3953609	1.397586458	0.951329084
Mean(Exp. revenue)	22493.03045	22498.22553	22499.17642
STD(Exp. revenue)	9.53030223	1.515791532	0.804847111

Table 3.1: Standard algorithm

3.1.2 Improved algorithm with transient phase.

In this section, we tackle the same problem when the bounds we have on the prices are too loose to enable us to use the random perturbation method right away.

The idea is that if we have no indication about the optimal price, the bounds are

$$M = P_{min} = c \text{ and } N = P_{max} = d.$$

We split this interval into j subintervals such that

$$M(k) = c + \frac{k \cdot (d - c)}{j} \text{ and } N(k) = c + \frac{(k + 1)(d - c)}{j} \text{ with } 0 \leq k \leq j - 1.$$

Then we start pricing in the first interval with bound $M(0)$ and $N(0)$. As soon as the optimal price reaches the upper bound $N(0)$ m times (with m an arbitrary number which is the parameter of the algorithm), we assume that we can conclude that the optimal price is in an upper interval. The intuition is that when we hit the upper boundary many times, the optimal price will be above this boundary. Then we move up and price using the bounds $M(1)$ and $N(1)$ and repeat the same procedure. This phase before we reach the optimal interval is *the transient phase*.

It makes sense that for m big enough, this algorithm is going to converge to the optimal price. In the next simulations, we will show numerically that even a small m guarantees convergence in practice. We will use $m=20$.

We run again simulations for the same demand functions and noises as before. Figure 3-4 shows a typical output for the algorithm. Note the transient phase at the beginning as well as the much smaller magnitude of the perturbation. As before, Table 3.2 gives the prices and expected revenue of this algorithm.

Note that at iteration 100, we are still in the transient phase. This algorithm does

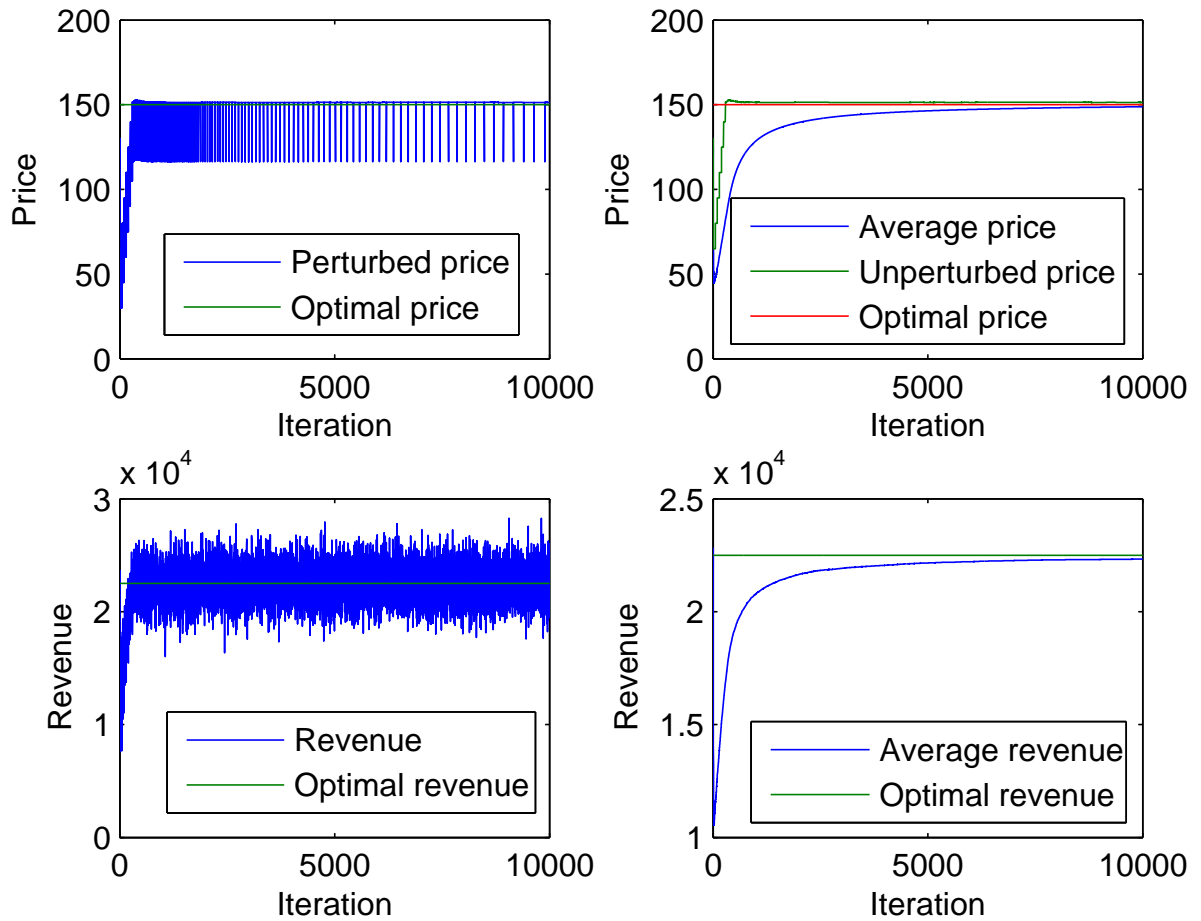


Figure 3-4: Single product, improved algorithm

	Iteration 100	Iteration 1000	Iteration 10000
Mean(Intercept)	299.5545713	299.8489679	299.809223
STD(Intercept)	4.026505575	0.774045259	0.79425532
Mean(Slope)	-0.993759293	-0.997554443	-0.998339059
STD(Slope)	0.082585621	0.006283208	0.005636395
Mean(Price)	74	150.2976043	150.1563104
STD(Price)	0	0.623281467	0.459431297
Mean(Exp. revenue)	16724	22499.5618	22499.7856
STD(Exp. revenue)	0	0.434997087	0.275381129

Table 3.2: Improved algorithm

not require any bound on the optimal price and has similar performances to the previous algorithm. Hence, we will use it during most of our simulations.

3.1.3 Capacitated case algorithm

The last case is the algorithm in the presence of capacity constraints. We assume that we do not know precise bounds on the optimal price.

The observed demand function is

$$d(p) = 300 - p + \varepsilon.$$

We will consider the following capacity constraint:

$$\mathbb{E}d(p) = 300 - p \leq 130.$$

Hence we want $p \geq 170$. In this case, the optimal price is $p^* = 170$ and the maximal expected profit is

$$\mathbb{E}\Pi(p^*) = p^* \cdot \mathbb{E}(d(p^*)) = 22,100.$$

In order to solve this case, we use an algorithm similar to the improved algorithm described above. But because we have the demand capacity constraint, instead of starting pricing from the lowest interval and increase the price, we will start pricing from the highest interval and decrease the price.

Following the notations above, we start pricing in the last interval with bounds $M(j-1)$ and $N(j-1)$. As soon as the optimal price for the demand constraint reaches the lower bound $M(j-1)$ m times (with m an arbitrary number which is the parameter of the algorithm), we assume that we can conclude that the optimum is in a lower interval. Then we move down and price using the bounds $M(j-2)$ and $N(j-2)$ and repeat the same procedure.

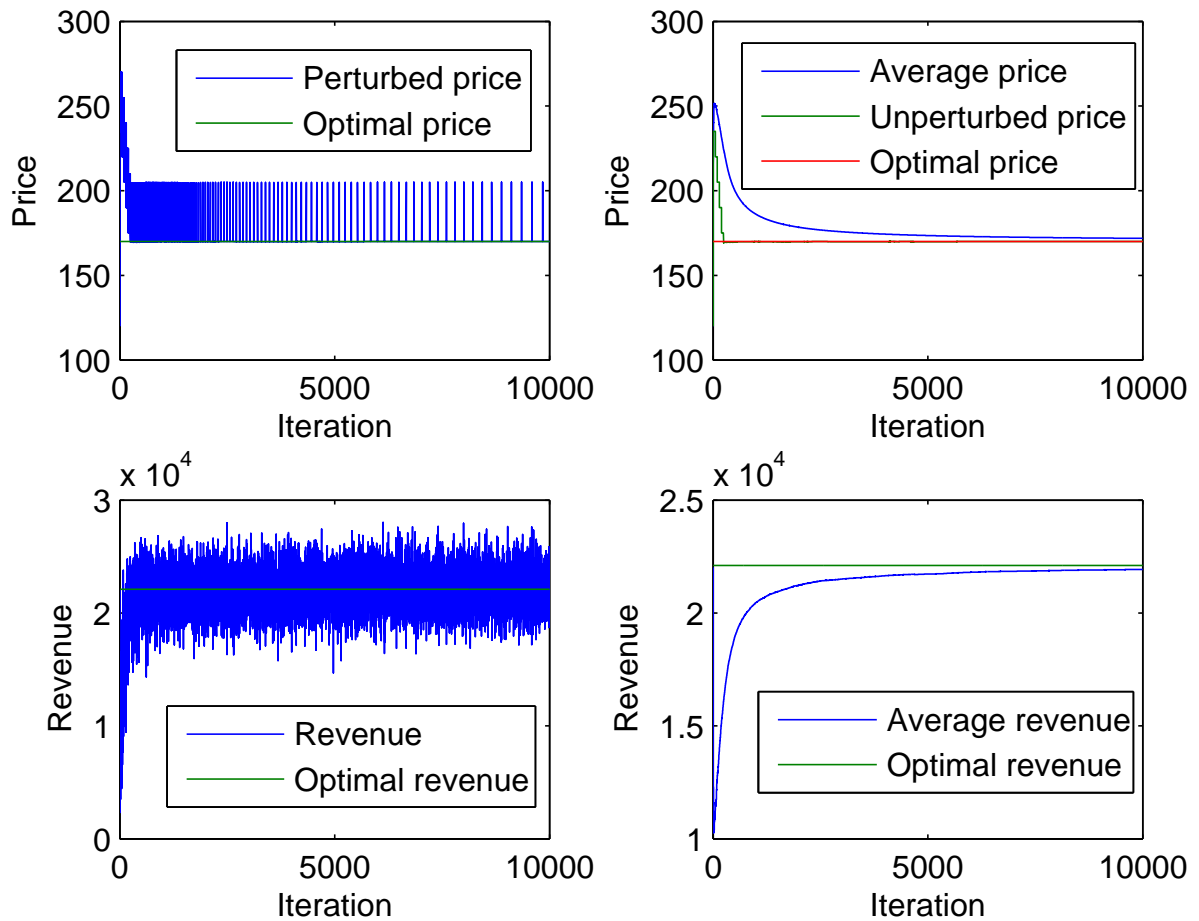


Figure 3-5: Single product with capacity

Figure 3-5 represents a typical output of the algorithm. Note that instead of offering a discount, we charge a premium to make sure that the demand capacity constraint is never violated.

We then run the algorithm 10 times and on Table 3.7 give the value for the estimated parameters, prices as well as expected revenue. We see that the accuracy is of the same order as the noncapacitated case for revenue and prices.

	Iteration 100	Iteration 1000	Iteration 10000
Mean(Intercept)	297.3647393	301.2906311	301.4223219
STD(Intercept)	9.449566541	4.660085273	5.008984546
Mean(Slope)	-0.989337712	-1.005833019	-1.007942529
STD(Slope)	0.037514594	0.024430805	0.029068429
Mean(Price)	205	170.2859381	170.0695446
STD(Price)	0	0.547175908	0.100124127
Mean(Exp. revenue)	19475	22088.21125	22097.20436
STD(Exp. revenue)	0	22.10664888	4.017414367

Table 3.3: Single product case with capacity

3.2 Sensitivity analysis

We will conclude this chapter with performing some sensitivity analysis on some parameters: initial prices, slope and intercept of the demand function, variance and distribution of the noise as well as the capacity.

3.2.1 Input: Initial prices

We consider the same demand function as before

$$d(p) = 300 - p + \varepsilon,$$

where ε has a Normal distribution with mean zero and standard deviation $\sigma = 10$.

We consider the five following initial prices, which corresponds to the two first prices set for the algorithm.

$$P = [5, 35], P = [3, 95], P = [5, 155], P = [5, 215] \text{ and } P = [5, 285].$$

Note that because we have two parameters to estimate, we always need to start with two initial prices.

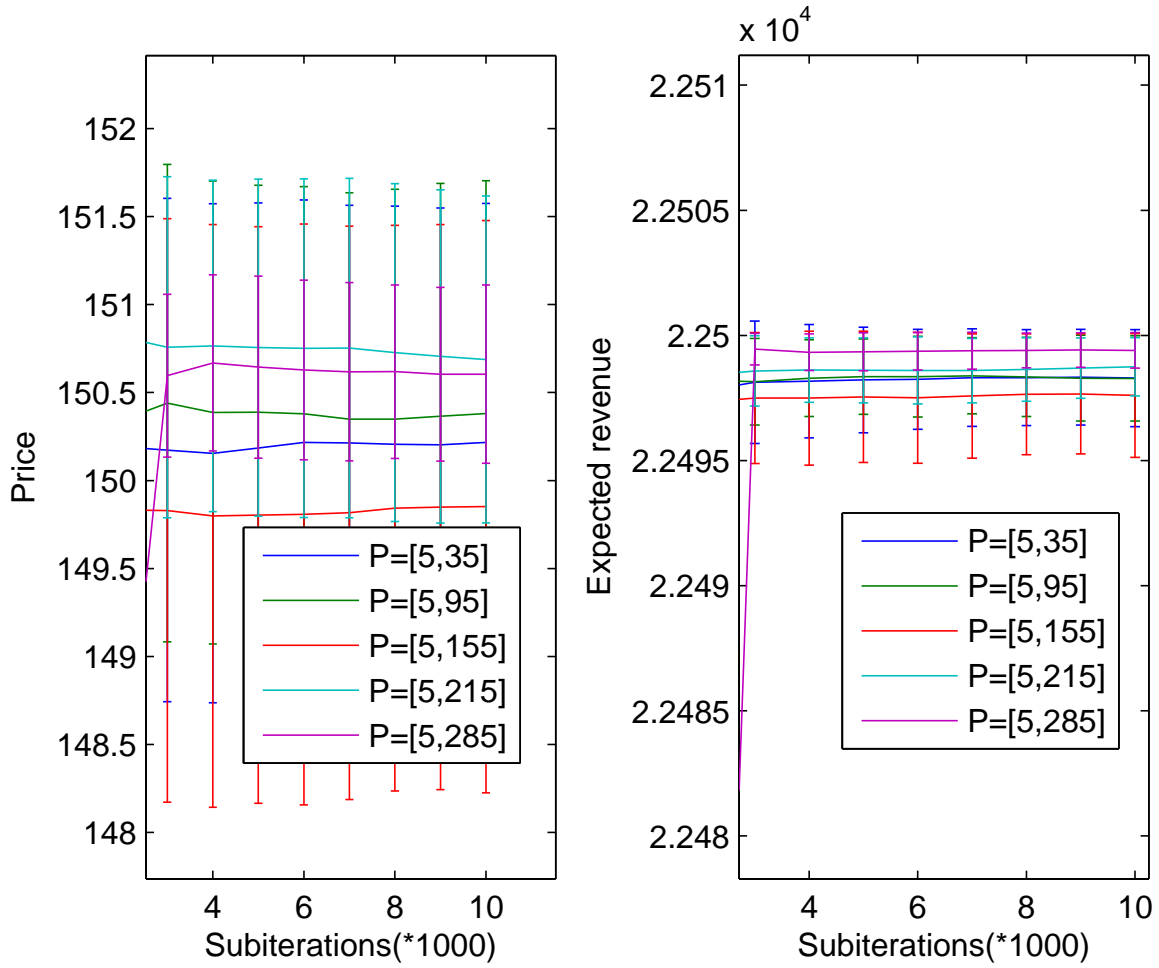


Figure 3-6: Sensitivity to the starting prices

On Figure 3-6, we represent the unperturbed price and the expected revenue as a function of the number of iterations for the five stated scenarios. As before, we run each scenario ten times and then consider the average as well as the standard deviation of the price.

We can conclude that there is no clear dependence on the initial prices. Another interesting observation is that the standard deviation decreases very slowly over time. This is probably due to the bad convergence rate of the pricing algorithm.

Intercept	100	300	500	700	900
Mean(D p-opt p)^a(%)	1.94313	0.144674	0.104305	0.082584	0.024464
STD(D p-opt p)(%)	0.739274	0.140144	0.085162	0.05029	0.014748
Mean(D exp r-opt r)^b(%)	0.042676	0.000386	0.000174	9.1E-05	7.94E-06
STD(D exp r-opt r) (%)	0.027837	0.000512	0.000242	7.99E-05	6.7E-06

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 3.4: Sensitivity to the intercept

3.2.2 Demand parameters: Slope and intercept

Now we will change the slope and intercept of the demand function. We take $[P_1, P_2] = [3, 5]$ and the noise ε has a Normal distribution with mean 0 and standard deviation 10. We note $d(p) = a - b.p + \varepsilon$. We consider the following cases: a fixed slope $b = 1$ and intercepts $a = 100, 300, 500, 700, 900$, intercept $a = 1000$ and slopes $b = 1, 2, 3, 4, 5$.

We use the improved version of the algorithm with the same size for all the subintervals. We represent the distance from the unperturbed price to the optimal price $|\frac{P-P^*}{P^*}| \cdot 100$, and from the expected revenue to the optimal revenue $|\frac{\mathbb{E}\Pi(P) - \mathbb{E}\Pi(P^*)}{\mathbb{E}\Pi(P^*)}| \cdot 100$ as function of the number of iterations.

We see on Figures 3-7 and 3-8 that the transient phase gets longer when the intercept increases and when the slope decreases. This is due to the fact that the optimal price gets farther away from the original interval.

Moreover, the longer the transient phase, the more accurate the final solution is eventually. This is intuitive because the longer the learning phase, the more spread out the inputs are. For example, for different values of the intercept, Table 3.4 provides the accuracy of the final price and of the final revenue.

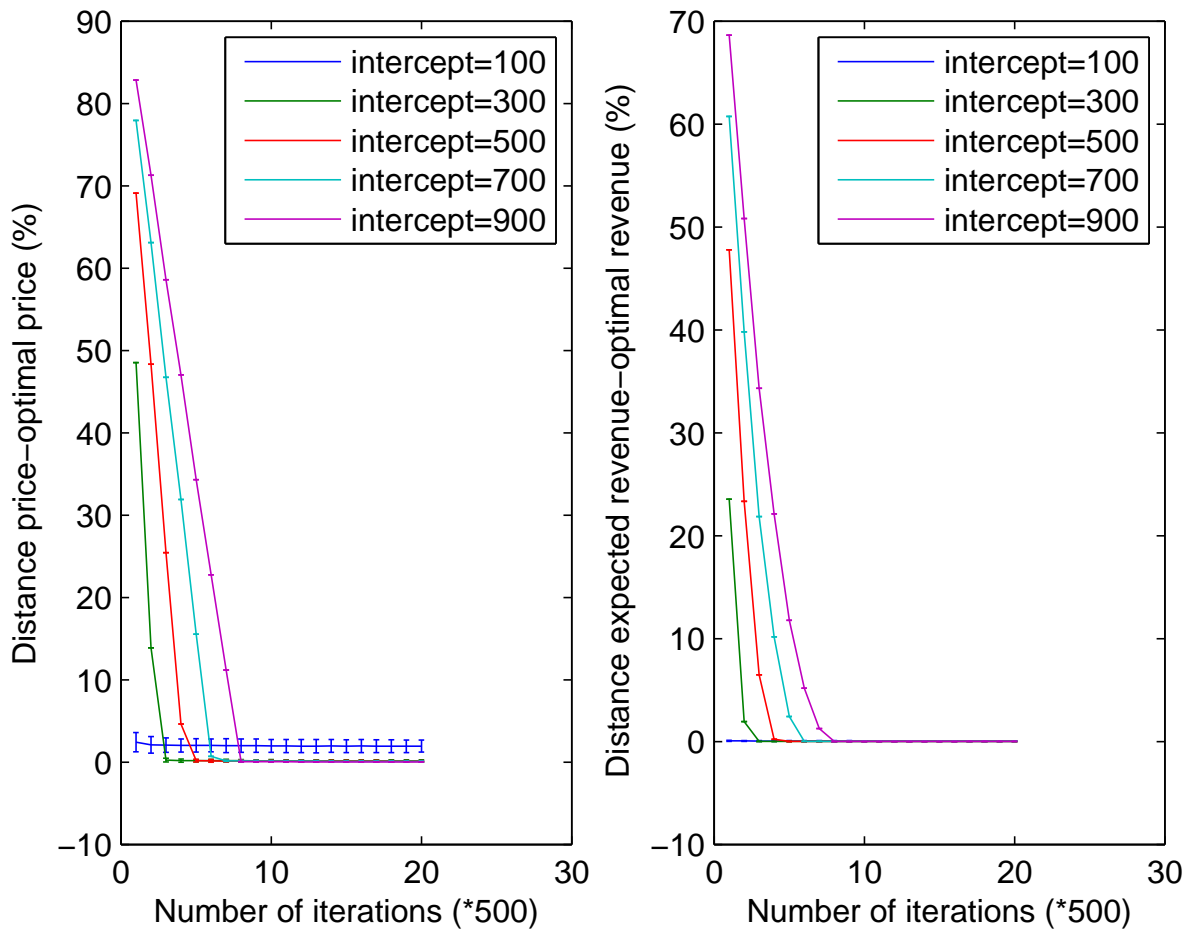


Figure 3-7: Sensitivity to the intercept

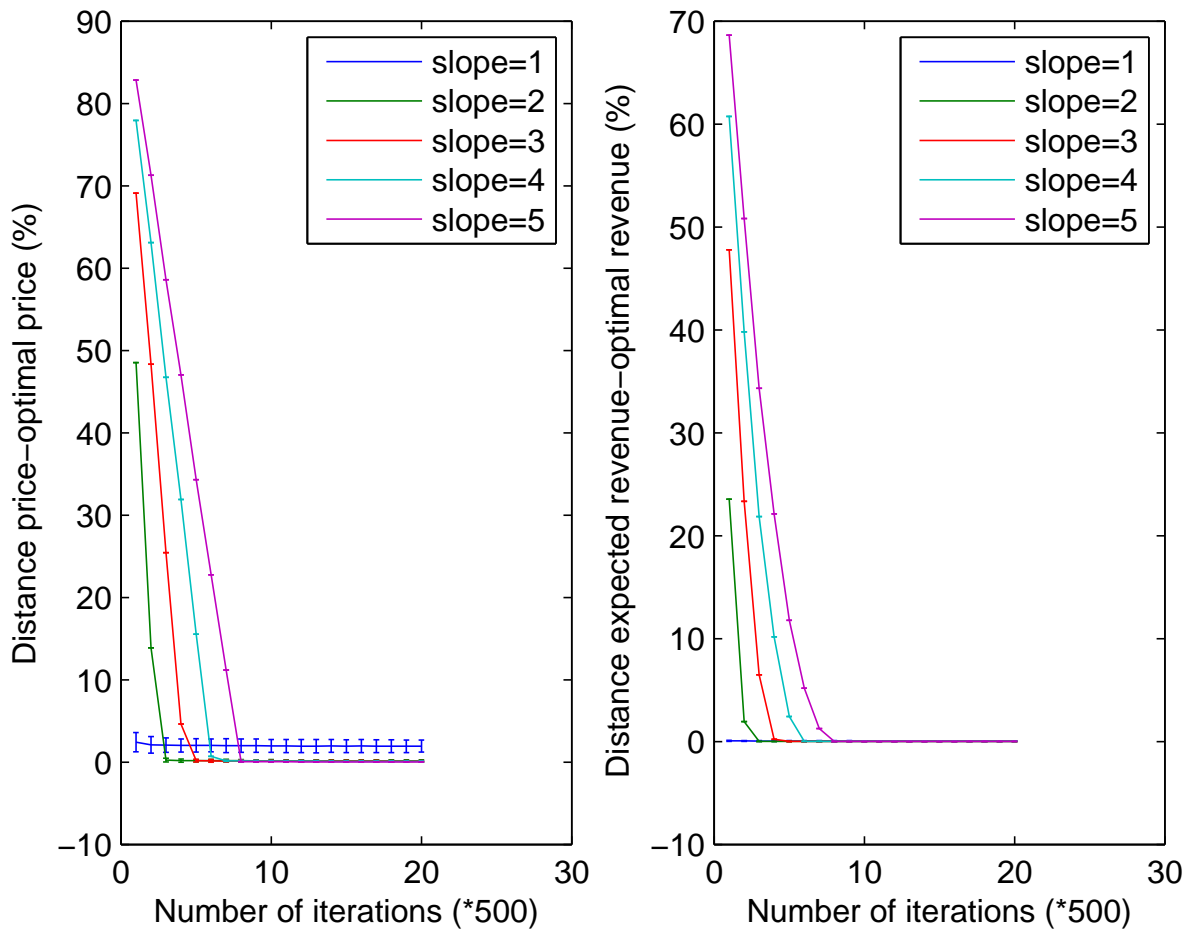


Figure 3-8: Sensitivity to the slope

	Std. dev	1	11	21	31	41
Mean(D p-o p) ^a (%)		0.020075	0.265787	0.269428	0.383859	1.015304
STD(D p-o p)(%)		0.018765	0.181128	0.230958	0.406946	0.441158
Mean(Dist exp r-o r) ^b (%)		7.2E-06	0.001002	0.001206	0.002964	0.01206
STD(Dist exp r-o. r) (%)		9.88E-06	0.001114	0.001832	0.003656	0.007301

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 3.5: Sensitivity to the standard deviation of the noise

3.2.3 Noise: Standard deviation and distribution

Finally for

$$d(p) = 300 - p + \varepsilon,$$

we change the standard deviation of the noise. The noise is Normal and we vary the standard deviation from 11, 21, 31, 41.

We see in Table 3.7 that the accuracy worsens when we increase the standard deviation.

Then, we try different distributions for the noise

- Normal distribution with standard deviation 1 and 10,
- Lognormal distribution with standard deviation 1 and 10,
- Pareto distribution with an infinite standard deviation.

Note that for all these distribution, we have to make sure also that the mean of the noise is zero by subtracting the mean. Moreover, the Pareto distribution has infinite variance, hence the results by Lai and Wei [22] do not apply. Hence, for this case we have no theoretical guarantee of convergence. Table 3.6 represents the results. We see that the accuracy is of the same level for the Normal and lognormal distribution when the variance is the same. Even for the Pareto distribution, the accuracy is satisfactory.

Distribution	Nor.(1)	Nor.(10)	Logn.(1)	Logn.(10)	Par.
Mn(P-o p)(%)^a	0.022305978	0.196098573	0.018599859	0.31834482	0.05562
STD(P-o p)(%)	0.016884704	0.122235608	0.014819237	0.403650006	0.031582
Mn(E r-o r)(%)^b	7.54141E-06	0.00051902	5.43604E-06	0.002479834	3.99E-05
STD(E r-o r) (%)	7.09338E-06	0.000528562	6.2115E-06	0.006018316	3.27E-05

^aMean(Distance price-optimal price)

^bMean(Distance expected revenue-optimal revenue)

Table 3.6: Sensitivity to the distribution of the noise

Capacity	30	78	126	174	222
Mean(Dist price-opt p)^a(%)	0.11131	0.095257	0.176728	0.072993	0.071025
STD(Dist price-opt p)(%)	0.053367	0.066147	0.185167	0.051513	0.059526
Mean(Dist E. rev.-o. rev)^b(%)	0.89038	0.175951	0.068148	7.72E-05	8.23E-05
STD(Dist E. rev.-o. rev) (%)	0.427038	0.122304	0.072215	9.39E-05	0.000134

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 3.7: Sensitivity to the capacity

3.2.4 Capacity

Finally, we vary the value of the capacity constraint and study the consequences on the accuracy of the price. As before, we consider the demand $d(p) = 300 - p + \varepsilon$. ε has mean zero and standard deviation 10. The capacity constraints we consider are

$$capacity = 30, 78, 126, 174, 222.$$

Note that the last two capacities are not tight when solving the optimization problem.

Table 3.7 represents the results. As expected, for the last two capacity constraints, the results are pretty similar. Overall, the accuracy is of the same level for the different capacities.

3.3 Insights

Despite the fact that in theory our algorithm needs precise bounds on the optimal price, we presented an heuristic with good performance in practice that gets rid of this requirement. The underlying idea is that this heuristic learns bounds on the localization of the optimal price during a transient phase. The same method can also be used under capacity constraints. We saw that the performance of the heuristic does not appear to depend on the initial price or the value of the capacity constraint. However, two critical factors are the length of the transient phase and the standard deviation of the demand noise.

Chapter 4

Multi-product case

In this chapter, we discuss how the approach we introduced in the single product case extends to the multi-product case. The multi-product case is much more realistic in practice, as a company typically sells several products. However, it is far more difficult to establish theoretical guarantees for an algorithm which jointly learns the demand parameters through least squares estimation and optimizes revenue. In particular, the conditions we need from [22] are much harder to establish. Hence, we will introduce a pricing strategy that uses the algorithm for the single product case as a subroutine in order to solve the multi-product case.

4.1 Demand model

4.1.1 Model

We consider a monopolistic setting where a firm offers m substitutable products. All the inequalities between vectors in the following assumptions will be considered componentwise. We impose the following assumptions:

Assumption 1: The demand function is an affine function with an additive noise.

We denote for each product i its price P_i and the corresponding demand D_i . Let \mathbf{P} and \mathbf{D} denote the m dimensional vector of price and demand respectively. Then the demand function is of the form:

$$\mathbf{D}(\mathbf{P}) = \mathbf{A} - \mathbf{B} \cdot \mathbf{P} + \boldsymbol{\varepsilon},$$

where

- \mathbf{A} is the vector of demand when all the prices are zero.
- $-\mathbf{B}$ is the symmetric matrix of demand sensitivities to price changes. For example $-B_{ij}$ denotes the demand change for product i as the result of an increase of one unit of the price for product j .
- $\boldsymbol{\varepsilon}$ is the m dimensional vector of noises of mean zero. We denote it by

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \dots \\ \varepsilon_m \end{pmatrix}.$$

Assumption 2: $\mathbf{P}_{\min} \leq \mathbf{P} \leq \mathbf{P}_{\max}$, where \mathbf{P}_{\min} and \mathbf{P}_{\max} are m dimensional vectors.

This is to make sure that we never price too low or too high.

Assumption 3: For all $\mathbf{P}_{\min} \leq \mathbf{P} \leq \mathbf{P}_{\max}$, $\mathbf{D}(\mathbf{P}) \geq 0$ almost surely.

This assumption guarantees that the demand never becomes negative. This condition is easy to check when the prices are bounded and the noise has finite support. However in practice we will be able to run the algorithm with bounded prices and unbounded noise. The only restriction will be in this case that the standard deviation should be small enough

to ensure a non negative demand with high probability.

Assumption 4: $\mathbf{diag}(\mathbf{B}) > \mathbf{0}$ and $\mathbf{offdiag}(\mathbf{B}) < \mathbf{0}$.

This is a common assumption in the literature to ensure that products are gross substitutes. If the price for a product increases and the other prices remain constant, then the demand for this product strictly decreases and the demand for its substitutes increases.

Assumption 5: \mathbf{B} is column strictly diagonally dominant ie $|B_{ii}| > \sum_{j \neq i} |B_{ji}|$. The degree of diagonal dominance of the sensitivity matrix is:

$$r = \max_{i=1..m} \frac{\sum_{j \neq i} |B_{ij}|}{|B_{ii}|}.$$

For strict diagonal dominance, $r < 1$.

Intuitively this condition means that if all the prices increase by one unit then the total demand decreases for all the products. In the market, when r is close to zero, then the total demand decreases by a lot with a price increase. When r is close to one, then the total demand decreases only slightly with a price increase.

4.1.2 Existence and uniqueness of the optimum

Theorem 4.1.1. *Under Assumptions 1-5, the expected profit is a strictly concave function.*

Proof. We write

$$\mathbb{E}(\Pi(\mathbf{P})) = \mathbf{P}^T \cdot (\mathbf{A} - \mathbf{B} \cdot \mathbf{P}).$$

The expected profit is a continuous twice differentiable function. The Hessian matrix is $H = -2 \cdot B$, thus it is a strictly diagonally dominant matrix. Using Gershgorin circle theorem [29], we can conclude that all the eigenvalues of the matrix $\Pi(\mathbf{P})$ are strictly positive. Hence the function is strictly concave. □

Theorem 4.1.2. *If we maximize the profit on a compact set, the maximum exists and is unique.*

Proof. The feasible region is a compact set and the objective function is a continuous function. Therefore there exists at least one maximum solution.

Because the expected profit is a strictly concave function, the maximum solution is unique. □

4.2 Uniform demand

4.2.1 Additional assumptions

Assumption 6: $\mathbf{B}=(1+x)\mathbf{I}-x\mathbf{H}$ where $\mathbf{I}=\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & & & \\ 0 & 0 & \dots & 1 \end{pmatrix}$

and $\mathbf{H}=\begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \dots & & & \\ 1 & 1 & \dots & 1 \end{pmatrix}$.

Assumption 7: $\mathbf{A}=A\cdot\begin{pmatrix} 1 \\ 1 \\ \dots \\ 1 \end{pmatrix}$.

Assumption 8: \mathbf{P}_{\min} and \mathbf{P}_{\max} are m dimensional constant vectors with all components equal to the constants P_{\min} and P_{\max} respectively.

These assumptions are will be important in our analysis in this subsection. Assumption 6 means that all the products have the demand sensitivities to their prices scaled without

loss of generality to -1 and demand sensitivities for the other products all equal to x . In other words, the demand for a product is sensitive to the average price of the other products. Moreover, all the demands when the prices are zero are equal. This model is called a uniform demand model [30].

Moreover, to satisfy the diagonal dominance condition with negative offdiagonals, we need:

$$0 \leq x < \frac{1}{m-1}.$$

Notice that: $r = (m-1) \cdot x$.

We consider as in the previous chapter the case where the parameters of the demand function are unknown and are estimated from the data. The goal, as in the previous chapter, is to optimize the profit and learn the parameters of the demand function.

4.2.2 Optimization

Throughout this section we will assume that the demand parameters are known. Our goal is to set prices in order to maximize the profit. We also assume that all the prices have to be within the lower bound P_{min} and the upper bound P_{max} .

Note that we assume that the upper bound P_{max} on the price guarantees that the demand will be nonnegative with high probability (see Assumption 3).

Then we want to solve the following problem:

$$\begin{aligned} & \text{maximize} && \mathbf{P}^T \cdot \mathbb{E}[\mathbf{D}(\mathbf{P})] \\ & \text{s.t.} && \mathbf{P}_{\min} \leq \mathbf{P} \leq \mathbf{P}_{\max} \end{aligned}$$

Since $\mathbb{E}[\mathbf{D}(\mathbf{P})] = \mathbf{A} - \mathbf{B} \cdot \mathbf{P}$, then this optimization formulation is equivalent to:

$$\begin{aligned} & \text{maximize} && \mathbf{P}^T \cdot (\mathbf{A} - \mathbf{B} \cdot \mathbf{P}) \\ & \text{s.t.} && \mathbf{P}_{\min} \leq \mathbf{P} \leq \mathbf{P}_{\max} \end{aligned}$$

Moreover, the \mathbf{B} matrix satisfies the uniform demand model assumptions (see Assumption 6). Then there exists an analytical expression for its inverse. This expression will be useful for the solution of the optimization problem.

Lemma 4.2.1. *If $\mathbf{B}=(1+x)\mathbf{I}-x\mathbf{H}$ then:*

$$\mathbf{B}^{-1} = \frac{1}{1+x} \left(\mathbf{I} + \frac{x}{1-r} \mathbf{H} \right).$$

Proof. We write

$$\mathbf{B}^{-1} = \frac{1}{1+x} \left(\mathbf{I} - \frac{x}{1+x} \mathbf{H} \right)^{-1}$$

The spectral radius of $\frac{x}{1+x} \mathbf{H}$ is less than 1 because $r = (m-1) \cdot x < 1$. Hence, we can express the inverse as an infinite sum:

$$\begin{aligned} \mathbf{B}^{-1} &= \frac{1}{1+x} \sum_{i=0}^{\infty} \left(\frac{x}{1+x} \right)^i \mathbf{H}^i \\ &= \frac{1}{1+x} \left(\mathbf{I} + \frac{x}{1+x} \sum_{i=1}^{\infty} \left(\frac{x}{1+x} \right)^i m^{i-1} \mathbf{H} \right) \quad (\text{because } H^i = m^i \cdot H) \\ &= \frac{1}{1+x} \left(\mathbf{I} + \frac{x}{1-r} \mathbf{H} \right). \end{aligned}$$

□

We then will use the following lemma:

Lemma 4.2.2. *The solution to the quadratic optimization formulation:*

$$\begin{aligned} &\text{maximize} \quad \mathbf{P}^T \cdot (\mathbf{A} - \mathbf{B} \cdot \mathbf{P}) \\ &\text{s.t.} \quad \mathbf{P}_{\min} \leq \mathbf{P} \leq \mathbf{P}_{\max} \end{aligned}$$

is

$$\mathbf{P} = \begin{cases} \frac{\mathbf{B}^{-1} \cdot \mathbf{A}}{2} & \text{if } \mathbf{P}_{\min} \leq \frac{\mathbf{B}^{-1} \cdot \mathbf{A}}{2} \leq \mathbf{P}_{\max} \\ \mathbf{P}_{\min} & \text{if } \frac{\mathbf{B}^{-1} \cdot \mathbf{A}}{2} \leq \mathbf{P}_{\min} \\ \mathbf{P}_{\max} & \text{if } \frac{\mathbf{B}^{-1} \cdot \mathbf{A}}{2} \geq \mathbf{P}_{\max} \end{cases}$$

Proof. This is a classical application of the KKT Conditions. Note that due to the expression of \mathbf{B}^{-1} and \mathbf{A} , $\frac{\mathbf{B}^{-1} \cdot \mathbf{A}}{2}$ as well as \mathbf{P}_{\min} are constant vectors. Hence the inequalities between vectors are well defined. □

This leads to the following lemma.

Lemma 4.2.3. *At the optimal solution, we must have*

$$P_1 = P_2 = \dots = P_m$$

Proof. It follows directly from the expression of \mathbf{P} in the previous lemma. □

4.2.3 Convergence algorithm

We consider the model described previously for a firm selling m products. We assume that the demand function is of the form

$$D(\mathbf{P}) = \mathbf{A} - \mathbf{B} \cdot \mathbf{P} + \boldsymbol{\varepsilon}, \text{ where}$$

$$\mathbf{B} = (1 + x)\mathbf{I} - x\mathbf{H}.$$

Due to the previous part, at every step all the prices are set to some value. This price will be noted denoted by P . Then the demand for product i will be

$$D_i(P) = A - (1 - x \cdot (m - 1)) \cdot P + \varepsilon_i.$$

The total profit will be:

$$\Pi(P) = m \cdot (A - (1 - x \cdot (m - 1)) \cdot P) + P \cdot \sum_{i=1}^m \varepsilon_i.$$

Notice that this formulation has a similar form to the single product case. Then at every step we are going to estimate the parameters and solve the following profit optimization problem:

$$\begin{aligned} & \text{maximize} && \Pi(P) = m.P(A - (1 - x.(m - 1)).P) + P.\sum_{i=1}^m \varepsilon_i \\ & \text{s. t.} && P_{min} \leq P \leq P_{max} \end{aligned}$$

We use the same approach as for the single product case. In this part, we will use the following notations:

- P_{min} is the lower bound for all the prices of the different products
- P_{max} is the upper bound for all the prices of the different products
- P^n is the price set by the firm for all the products at stage n

We use the same approach as in the single product case. Assuming that we have $D'(P) = m.(A - (1 - x.(m - 1)).P) + \sum_{i=1}^m \varepsilon_i$ ($D'(P)$ is the total demand and is such that $P \cdot D'(P)$ corresponds to the total profit). At every step n, we compute the estimated parameters A^n and x^n of the parameters A and x of the demand using linear regression between $D'(P)$ and P .

The basic idea of this algorithm is that we want to optimize the revenue over $[P_{min}, P_{max}]$ but that we know with a very good likelihood that the optimal price should lie between M and N with

$$P_{min} \leq M \leq N \leq P_{max}.$$

As a result, we will price most of the time between M and N and sometimes we will offer a discount which will make the price between P_{min} and M.

We want the price to be between M and N when no discount is offered. We also need to make sure to have a nonnegative demand and price. We use the following algorithm:

Algorithm:

1. Let x^n and A^n discussed above denote the parameters estimated through regression on the prices and corresponding observed demands thus far.

2. Set:

$$P^n = \max(\min(\frac{A^n}{2((1 - x^n).(m - 1))}, N), M),$$

if there exists no i such that $n = \lfloor 2^{\sqrt{i}} \rfloor$. Otherwise, set:

$$P^n = \max(\min(\frac{A^n}{2((1 - x^n).(m - 1))}, N), M) - \gamma.$$

3. Update the parameter estimates and repeat the procedure.

The theorem below follows:

Theorem 4.2.4. *We consider the Assumptions 1-7 and assume that at every step n we compute the estimated parameters A^n and x^n using linear regression. We pick the price at step m such as:*

$$P^n = \max(\min(\frac{A^n}{2(1 - x^n.(m - 1))}, N), M) + \gamma_n$$

where

$$\gamma_n = \begin{cases} -\gamma & \text{if there exists } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

where N , M and γ are such that $\gamma > 2.(N - M)$ and $M - \gamma > P_{\min}$.

We assume that the noise is such that:

$$\mathbb{E}(|\sum \varepsilon_i|^\alpha) \leq \infty \text{ a. s for some } \alpha > 2$$

Then the following holds:

1. The parameters A^n and x^n converge almost surely to the true parameters A and x .
2. The price converges in the Cesaro sense to the optimal price which means that the average price converges to the optimal price.

3. A subset $\mathfrak{A}(\mathbb{N}) = \{n \mid \text{there exists no } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor\}$ of the prices converges to the optimal price almost surely.

4. The average revenue converges also almost surely to the optimal revenue.

Proof. The proof is similar to the single product case. The only difference is the condition on the noise. □

Moreover we can simplify the condition on the noise:

Lemma 4.2.5. *If there exist $\alpha > 2$ integer such that for all i :*

$$\mathbb{E}(|\varepsilon_i|^\alpha) \leq \infty$$

then we have

$$\mathbb{E}(|\sum \varepsilon_i|^\alpha) \leq \infty$$

Proof. Notice that: $\mathbb{E}(|\sum \varepsilon_i|^\alpha) \leq \mathbb{E}((\sum |\varepsilon_i|)^\alpha)$

Then this sum is a sum a products which correspond to j -th orders , with $j \leq \alpha$, of ε_i which are known to be finite because the α -th order is finite. □

Hence the uniform demand case can be solved using similar techniques as the single product case. The results can be extended to a slightly more general demand function: the “semi-uniform” demand model. The demand form is the same as before with a difference on the form of vector \mathbf{A} and \mathbf{B} (Assumption 1-5 remain the same.) That is:

Assumption 6’:

$$\mathbf{B} = \begin{bmatrix} B_1 & -1 & -1 & \dots & -1 \\ -1 & B_2 & -1 & \dots & -1 \\ \dots & & & & \\ -1 & -1 & -1 & \dots & B_m \end{bmatrix}$$

Assumption 7':

$$\mathbf{A} = \begin{pmatrix} A_1 \\ A_2 \\ \dots \\ A_m \end{pmatrix}$$

The interpretation is that the demand is still sensitive to the average price of the other products but for different products, the intercept and the sensitivity to their own prices may be different. The proof technique is similar as before. It is based on the fact that for each product, we just have two demand parameters to learn and that these parameters do not depend on the prices of the other products. The proof is omitted for the sake of brevity.

In what follows, we will focus on the general demand case.

4.3 General symmetric demand case

We assume that Assumptions 1-5 still hold. We further relax Assumptions 6 and 7 on the form of the demand function. We assume that the demand has the general form:

$$\mathbf{D}(\mathbf{P}) = \mathbf{A} - \mathbf{B} \cdot \mathbf{P} + \varepsilon,$$

where

- \mathbf{P} is a vector of n prices.
- \mathbf{A} is the vector of demand when all the prices are zero.
- $-\mathbf{B}$ is a symmetric matrix of demand sensitivities to price changes. For all i , $B_{ii} > 0$ and, if the products are substitute, $B_{ij} < 0$ for all $j \neq i$. For example $-B_{ij}$ denotes the change in demand for product i as the result of an increase of one unit of the price

of product j . The symmetry of matrix \mathbf{B} is an assumption often used in the literature. Sing and Vives in [35] show that if a representative consumer maximizes a quadratic utility, then the corresponding demand function is symmetric.

- ε is the m dimensional vector of noises of mean zero.

Throughout this part we will use the following notation:

For a vector \mathbf{x} with m components x_i we define the following norm:

$$\|\mathbf{x}\| = \max_{i=1..m} |x_i|$$

It corresponds to the maximum norm. Note that all the norms are equivalent because the space dimension is finite.

For time period n , the price set for product i is denoted by P_i^n .

4.3.1 Subroutine focusing on one product

We will use throughout this part the following subroutine. For a given product, given that the prices for the other products are fixed, we wish to learn the price for this product that optimizes the total revenue for all products. In the next section, we will show that repeating this subroutine iteratively on every product returns a price vector that converges to the global optimal price.

Description of the problem

We give an expression for the price that maximizes the total profit given that the prices of the other products are fixed.

Theorem 4.3.1. *Consider a product i . Suppose that the prices for all products $j \neq i$ are fixed at value P_j . We know that the demand for any product i will then be:*

$$D_i(P_i, \mathbf{P}_{-i}) = A_i - \sum_{j \neq i} B_{ij} \cdot P_j - B_{ii} \cdot P_i + \varepsilon_i.$$

We assume that $\mathbb{E}(|\varepsilon_i|^\alpha) \leq \infty$ a. s for some $\alpha > 2$, and that there exists $(P_{min})_i$ and $(P_{max})_i$ such that for all n

$$(P_{min})_i \leq P_i^n \leq (P_{max})_i.$$

Then the price for product i that maximizes the total profit is:

$$P_i^* = \begin{cases} \frac{A_i - \sum_{j \neq i} 2B_{ij}P_j}{2B_{ii}} & \text{if } (P_{min})_i \leq \frac{A_i - \sum_{j \neq i} 2B_{ij}P_j}{2B_{ii}} \leq (P_{max})_i \\ (P_{min})_i & \text{if } \frac{A_i - \sum_{j \neq i} 2B_{ij}P_j}{2B_{ii}} \leq (P_{min})_i \\ (P_{max})_i & \text{if } \frac{A_i - \sum_{j \neq i} 2B_{ij}P_j}{2B_{ii}} \geq (P_{max})_i \end{cases}$$

Proof. The total profit is:

$$\Pi(P) = \sum_{i=1..m} P_i \cdot (A_i - B_{ii} \cdot P_i - \sum_{j \neq i} B_{ij} \cdot P_j).$$

Hence, if we assume that the constraints are not binding, by taking the partial derivative with respect to i , the optimality condition is

$$(A_i - \sum_{j \neq i} 2B_{ij}P_j) - 2B_{ii} \cdot P_i = 0.$$

Hence

$$P_i^* = \frac{A_i - \sum_{j \neq i} 2B_{ij}P_j}{2B_{ii}} \tag{4.1}$$

The result follows when the constraints are binding by using the KKT conditions. \square

We want to use our results on the single product case to solve this problem when the demand parameters are unknown. However, using the Algorithm of Chapter 2 will return a price that maximizes the revenue on product i , and not the total revenue as shown in the next part. Note that the numerator of (4.1) is equal to twice the intercept for the demand for product i when the other products are fixed minus the value A_i , which is unknown.

Optimization of the revenue for this product

For a product, given that the prices of the other products are fixed, if we only want to optimize the revenue for this product and not the total revenue, then the problem becomes equivalent to the single product case.

Note that if we focus on product i , given that $P_j, j \neq i$ are fixed, if the demand for i is

$$D_i(P_i, \mathbf{P}_{-i}) = A_i - \sum_{j \neq i} B_{ij} \cdot P_j - B_{ii} \cdot P_i + \varepsilon_i,$$

where $B_{ij} \leq 0$ and $B_{ii} > 0$, then the price that optimizes the profit on product i is:

$$\bar{P}_i = \frac{A_i - \sum_{j \neq i} B_{ij} P_j}{2B_{ii}}. \quad (4.2)$$

This is the direct analog to the single product case. Given that the demand parameters are unknown, the theorem below follows:

Theorem 4.3.2. *Consider a product i . If the prices for all products $j \neq i$, are fixed at the value P_j , then the demand for product i is*

$$D_i(P_i, \mathbf{P}_{-i}) = A_i - \sum_{j \neq i} B_{ij} \cdot P_j - B_{ii} \cdot P_i + \varepsilon_i.$$

We assume that $\mathbb{E}(|\varepsilon_i|^\alpha) \leq \infty$ a. s. for some $\alpha > 2$. There exists $(P_{min})_i$ and $(P_{max})_i$ such that for all n :

$$(P_{min})_i \leq P_i^n \leq (P_{max})_i.$$

Moreover, there exists M and N such that:

$$M \leq \bar{P}_i \leq N.$$

We assume that

$$M - 2(M - N) > (P_{min})_i.$$

We use the same algorithm as for the single product case to optimize the revenue for product i and learn its demand given the prices for the other products. We get at step n , P_i^n , $(A_i - \sum_{j \neq i} B_{ij} \cdot P_j)^n$, B_{ii}^n such that:

$$(A_i - \sum_{j \neq i} B_{ij} \cdot P_j)^n \rightarrow A_i - \sum_{j \neq i} B_{ij} \cdot P_j \text{ almost surely}$$

$$B_{ii}^n \rightarrow B_{ii} \text{ almost surely}$$

$$P_i^n \rightarrow \frac{A_i - \sum_{j \neq i} B_{ij} P_j}{2B_{ii}} \text{ almost surely for } n \text{ in}$$

$$\mathfrak{A}(\mathbb{N}) = (n | \text{there exists no } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor).$$

Proof. The algorithm is the same as for the single product case. Note that in practice, we can get rid of the bounds M and N as shown in the previous chapter. \square

The output of the subroutine is the price that maximizes the revenue for this particular product given fixed prices on the other products. Note that the optimal price of this problem does not maximize the actual total expected revenue but only the revenue on product i . However, the expressions for the optimal prices for the two cases above ((4.1) and (4.2)) are similar. The only difference is in the numerator, where we have a coefficient 1 in front of B_{ij} , $j \neq i$, in the sum and we need a coefficient 2 for the subroutine. This similarity enables us to use the above algorithm to solve the subroutine.

The subroutine algorithm

In this part, we connect the two cases above (4.1) and (4.2). To show how the subroutine can be solved, we will first assume that the demand intercept vector \mathbf{A} is known and then show how we can learn \mathbf{A} .

For now assume that the demand intercept vector \mathbf{A} is known. We use the following algorithm:

Algorithm:

1. For every product i , given that the prices for the products $j \neq i$ are set at P_j , let $(A_i - \sum_{j \neq i} B_{ij} \cdot P_j)^n$ and B_{ii}^n denote the parameters estimated through regression on the prices P_i and corresponding observed demands D_i thus far.
2. Set for every product i ,

$$P_i^n = \max(\min(\frac{2 \cdot (A_i - \sum_{j \neq i} B_{ij} P_j)^n - A_i}{2B_{ii}^n} - N, M), 0) + \gamma_n$$

where

$$\gamma_n = \begin{cases} -\gamma & \text{if there exists } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

3. Update the parameter estimates and repeat the procedure.

Then the following theorem applies:

Theorem 4.3.3. *Consider a product i . All the prices all the products $j \neq i$ are fixed at the value P_j . The demand for product i will then be:*

$$D_i(P_i, P_{-i}) = A_i - \sum_{j \neq i} B_{ij} \cdot P_j - B_{ii} \cdot P_i + \varepsilon_i,$$

with $\mathbb{E}(|\varepsilon_i|^\alpha) \leq \infty$ a. s. for some $\alpha > 2$. For every product i at iteration n , there exists $(P_{min})_i$ and $(P_{max})_i$ such that

$$(P_{min})_i \leq P_i^n \leq (P_{max})_i.$$

Moreover, there exists M and N such that:

$$M \leq P_i^* \leq N.$$

We assume that

$$M - 2(N - M) > (P_{min})_i.$$

At every step n the estimated demand intercept $(A_i - \sum_{j \neq i} B_{ij} \cdot P_j)^n$ and slope B_{ii}^n are computed using linear regression between D_i and P_i .

We price at the value:

$$P_i^n = \max(\min(\frac{2 \cdot (A_i - \sum_{j \neq i} B_{ij} P_j)^n - A_i}{2B_{ii}^n}, N), M) + \gamma_n$$

where

$$\gamma_n = \begin{cases} -\gamma & \text{if there exists } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

where N , M and γ are such that $\gamma > 2 \cdot (N - M)$ and $M - \gamma > (P_{\min})_i$.

Then the following holds:

$$2 \cdot (A_i - \sum_{j \neq i} B_{ij} \cdot P_j)^n - A_i \rightarrow A_i - \sum_{j \neq i} 2B_{ij} \cdot P_j \text{ almost surely}$$

$$B_{ii}^n \rightarrow B_{ii} \text{ almost surely}$$

$$P_i^n \rightarrow \frac{A_i - \sum_{j \neq i} 2B_{ij} P_j}{2B_{ii}} \text{ almost surely for } n \text{ in}$$

$$\mathfrak{A}(\mathbb{N}) = \{n \mid \text{there exists no } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor\}.$$

Proof. The proof is the same as for the single product case. The only difference is the value we set for the price. Notice that we use the true value of the parameter A_i for the price. \square

Although in theory, we need precise bounds N and M on the optimal solutions, in practice they are not required. At this point, we discussed how to compute the optimal price when the vector \mathbf{A} is known. But in a more general setting \mathbf{A} is unknown but can be learned. The learning phase for product i is summarized in Algorithm 1 below assuming without loss

of generality that

$$\mathbf{P}_{\max} \geq 1.5\mathbf{P}_{\min}.$$

Algorithm 1 Learning phase during N steps

Input: Number of steps N .

Output: Estimate A_i^N of A_i .

At every even step of the learning phase, the prices \mathbf{P}_{-i} are fixed at the value $(\mathbf{P}_{\min})_{-i}$.

At every odd step, the prices \mathbf{P}_{-i} are fixed at the value $1.5(\mathbf{P}_{\min})_{-i}$.

$n \leftarrow 1$

$Intercept_{even} = 0$

$Intercept_{odd} = 0$

repeat

if $n=2.l$ even **then**

$$P_i^{2l} = \begin{cases} (P_{min})_i & \text{if } l \text{ even} \\ (P_{max})_i & \text{otherwise} \end{cases}.$$

$Intercept_{even} = (A_i - \sum_{j \neq i} B_{ij}(P_{min})_j)^n$ and slope B_{ii}^n computed using linear regression between $(P_i^{2k}, k \leq l)$ and $(D_i^{2k}, k \leq l)$, i.e. by restricting the linear regression between price and demand to the even steps.

end if

if $n=2.l+1$ odd **then**

$$P_i^{2l+1} = \begin{cases} (P_{min})_i & \text{if } l \text{ even} \\ (P_{max})_i & \text{otherwise} \end{cases}.$$

$Intercept_{odd} = (A_i - \sum_{j \neq i} B_{ij} \cdot 1.5 \cdot (P_{min})_j)^n$ and slope B_{ii}^n computed using linear regression between $(P_i^{2k+1}, k \leq l)$ and $(D_i^{2k+1}, k \leq l)$, i.e. by restricting the linear regression between price and demand to the even steps.

end if

$A_i^n = 2 \cdot [1.5 \cdot Intercept_{even} - Intercept_{odd}]$

$n \leftarrow n + 1$

until $n = N$

The following theorem applies:

Theorem 4.3.4 (Learning phase). *We assume without loss of generality that:*

$$\mathbf{P}_{\max} \geq 1.5\mathbf{P}_{\min}.$$

Consider a product i . At every even step of the subroutine, the prices \mathbf{P}_{-i} are fixed at the value $(\mathbf{P}_{\min})_{-i}$. At every odd step, the prices \mathbf{P}_{-i} are fixed at the value $1.5(\mathbf{P}_{\min})_{-i}$.

There exists $(P_{\min})_i$ and $(P_{\max})_i$ such that for all n , we want:

$$(P_{\min})_i \leq P_i^n \leq (P_{\max})_i.$$

At every even step $2n$ the estimated demand intercept for product i , $(A_i - \sum_{j \neq i} B_{ij}(P_{\min})_j)^{2n}$ and slope $B_{ii}^{2:n}$ are computed using linear regression between $(P_i^{2k}, k \leq n)$ and $(D_i^{2k}, k \leq n)$, i.e. by restricting the linear regression between price and demand to the even steps.

Moreover, we price at the value:

$$P_i^{2n} = \begin{cases} (P_{\min})_i & \text{if } n \text{ even} \\ (P_{\max})_i & \text{otherwise} \end{cases},$$

At every odd step $2n+1$, the estimated demand intercept for product i , $(A_i - \sum_{j \neq i} B_{ij} \cdot 1.5 \cdot (P_{\min})_j)^{2n+1}$ and slope B_{ii}^{2n+1} are computed using linear regression between $(P_i^{2k+1}, k \leq n)$ and $(D_i^{2k+1}, k \leq n)$, i.e. by restricting the linear regression between price and demand to the odd steps.

Moreover, we price at the value:

$$P_i^{2n+1} = \begin{cases} (P_{\min})_i & \text{if } n \text{ even} \\ (P_{\max})_i & \text{otherwise} \end{cases},$$

Then:

$$2. \left[1.5 \cdot (A_i - \sum_{j \neq i} B_{ij}(P_{\min})_j)^{2n} - (A_i - \sum_{j \neq i} 1.5 \cdot B_{ij} \cdot (P_{\min})_j)^{2n+1} \right] \text{ converges a.s. to } A_i.$$

Moreover, the converge rate is $\frac{\ln(n)^{0.5}}{n^{0.5}}$ which means

$$|A_i^n - A_i| = O\left(\frac{\ln(n)^{0.5}}{n^{0.5}}\right) \text{ a.s.}$$

Proof. We know that the demand estimates will converge to the true value almost surely since the condition by Lai and Wei [22] holds. The convergence rate is

$$O\left(\left[\frac{\ln(n)}{n \cdot s_n^2}\right]^{0.5}\right) = O\left(\frac{\ln(n)^{0.5}}{n^{0.5}}\right)$$

□

Note that this rate of convergence is much faster than in Chapter 2 because the prices we set for product i are independent and just take two values. In conclusion, the previous algorithm can be used also for an unknown demand intercept vector \mathbf{A} after a preliminary step is applied. Hence, we have a way of solving the subroutine for the general case.

4.3.2 Tatonnement algorithm

In this section, we introduce an algorithm for optimizing the total profit and learning the demand for the total product. The difficulty arises from the fact that for the multi-product case, the consistency conditions by Lai and Wei [22] are difficult to check directly. Hence, we use the following idea using the results we have established for the subroutine. At each step, the price for each product is updated assuming that the other products have the same price as in the last step. To implement this strategy, we will use the subroutine we discussed in the previous subsection. For a product i at iteration n , given that the prices for the other products at iteration $n-1$ were $P_{-i}^{(n-1)}$, we denote by $P_i^{(n)} = SR(\mathbf{P}_{-i}^{(n-1)})$ the price which optimizes the total profit given that $P_j, j \neq i$ are fixed.

Learning the optimal prices will be achieved when the vector of prices between iteration n and iteration $n+1$ do not differ by more than a small constant $\eta > 0$ so we stop when

$$\|\mathbf{P}^{n+1} - \mathbf{P}^n\| \leq \eta$$

The tatonnement algorithm is summarized in Algorithm 2 below.

Algorithm 2 Tatonnement algorithm

Input: Set of initial values for the prices $P_j^0, j = 1, \dots, m$.

Output: Set of prices.

$n = 0$

for $i=1.. m$ **do**

Initialize $P_i \leftarrow P_i^0$.

A_i given or computed using the learning algorithm 4.3.4.

end for

repeat

for $i=1..m$ **do**

$P_i^{(n)} = SR(P_{-i}^{(n-1)})$

Set $P_i^{(n)} \leftarrow SR(P_{-i}^{(n-1)})$

end for

$n \leftarrow n+1$

until $\|P^{n+1} - P^n\| \leq \eta$

We use the subroutine for product i at step n with the prices for the other products $j \neq i$ fixed at $P_j^{(n-1)}$. This will return a price P_i^n , as well as estimates of $A_i - \sum_{j \neq i} 2B_{ij}P_j^{n-1}$ and of B_{ii} . The price as well as the estimates converge almost surely to the optimal price and true parameters respectively. Note that even if we have an estimate of $\left[A_i - \sum_{j \neq i} B_{ij}P_j^{n-1} \right]$ converging to the true sum we still do not have an estimate of the individual parameters A_i and B_{ij} .

Theorem 4.3.5. *Assume that Assumptions 1-5 hold. If the initial price is fixed for every product i at the value P_i^0 . At every step n , we use $P_i^{(n)} = SR(P_{-i}^{(n-1)})$.*

Moreover we assume that

$$\mathbb{E}(\|\epsilon\|^\alpha) \leq \infty \text{ a. s for some } \alpha > 2.$$

Then:

$$P^n \rightarrow P^* \text{ almost surely}$$

$$\mathbb{E}(\Pi^n) \rightarrow \Pi^*.$$

The convergence is geometric i.e., $\|\mathbf{P}^{n+1} - \mathbf{P}^*\| \leq a \cdot \|\mathbf{P}^n - \mathbf{P}^*\|$, with $a < 1$ (assuming that the subroutine described in the previous subsection is a black box). Recall that the norm is the max norm.

Proof. We compute $P_i^n = SR(P_i^{n-1})$. That is:

$$P_i^n = \frac{A_i - \sum_{j \neq i} 2B_{ij}P_j^{n-1}}{2B_{ii}}$$

At the optimal solution, we have that:

$$P_i^* = \frac{A_i - \sum_{j \neq i} 2B_{ij}P_j^*}{2B_{ii}}.$$

Then we have

$$P_i^{n+1} - P_i^* = \frac{\sum_{j \neq i} -2B_{ij}(P_j^n - P_j^*)}{2B_{ii}}, \text{ which implies that}$$

$$|P_i^{n+1} - P_i^*| \leq \frac{\sum_{j \neq i} 2|B_{ij}| \|\mathbf{P}^n - \mathbf{P}^*\|}{2B_{ii}}$$

(using definition of the max norm.)

It follows that:

$$|P_i^{n+1} - P_i^*| \leq r \cdot \|\mathbf{P}^n - \mathbf{P}^*\| \text{ (where } r = \max_{i=1..m} \frac{\sum_{j \neq i} |B_{ij}|}{B_{ii}} \text{.)}$$

Then we are able to conclude:

$$\begin{aligned} \|\mathbf{P}^{n+1} - \mathbf{P}^*\| &= \max_{1 \leq i \leq m} |P_i^{n+1} - P_i^*| \\ &\leq r \cdot \|\mathbf{P}^n - \mathbf{P}^*\|. \end{aligned}$$

It follows that

$$\|\mathbf{P}^n - \mathbf{P}^*\| \leq r^n \cdot \|\mathbf{P}^0 - \mathbf{P}^*\|.$$

As a result, the vector of prices converges to a vector of prices P^* and the convergence is geometric.

We can also show in a similar way that

$$\|\mathbf{P}^{n+1} - \mathbf{P}^n\| \leq r \cdot \|\mathbf{P}^n - \mathbf{P}^{n-1}\|$$

This will be useful for the computations since the optimal solution \mathbf{P}^* is unknown.

Moreover the expected profit also converges almost surely. We have that:

$$\begin{aligned} \mathbb{E}(\Pi(P)) &= \sum_{1 \leq i \leq m} P_i \cdot D_i(P_i, P_{-i}) \\ &= \sum_{1 \leq i \leq m} (P_i \cdot ((A_i - B_i \cdot P_i - B_{-i} \cdot P_{-i})). \end{aligned}$$

Since P_i^n and P_{-i}^n converge to the optimal solution, the expected profit also converges to the optimal solution.

□

Note that an important aspect of this algorithm is that if enough subroutines are run, the true demand parameters can be learned. For example, for m products, we need to run $m - 1$ subroutines for each product. Then for product i , A_i will be known after the learning phase, and at step n , B_{ii} as well as $\sum_{j \neq i} B_{ij} \cdot P_j^{n-1}$ are learned. Hence, if the price vectors are linearly independent, the $m - 1$ unknown parameters B_{ij} , $j \neq i$ can be learned by solving a system of equations after $m - 1$ subroutines for each product.

Moreover, the convergence rate has the following interesting interpretation. Recall that:

$$r = \max_{i=1..m} \frac{\sum_{j \neq i} |B_{ij}|}{B_{ii}}.$$

When $r=0$, hence for all $i \neq j$ $B_{ij} = 0$. Then the convergence is very fast, i.e. we apply the subroutine once for each product. This makes sense because in this case the prices of the products do not affect each other demand. (i.e. the products are independent). Then for each product, the subroutine solves the problem directly.

In the worst case scenario, r will be close to 1 and hence the convergence will be slow.

We test all this numerically in Chapter 5. Notice that the proof holds for any strictly diagonally dominant matrix, hence we are not restricted to the substitutable products case.

4.3.3 Algorithm in practice

The previous algorithm assumed that the problem was solved exactly, which means that the subroutine is run an infinite number of times. In practice, we will show that this algorithm also works if we run the subroutine only within an accuracy ε to the optimal solution.

Learning the optimal price will be achieved when the vector of best response prices between step n and step $n+1$ do not differ by more than a small constant $\eta > 0$. That is, when

$$\|\mathbf{P}^{n+1} - \mathbf{P}^n\| \leq \eta.$$

Recall that:

$$r = \max_{i=1..m} \frac{\sum_{j \neq i} |B_{ij}|}{B_{ii}}$$

Let us assume without loss of generality that demand vector $\mathbf{A} = \mathbf{D}(\mathbf{0})$ is known.

Let: $T = \|\mathbf{P}_{\max} - \mathbf{P}_{\min}\|$ and $H = \lfloor \frac{\ln(\frac{\eta}{T})}{\ln(\frac{1+r}{2})} \rfloor + 2$. This choice will be justified later.

For every $n \leq H$, we define:

$$\psi^n = \frac{1-r}{4} \cdot T \cdot \left(\frac{1+r}{2}\right)^{n-2},$$

to be the precision with which the subroutine is run at step n . We show that the algorithm sketched below still converges.

Algorithm: For product i at step n , let $P_i^n = SR(P_{-i}^{n-1})$. We run $SR(p_i^{n-1})$ as many iterations required to be within ψ^n of the optimum. We call SR' this approximate subroutine price and we note $P_i^n = SR'(P_{-i}^{n-1})$.

Algorithm 3 Tatonnement algorithm in practice

Input: Set of initial values for the prices during 2 consecutive iterations \mathbf{P}^0 and \mathbf{P}^1 .

Output: Set of prices.

$n \leftarrow 2$

repeat

for $i=1..m$ **do**

 Compute $SR'(P_{-i}^{(n-1)})$

 Set $P_i^{(n)} \leftarrow SR'(P_{-i}^{(n-1)})$

end for

$n \leftarrow n+1$

until $\|\mathbf{P}^{n+1} - \mathbf{P}^n\| \leq \eta$

Theorem 4.3.6. *We assume that Assumptions 1-5 hold. Then the price vector converges almost surely to the optimal price. Furthermore, the expected profit converges to the optimal expected profit using the algorithm described above.*

Proof. For product i at iteration n :

$$\frac{A_i - \sum_{j \neq i} B_{ij} P_j^{n-1}}{2B_{ii}} - \psi^n \leq P_i^n \leq \frac{A_i - \sum_{j \neq i} B_{ij} P_j^{n-1}}{2B_{ii}} + \psi^n$$

$$\frac{A_i - \sum_{j \neq i} B_{ij} P_j^{n-2}}{2B_{ii}} - \psi^{n-1} \leq P_i^{n-1} \leq \frac{A_i - \sum_{j \neq i} B_{ij} P_j^{n-2}}{2B_{ii}} + \psi^{n-1}$$

where $\psi^{n-1} \geq \psi^n$.

Then it is easy to show that for $n \geq 3$:

$$\begin{aligned} \|\mathbf{P}^n - \mathbf{P}^{n-1}\| &\leq r \cdot \|\mathbf{P}^{n-1} - \mathbf{P}^{n-2}\| + 2 \cdot \psi^{n-1} \\ &\leq r \cdot \|\mathbf{P}^{n-1} - \mathbf{P}^{n-2}\| + \frac{1-r}{2} \cdot T \cdot \left(\frac{1+r}{2}\right)^{n-3}. \end{aligned}$$

Note that $\|\mathbf{P}^1 - \mathbf{P}^0\| \leq T$ and $\|\mathbf{P}^2 - \mathbf{P}^1\| \leq T$. Therefore:

$$\begin{aligned} \|\mathbf{P}^3 - \mathbf{P}^2\| &\leq r \cdot \|\mathbf{P}^2 - \mathbf{P}^1\| + \frac{T \cdot (1-r)}{2} \\ &\leq r \cdot T + \frac{T \cdot (1-r)}{2} \\ &\leq \frac{1+r}{2} T. \end{aligned}$$

Hence using an easy recursion, it follows that

$$\|\mathbf{P}^{n+1} - \mathbf{P}^n\| \leq \left(\frac{1+r}{2}\right)^{n-1} T.$$

This implies that \mathbf{P} converges almost surely.

We stop when

$$\|\mathbf{P}^{n+1} - \mathbf{P}^n\| \leq \eta.$$

This holds for

$$n \geq \frac{\ln \frac{\eta}{T}}{\ln \frac{1+r}{2}} + 1.$$

steps. Hence the choice of $H = \lfloor \frac{\ln(\frac{\eta}{T})}{\ln(\frac{1+r}{2})} \rfloor + 2$ is justified.

As before, we can show that the expected profit converges almost surely to the optimal profit. \square

In practice, we will not be able to run the subroutine and have a guarantee on the number of iterations necessary to stop within any desired accuracy of the optimum. However, the above computations suggest that, even when the subroutine is stopped after a finite number of iterations, the general algorithm is still robust.

4.4 Conclusions

In this chapter, we extended the learning and optimization approach from the single product case to the uniform multi-product case as well as to the general multi-product case as long as demand is an affine function of the price. Because we cannot find consistency conditions for the demand parameters when $m > 1$, we introduced an algorithm that uses the single product case algorithm, where consistency conditions are easier to check, as a subroutine. In the next chapter, we will test the algorithm in practice by running simulations.

Chapter 5

Simulations: Affine demand, multi-product case.

5.1 General algorithm

We now run simulations for the multi-product case, using the single product case as a subroutine as described in the previous Chapters. For the subroutine, we do not assume that precise bounds on the optimal prices are known. Hence, we will use the “improved version” of the single product case algorithm (see Chapter 3).

We consider as a base case a two product case setting. The demand functions are of the following form:

$$D_1(P_1, P_2) = 200 - P_1 + 0.5P_2 + \varepsilon_1,$$

$$D_2(P_1, P_2) = 150 + 0.5P_1 - P_2 + \varepsilon_2.$$

Hence, in the matrix form:

$$\mathbf{D}(\mathbf{P}) = \mathbf{A} - \mathbf{B} \cdot \mathbf{P} + \boldsymbol{\epsilon}, \text{ with } \mathbf{A} = \begin{bmatrix} 200 \\ 150 \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$

Notice that Assumptions 1-5 are satisfied. For Assumption 2, we use $\mathbf{P}_{\min} = [100; 100]$ and $\mathbf{P}_{\max} = [250; 250]$. The matrix is diagonal dominant with $r = 0.5$. The optimal price is vector $\mathbf{P}^* = \frac{\mathbf{B}^{-1}\mathbf{A}}{2}$. Hence $P_1^* = 183.33$ and $P_2^* = 166.67$.

The optimal expected profit is $\Pi(P_1, P_2) = 30,833$.

In the simulations, we assume that the vector \mathbf{A} is known and that ε_1 and ε_2 follow a Normal distribution with mean zero and standard deviation 10. We run 10 subroutines and every subroutine is stopped after 1,000 iterations. Figure 5-1 shows the typical output of the algorithm. Note that because vector \mathbf{A} is known, each subroutine will give us an estimated value of the demand sensitivity matrix \mathbf{B} . Hence after having run k subroutines, we will have a precise estimate of the demand sensitivity matrix \mathbf{B} by taking the average of the estimated the demand sensitivity matrix \mathbf{B} returned after each subroutine.

We can see on Figure 5-1 above phases where one the prices is constant. This is due to the nature of the general algorithm, which set the prices of all the products but one at a constant value. Moreover, when for example price P_1 is fixed, price P_2 converges similarly as in the single product case. This is due to the nature of the subroutine algorithm.

We repeat the procedure 10 times and give in Table 5.1 the mean and standard deviation of the estimated demand matrix \mathbf{B} , unperturbed prices and expected revenues. Note that the standard deviation of the output is smaller than in the single product case. This might be due to the robustness of the algorithm to the errors in the subroutine, as explained in Chapter 4.

5.2 Sensitivity analysis: Learning phase.

We presented in Chapter 4 a way of learning the demand intercept vector \mathbf{A} when it is unknown. We test the performance of the algorithm for a demand function of the following

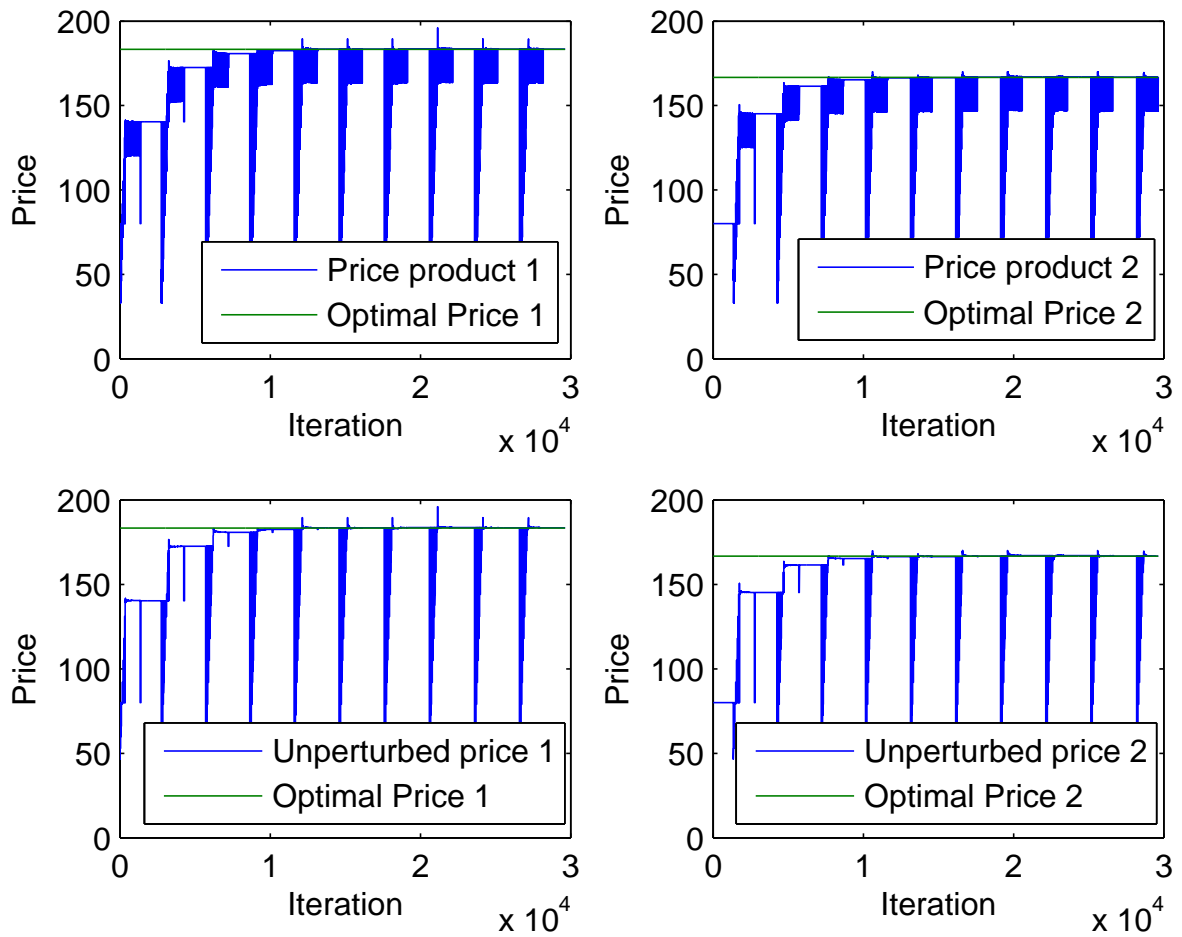


Figure 5-1: Output for the multi product case

Num. subroutine calls	2	4	6	8	10
mean(B11 estimated)	0.996428751	1.001099	1.002126	1.000075	0.998847
STD(B11 estimated)	0.023598062	0.018683	0.00983	0.009711	0.008456
mean(B12 estimated)	-0.49482208	-0.49987	-0.50103	-0.49943	-0.49842
STD(B12 estimated)	0.025574815	0.019415	0.009903	0.009453	0.007988
mean(B21 estimated)	-0.50547756	-0.50279	-0.5014	-0.50392	-0.50497
STD(B21 estimated)	0.022959761	0.013083	0.01304	0.008968	0.00956
mean(B22 estimated)	1.005155486	1.002559	1.001222	1.003574	1.004585
STD(B22 estimated)	0.018076563	0.011004	0.011668	0.008444	0.008619
mean(P1)	140.1180027	172.588	181.3565	183.5912	183.979
STD(P1)	0.433829631	0.449894	1.4283	0.826416	0.076365
mean(P2)	145.1885063	161.2221	165.9413	167.6073	167.4083
STD(P2)	0.529393976	0.491372	1.089493	0.550397	0.526768
mean(Exp. revenue)	29432.32773	30746.44	30828.8	30831.77	30832.6
STD(Exp. revenue)	28.18179636	7.135757	2.089378	1.624393	0.517264

Table 5.1: Output of the algorithm

form:

$$D_1(P_1, P_2) = 100 - P_1 + 0.5P_2 + \varepsilon_1,$$

$$D_2(P_1, P_2) = 100 + 0.5P_1 - P_2 + \varepsilon_2.$$

where ε_1 and ε_2 both follow the Normal distribution with mean zero and standard deviation 10.

We have $\mathbf{A} = \begin{bmatrix} 100 \\ 100 \end{bmatrix}$.

Table 5.2 represents the mean and standard deviation for the demand intercept at different steps of the algorithm over 10 runs. We see that the accuracy is satisfactory and moreover, the standard deviation of the error decreases rapidly. This is probably due to the good convergence rate.

In the next part, we will consider that the vector \mathbf{A} is known in order to measure the performance of the tatonnement algorithm.

Step	2000	4000	6000	8000	10000
mean(A1)	100.1538346	100.2332	100.4497	99.41243	99.79282
STD(A1)	1.6191181	1.4591	1.027497	1.205294	0.612911
mean(A2)	99.25372496	99.8816	99.69817	99.57608	99.59459
STD(A2)	1.97974574	1.19969	0.969969	0.822882	0.466183

Table 5.2: Learning phase

5.3 Sensitivity analysis: Tatonnement algorithm.

As for the single product case, we test the sensitivity of our algorithm to different parameters: standard deviation of the noise, degree of diagonal dominance of the matrix as well as number of products. In all these simulations, we want to measure the accuracy of the tatonnement algorithm and not of the learning phase. Hence, we will assume that the demand intercept vector \mathbf{A} is known a priori.

5.3.1 Standard deviation of the noise

We consider the two products case. The demand functions are of the following form:

$$D_1(P_1, P_2) = 100 - P_1 + 0.5P_2 + \varepsilon_1$$

$$D_2(P_1, P_2) = 100 + 0.5P_1 - P_2 + \varepsilon_2$$

ε_1 and ε_2 both follow the same normal distribution with mean zero and some standard deviation σ . The optimal price vector is $\mathbf{P}^* = [100, 100]$ and the optimal expected profit is $\mathbb{E}\Pi(\mathbf{P}^*) = 10,000$. We will consider the values:

$$\sigma = 1, 11, 21, 31, 41.$$

For every value of the standard deviation, we run the algorithm stopping after 10 runs of the subroutine for every product, which means 20 runs in total. Each time, the subroutine is run 1000 steps. We repeat the procedure 10 times and plot on Figure 5-2 the average and standard deviation of the price vector returned after each run of the subroutine.

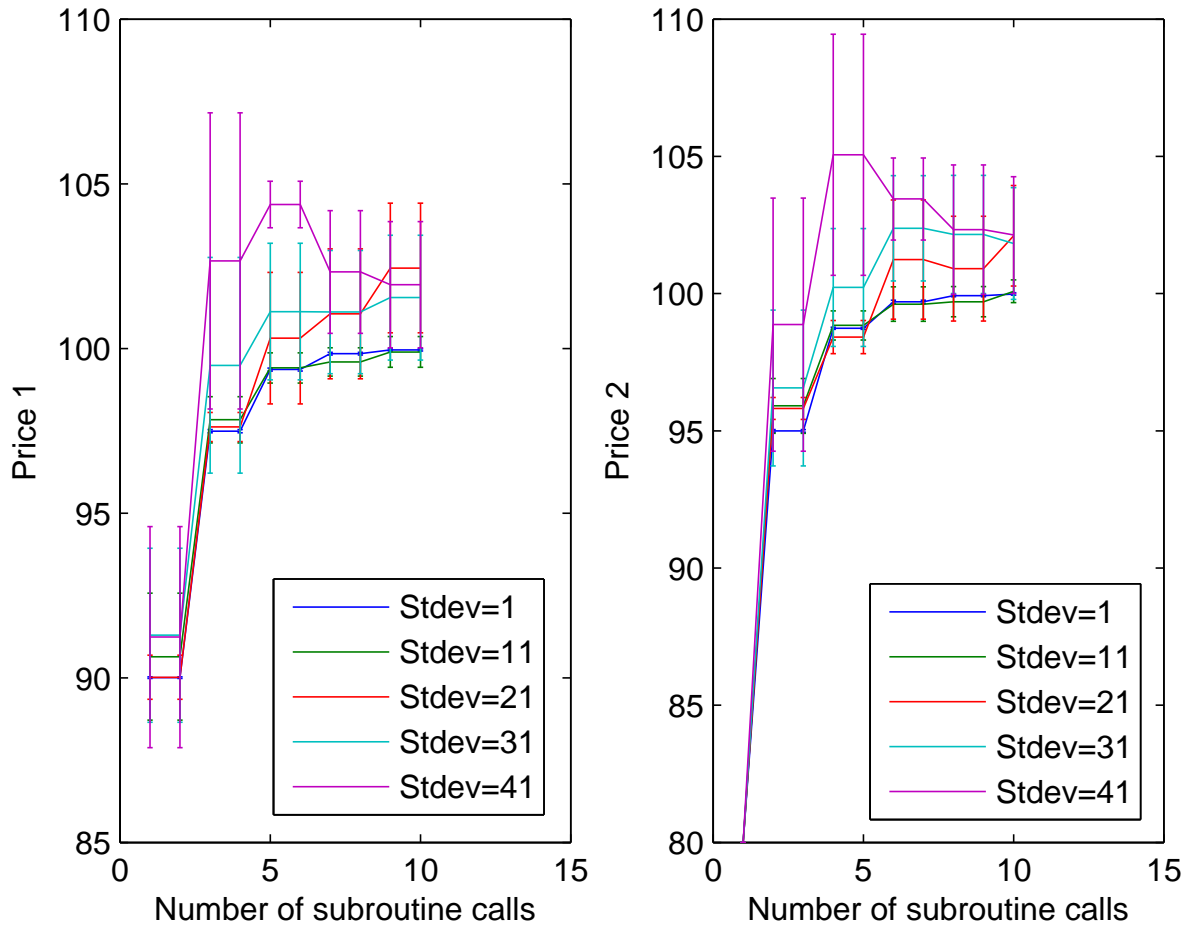


Figure 5-2: Sensitivity to the noise standard deviation

Standard deviation	1	11	21	31	41
mean(P1 final)	99.95867561	99.89396	102.4405	101.5441	101.9376
STD(P1 final)	0.031913106	0.468529	1.967849	1.894596	1.914215
mean(P2 final)	99.98650065	100.0839	102.1074	101.8222	102.1332
STD(P2 final)	0.040196179	0.413365	1.831658	2.036612	2.127919
mean(Exp. revenue)	9999.997172	9999.708	9991.152	9992.412	9989.903
STD(Exp. revenue)	0.002049619	0.290352	7.241987	7.157818	5.319322

Table 5.3: Sensitivity to the standard deviation of the noise

Hence, when the variance gets significantly bigger, the precision worsens significantly. However, we see in Table 5.3 that when the standard deviation of the noise increases significantly, the error on the price stays pretty limited.

5.3.2 Number of iterations of the subroutine

Consider the same case as above with demand functions of the following form:

$$D_1(P_1, P_2) = 100 - P_1 + 0.5P_2 + \varepsilon_1,$$

$$D_2(P_1, P_2) = 100 + 0.5P_1 - P_2 + \varepsilon_2,$$

where ε_1 and ε_2 both follow the Normal distribution with mean zero and a fixed standard deviation 10.

We vary the number of iterations n we run the subroutine. We consider

$$n = 1000, 3000, 5000, 7000.$$

As before, we run 10 subroutines and measure the accuracy of the mean and standard deviation of the final solution over 10 runs on Table 5.4.

There does not seem to be a significant increase in the accuracy when we increase the number of iterations for each subroutine. This is probably due to the bad convergence rate of the subroutine.

Steps Subroutine	1000	3000	5000	7000
mean(P1 final)	100.3849037	101.0671	101.2989	100.5613
STD(P1 final)	0.895537321	1.637912	1.292169	0.855578
mean(P2 final)	100.2297109	100.6475	100.9752	100.2386
STD(P2 final)	0.581486868	1.347666	1.208428	0.429011
mean(Exp. revenue)	9999.2752	9996.899	9996.656	9999.248
STD(Exp. revenue)	1.261703654	5.857363	4.730333	1.341156

Table 5.4: Sensitivity to the number of iterations of the subroutine

5.3.3 Degree of diagonal dominance

We showed in Chapter 4 that the degree of diagonal dominance of the matrix r plays a significant role. We will now check it in practice.

We consider the two product case. The demand functions are of the following form:

$$D_1(P_1, P_2) = 100 - P_1 + r.P_2 + \varepsilon_1$$

$$D_2(P_1, P_2) = 100 + r.P_1 - P_2 + \varepsilon_2$$

ε_1 and ε_2 both follow the Normal distribution with mean zero and a fixed standard deviation $\sigma = 10$.

r is the degree of diagonal dominance of the matrix. We consider values:

$$r = 0.2, 0.4, 0.6, 0.8, 0.98$$

For every value of the standard deviation, we run the algorithm stopping after 10 runs of the subroutine for every product, which means 20 runs in total. Each time, the subroutine is run 1000 steps.

We define the relative distance between price vector \mathbf{P} and the optimal price P^* as $\frac{\|P - P^*\|}{\|P^*\|} \cdot 100$. Note that the norm is the max norm.

In a similar way, the distance between the expected revenue $\mathbb{E}\Pi(\mathbf{P})$ and the optimal expected revenue $\mathbb{E}\Pi(\mathbf{P}^*)$ is : $\frac{|\mathbb{E}\Pi P - \mathbb{E}\Pi P^*|}{|\mathbb{E}\Pi P^*|} \cdot 100$. We repeat the procedure 10 times and plot the

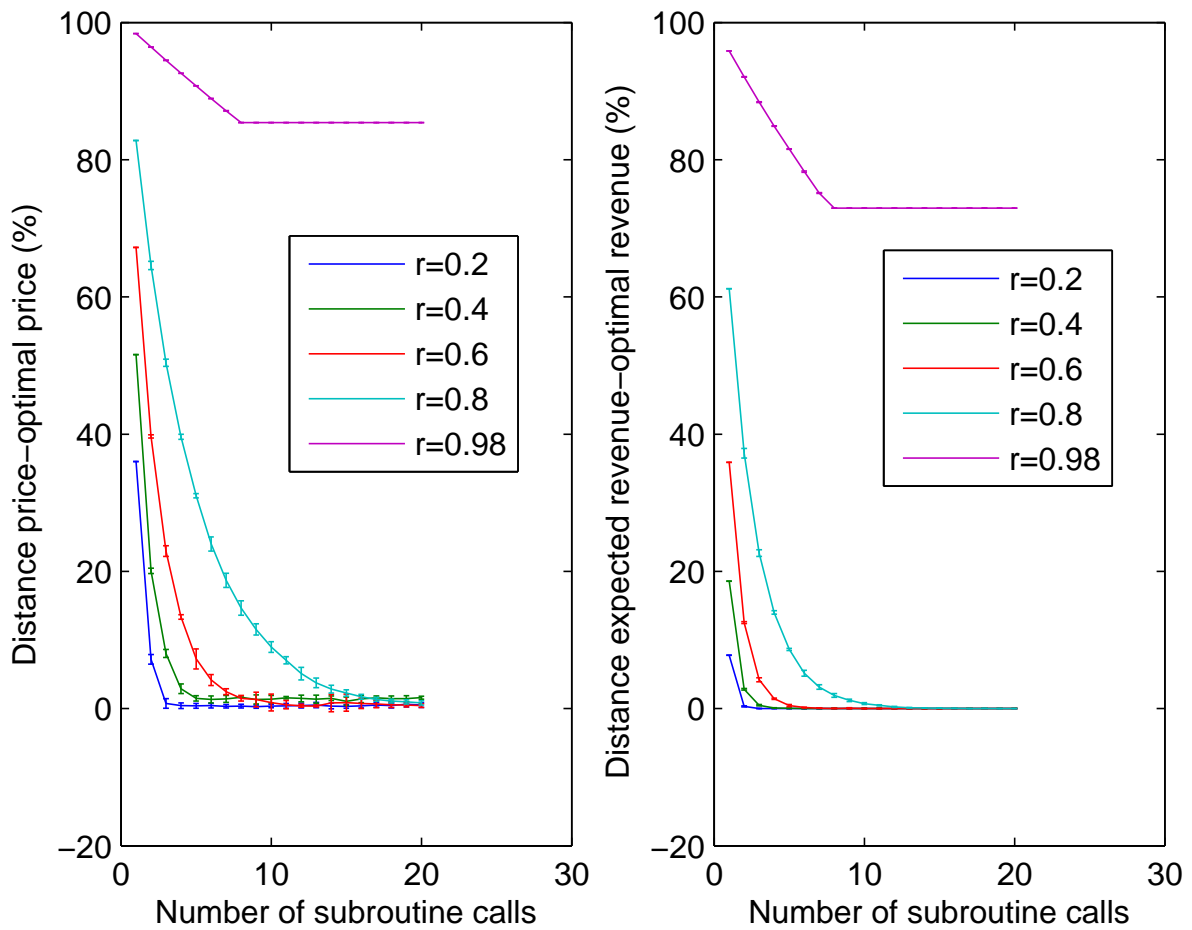


Figure 5-3: Sensitivity to the degree of diagonal dominance

average and standard deviation of the relative distance of the price vector to the optimal price returned after each run of the subroutine.

We see that the convergence is geometric, as expected. Moreover, when r gets bigger, the convergence rate of the algorithms worsens significantly. This is particularly visible for r bigger than 0.95.

The next tables show the difference in the output for different values of r .

Number of subroutine calls	4	8	12	16	20
mean(Dist p-opt p) ^a (%)	0.425017476	0.370406	0.44385	0.413307	0.517072
STD(Dist p-opt p)(%)	0.405485612	0.277166	0.350496	0.341577	0.359642
mean(Dist e r-opt rev) ^b (%)	0.002238746	0.001369	0.002087	0.002014	0.003083
STD(Dist e r-opt rev)(%)	0.00370424	0.001698	0.002797	0.002707	0.003685

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 5.5: Output for r=0.2

Number of subroutine calls	4	8	12	16	20
mean(Dist p-opt p) ^a (%)	13.35177517	1.498013	0.408528	0.758367	0.445403
STD(Dist p-opt p)(%)	0.343555283	0.412995	0.20385	0.763371	0.268846
mean(Dist e r-opt rev) ^b (%)	1.446126146	0.019715	0.001769	0.009076	0.003382
STD(Dist e r-opt rev)(%)	0.097680391	0.010934	0.001778	0.019402	0.002857

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 5.6: Output for r=0.6

Number of subroutine calls	4	8	12	16	20
mean(Dist p-opt p) ^a (%)	92.60373514	85.408	85.408	85.408	85.408
STD(Dist p-opt p)(%)	0.020624199	1.5E-14	1.5E-14	1.5E-14	1.5E-14
mean(Dist e r-opt rev) ^b (%)	84.89700859	72.94526	72.94526	72.94526	72.94526
STD(Dist e r-opt rev)(%)	0.037832721	1.5E-14	1.5E-14	1.5E-14	1.5E-14

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 5.7: Output for r=0.98

Num. products	10	15	20	25	30
Mean(Running time)	19.61576	24.55999	29.89096	37.12812	44.11999
STD(Running time)	0.171212	0.20863	0.032755	0.087302	0.651396
Mean(dist. price-opt p)(%)	1.206223	0.709959	0.193776	0.343476	0.871809
STD(dist. price-opt p)(%)	0.016666	0.009886	0.010518	0.011491	0.006673

Table 5.8: Sensitivity to the number of products

5.3.4 Number of products

Finally, we measure the computation time and the precision of the algorithm as the number of products increase significantly. We introduce the parameter dim , which corresponds to the number of products as well as to the dimension of the demand matrix.

The demand is of the following form

$$\mathbf{D}(\mathbf{P}) = \mathbf{A} - \mathbf{B} \cdot \mathbf{P} + \epsilon \text{ with,}$$

$$A = \begin{bmatrix} 5000 \\ 5000 \\ \dots \\ 5000 \end{bmatrix}, B = \begin{bmatrix} dim-1 & -0.2 & -0.2 & \dots & -0.2 \\ -0.2 & dim-1 & \dots & -0.2 & -0.2 \\ & & \dots & & \\ -0.2 & -0.2 & \dots & -0.2 & dim-1 \end{bmatrix} \text{ and } P = \begin{bmatrix} P_1 \\ P_2 \\ \dots \\ P_{dim} \end{bmatrix}.$$

As before, all the noises follow the Normal distribution and standard deviation 10. We consider values

$$dim = 10, 15, 20, 25, 30, 35, 40, 45, 50.$$

For every value of dim , the degree of diagonal dominance is $r = 0.2$. We consider for every component a Normal noise with mean zero and standard deviation 10. For every value of dim , we run 5 subroutines of 1,000 iterations per product and we compute the running time as well as the distance between the final price and the optimal price. Then the algorithm is repeated 10 times and we take the mean and the standard deviation of the quantities defined above. The results are presented in Table 5.8 and 5.9 as well as in Figure 5-4.

Num. products	35	40	45	50
Mean(Running time)	52.13571	61.47441	71.89239	83.11926
STD(Running time)	0.095289	0.13143	0.321532	0.243009
Mean(dist. price-opt. price)(%)	1.423016	1.95963	2.49596	3.042724
STD(dist. price-opt. price)(%)	0.009432	0.00802	0.011469	0.009432

Table 5.9: Sensitivity to the number of products (continued)

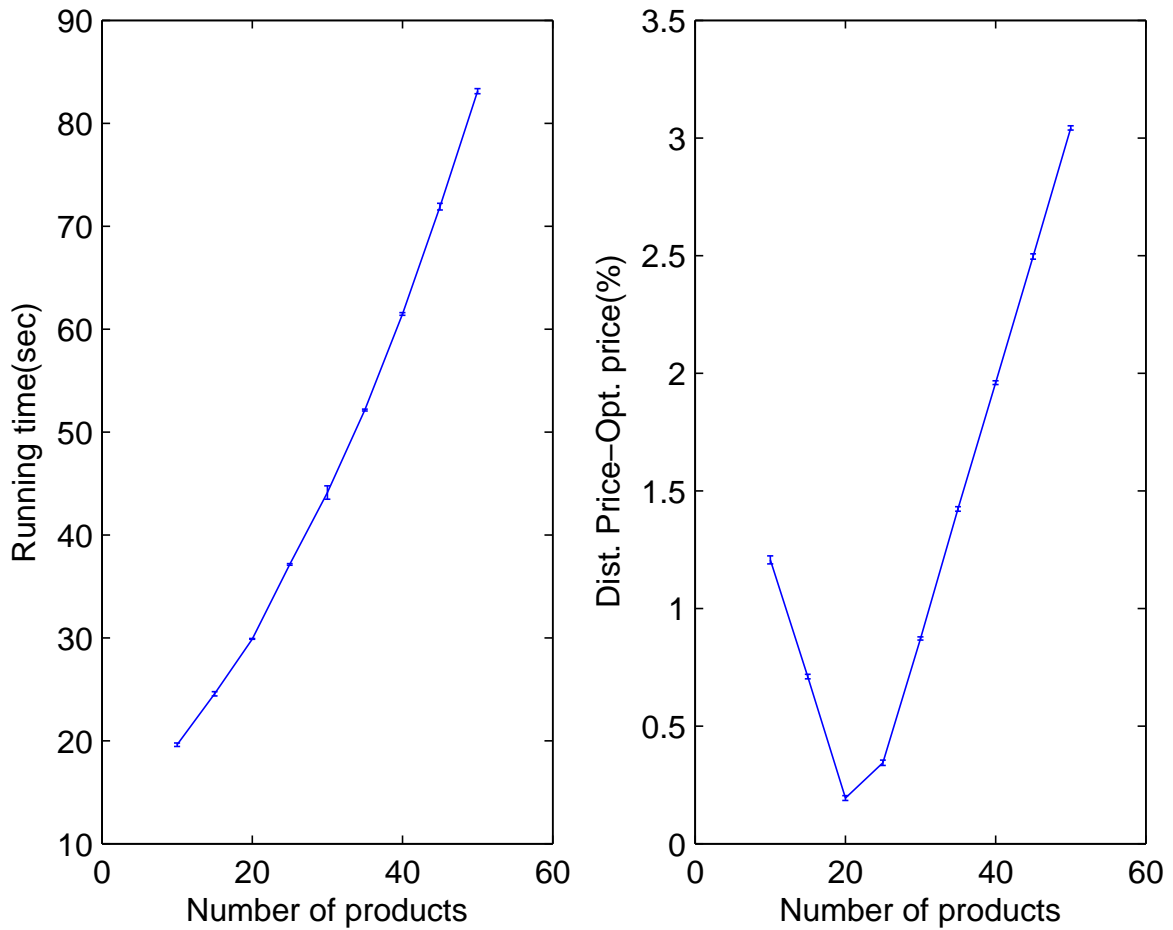


Figure 5-4: Sensitivity to the number of products

Surprisingly, the accuracy of the final output improves when the number of products increases from 10 to 20 and then it worsens. The best accuracy is reached for $dim = 20$.

5.4 Insights

Because the learning phase appeared to estimate the demand intercept accurately with a good convergence rate, we measured the performance of the tatonnement algorithm assuming that the intercept was known a priori. As expected, the accuracy of the solution decreases when the standard deviation of the noise on the demand increases, even if the algorithm seems somehow robust. A critical parameter is the degree of diagonal dominance of the demand sensitivity matrix. When this degree is above 0.95, the performance worsens significantly. Finally, as the number of products increase beyond a threshold, the accuracy decreases but remains satisfactory.

Chapter 6

Non linear parametric models

We conclude this thesis by showing how the same data-driven approach can be used for nonlinear parametric demand models. In this chapter, we focus on the loglinear and constant elasticity demand models.

6.1 Demand models

The demand models we consider are widely used in the revenue management literature. For a thorough description, we refer the reader to [37].

6.1.1 Loglinear demand

Single product case

For the single-product case, the assumptions are the same as for the linear demand single product case. The only difference is in the form of the demand function (see chapter 2).

In this case, a warehouse is selling a product at a price p , which has to be between p_{min} and p_{max} . The demand function is of the form

$$d(p) = \varepsilon \cdot \exp(a - b \cdot p),$$

where $a > 0$ and $b > 0$ are unknown demand parameters and the noise $\varepsilon \geq 0$ is multiplicative such that $\mathbb{E}(\varepsilon) = 1$. Under these conditions, the expected revenue is concave for $p \in [0, \frac{2}{b}]$ and the optimal price is $p^* = \frac{1}{b}$.

The time horizon is infinite. The goal of the firm is to define a pricing strategy in order to achieve the following objectives. (Theses are similar to the linear case demand in Chapter 2.):

- Learn the demand parameters a and b .
- Learn the optimal pricing strategy.

Multi product case

This model can be extended to the multi-product case also, in which a warehouse is selling m products. The price for product i is bounded by $P_i \in [(P_{min})_i, (P_{max})_i]$. The demand for product i is

$$D_i(\mathbf{P}) = \varepsilon_i \cdot \exp(A_i - B_{ii} \cdot P_i + \sum_{j \neq i} B_{ij} \cdot P_j),$$

where for all i , $\varepsilon_i \geq 0$, $\mathbb{E}(\varepsilon_i) = 1$, $A_i > 0$, $B_{ii} > 0$ and for $j \neq i$, $B_{ij} > 0$. As before, the demand parameters are unknown.

The total expected revenue is given by the following

$$\mathbb{E}(\Pi(\mathbf{P})) = \sum_{i=1..m} [P_i \cdot \mathbb{E}(D_i(\mathbf{P}))]$$

This model is popular in econometric studies and has several beneficial theoretical and practical properties. For example, demand is always positive.

As before, the time horizon is infinite and the goal of the firm is to maximize its total revenue as well as to learn its demand parameters. Note the importance of restricting the

prices to a constrained set.

6.1.2 Constant elasticity demand

Single-product case

The constant elasticity demand function in the single product case is of the form:

$$d(p) = \varepsilon \cdot a \cdot p^{-b},$$

where $\varepsilon \geq 0$, $\mathbb{E}(\varepsilon) = 1$, $a > 0$ and $b \geq 0$ are constant. As before, the prices will be bounded by $[p_{min}, p_{max}]$.

Note that the price that maximizes the revenue will be p_{max} if $b < 1$ and p_{min} if $b > 1$. From this standpoint, it is a somewhat ill-behaved demand model. To make the model more realistic, we will assume that there is a fixed per unit cost and that $b > 1$.

Then the expected profit $(p - c) \cdot \exp(d(p))$ is concave for $p \leq \frac{(b+1) \cdot c}{b-1}$ and the optimal price is $p^* = \frac{c \cdot b}{b-1}$.

Multi-product case

This model can also be extended to the multi-product case. A warehouse is selling m products and the price for product i is P_i bounded by $[(P_{min})_i, (P_{max})_i]$.

The demand for product i is

$$D_i(\mathbf{P}) = \varepsilon_i \cdot A_i \cdot \frac{P_i^{B_{ii}}}{\prod_{j \neq i} P_j^{B_{ij}}},$$

where for all i , $\mathbb{E}(\varepsilon_i) = 1$, $A_i > 0$, $B_{ii} > 0$, and for $j \neq i$, $B_{ij} > 0$.

The total expected revenue is given by

$$\mathbb{E}(\Pi(\mathbf{P})) = \sum_{i=1..m} [P_i \cdot \mathbb{E}(D_i(\mathbf{P}))].$$

6.2 Single-product case

6.2.1 Loglinear demand

Optimization and learning

Eventually, the firm wants to optimize its expected profit. Hence the optimal price p^* is the solution of the following nonlinear program:

$$\max_{p \in [p_{min}, p_{max}]} p \cdot \mathbb{E}d(p).$$

Here the optimal price would be

$$p^* = \max \left(\min \left(\frac{1}{b}, p_{max} \right), p_{min} \right).$$

Note that unlike the linear demand case, the optimal price depends only on one of the demand parameters, namely b .

The underlying idea is to learn the demand parameters by linear regression after a change of variables. Hence, the same consistency theorem as in Chapter 2 will apply.

By taking the logarithm of the demand, we get

$$\ln(d(p)) = a - b \cdot p + \ln(\varepsilon),$$

where $\ln(\varepsilon)$ is a noise. This is a similar form to the single product case for an affine demand

model. Note that using Jensen's inequality because \ln is a concave function

$$\mathbb{E}(\ln(\varepsilon)) \leq \ln(\mathbb{E}(\varepsilon)) = 0.$$

Therefore, a bias exists even if in practice, it will be close to zero.

We can write

$$\ln(d(p)) = a + \mathbb{E}(\ln(\varepsilon)) - b.p + [\ln(\varepsilon) - \mathbb{E}(\ln(\varepsilon))].$$

Pricing algorithm

Exactly as before, we assume that we have bounds on the optimal price. We know that the optimal price p^* is such that

$$M \leq p^* \leq N$$

with

$$M - 2(N - M) > p_{min}.$$

We use the following algorithm:

Algorithm:

1. Let a_n and b_n denote the parameters estimated through regression on the prices and the logarithm of the corresponding observed demands thus far.

2. Set:

$$p_n = \max\left(\min\left(\frac{1}{b_n}, N\right), M\right)$$

if there does not exist i such that $n = \lfloor 2^{\sqrt{i}} \rfloor$. Otherwise, set

$$p_n = \max\left(\min\left(\frac{1}{b_n}, N\right), M\right) - \gamma.$$

3. Update the parameter estimates and repeat the procedure.

Then the following theorem applies:

Theorem 6.2.1. *Assume that we have the loglinear demand function $d(p) = \varepsilon \cdot \exp(a - b \cdot p)$ and that at every step we compute the estimated parameters a_n and b_n using linear regression between the observed prices and the logarithm of the observed demands. Moreover we pick the next price such that:*

$$p_n = \max \left(\min \left(\frac{1}{b_n}, N \right), M \right) + \gamma_n$$

where

$$\gamma_n = \begin{cases} -\gamma & \text{if there exists } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

and $[M, N]$ is a price range within which we want to be most of the time.

We also assume that the noise is such that:

$$\mathbb{E}(|\ln(\varepsilon)|^\alpha) \leq \infty \text{ a.s. for some } \alpha > 2,$$

and that

$$2 \cdot (N - M) < \gamma \leq M - p_{\min}.$$

Then the following holds:

1. *The estimated demand parameters a_n and b_n converge almost surely to the true parameters $a + \mathbb{E}(\ln(\varepsilon))$ and b .*
2. *If the bounds are adequate, i.e. $M - 2 \cdot (N - M) > p_{\min}$, then the price converges in the Cesaro sense to the optimal price, which means that the average price converges to the optimal price.*
3. *A subset $\mathfrak{A}(\mathbb{N}) = \{n \mid \text{there exists no } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor\}$ of the prices converges to the optimal price almost surely. Moreover on this subset the expected revenue also converges to the optimal revenue.*

Proof. The proof is similar to the single product linear demand case. Note that the estimate a_n of the demand parameter a will be biased because of the bias in the noise but b_n is unbiased. However, it has no effect on the price, which is a function only of b_n . \square

Single-product with demand capacity case

As for the single product case, we will now add a capacity constraint. We assume that at every time period, there is a demand capacity d_{max} . Hence, we must have

$$\mathbb{E}d(p) = \exp(a - b.p) \leq d_{max}.$$

Mathematically, the objective is to solve

$$\begin{aligned} \max \mathbb{E}(\Pi(p)) &= p.\exp(a - b.p) \\ \text{s.t.} \quad &p_{min} \leq p \leq p_{max} \end{aligned}$$

$$\mathbb{E}d(p) = \exp(a - b.p) \leq d_{max}$$

This optimization problem is equivalent to

$$\begin{aligned} \max p.\exp(a - b.p) \\ \text{s. t} \quad \max(p_{min}, \frac{a - \ln(d_{max})}{b}) \leq p \leq p_{max}. \end{aligned}$$

Lemma 6.2.2. *The solution to the above problem is*

$$p^*(a, b, d_{max}) = \begin{cases} p_{max} & \text{if } \frac{1}{b} \geq p_{max} \\ \frac{1}{b} & \text{if } \max(p_{min}, \frac{a - \ln(d_{max})}{b}) \leq \frac{1}{b} \leq p_{max} \\ \max(p_{min}, \frac{a - \ln(d_{max})}{b}) & \text{if } \frac{1}{b} \leq \max(p_{min}, \frac{a - \ln(d_{max})}{b}) \end{cases}$$

Proof. This is an easy application of the KKT Conditions. □

As before, the principal difficulty is that the demand parameters are unknown. To solve the problem with a capacity constraint, the methodology we use is very similar to the problem without capacity. At every step, a_n and b_n are estimated using linear regression. We denote by $p^*(a_n, b_n, d_{max})$ the optimal solution of the optimization problem using the estimated parameters.

As for the single product case without any capacity constraint, we assume that we know that the optimal price lies within M and N . Then, we have the following algorithm:

We use the following algorithm:

Algorithm:

1. Let a_n and b_n denote the parameters estimated through regression on the prices and the logarithm of the corresponding observed demands thus far.

2. Set:

$$p_n = \max(\min(p^*(a_n, b_n, d_{max}), N), M) + \gamma_n$$

where

$$\gamma_n = \begin{cases} -\gamma & \text{if there exists } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

3. Update the parameter estimates and repeat the procedure.

Then the following theorem holds,

Theorem 6.2.3. *Assuming that we have a loglinear demand function $d(p) = \epsilon \cdot \exp(a - b \cdot p)$, and that at every step we compute the estimated parameters a_n and b_n using linear regression between $\ln(d(p))$ and p . Moreover we pick the next price such that:*

$$p_n = \max(\min(p^*(a_n, b_n, d_{max}), N), M) + \gamma_n$$

where

$$\gamma_n = \begin{cases} -\gamma & \text{if there exists } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

with $\gamma > 2.(N - M)$.

We also assume that the noise is such that,

$$\mathbb{E}(|\ln(\epsilon)|^\alpha) \leq \infty \text{ a. s for some } \alpha > 2$$

and that

$$2.(N - M) < \gamma \leq M - p_{min}.$$

Then the following hold:

1. The parameters a_n and b_n converge almost surely to the parameters $a + \mathbb{E}(\ln(\epsilon))$ and b .
2. If the bounds are adequate, i.e. $M - 2.(N - M) > p_{min}$, the price converges in the Cesaro sense to $p^*(a + \mathbb{E}(\ln(\epsilon)), b, d_{max})$, which means that the average price converges to $p^*(a + \mathbb{E}(\ln(\epsilon)), b, d_{max})$.
3. A subset $\mathfrak{A}(\mathbb{N}) = (n | \text{there exists no } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor)$ of the prices converges to $p^*(a + \mathbb{E}(\ln(\epsilon)), b, d_{max})$ almost surely. Moreover on this subset the expected revenue converges to $\mathbb{E}(\Pi(p^*(a + \mathbb{E}(\ln(\epsilon)), b, d_{max}))$

Proof. The proof is similar to the uncapacitated case. Unlike before, because a_n is biased, the price returned will be biased if the capacity is binding. Nevertheless, in practice, for small noises, the bias will not be significant. \square

6.2.2 Constant elasticity demand

Optimization and learning

We assume that there is a fixed per unit cost $c > 0$. Eventually, the firm wants to optimize its expected profit. Hence, the optimal price p^* is the solution of the following nonlinear

program:

$$\max_{p \in [p_{min}, p_{max}]} (p - c) \cdot \mathbb{E}d(p).$$

Here the optimal price would be

$$p^* = \max(\min(\frac{b \cdot c}{b - 1}, p_{max}), p_{min}).$$

By taking the logarithm of the demand, we get

$$\ln(d(p)) = \ln(a) - b \cdot \ln(p) + \ln(\varepsilon),$$

where $\ln(\varepsilon)$ is a noise. Note that as for the loglinear demand case, there is also a bias.

Pricing algorithm

As before we assume that we have bounds on the optimal price. We know that the optimal price p^* is such that

$$M \leq p^* \leq N$$

with

$$\ln(M) - 2(\ln(N) - \ln(M)) > \ln(p_{min}).$$

We use the following algorithm:

Algorithm:

1. Let a_n and b_n denote the parameters estimated through regression on the logarithm of the prices and the logarithm of the corresponding observed demands thus far.
2. Set:

$$\ln(p_n) = \ln \left[\max(\min(\frac{c \cdot b_n}{b_n - 1}, N), M) \right] + \gamma_n,$$

where

$$\gamma_n = \begin{cases} -\gamma & \text{if there exists } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

3. Update the parameter estimates and repeat the procedure.

Then the following theorem holds:

Theorem 6.2.4. *Assume that we have the following demand function $d(p) = \epsilon \cdot a \cdot p^b$ and that at every step we compute the estimated parameters a_n and b_n using linear regression between the logarithm of the observed price and the logarithm of the observed demand. Moreover, we pick the next price such that:*

$$\ln(p_n) = \ln \left[\max\left(\min\left(\frac{c \cdot b_n}{b_n - 1}, N\right), M\right) \right] + \gamma_n,$$

where

$$\gamma_n = \begin{cases} -\gamma & \text{if there exists } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor \\ 0 & \text{otherwise} \end{cases}$$

where N and M is a price range within which we want to be most of the time.

We also assume that the noise is such that:

$$\mathbb{E}(|\ln(\epsilon)|^\alpha) \leq \infty \text{ a. s. for some } \alpha > 2$$

and that

$$2 \cdot (\ln(N) - \ln(M)) < \gamma \leq \ln(M) - \ln(p_{\min}).$$

Then the following points hold:

1. The estimated demand parameters $\ln(a_n)$ and b_n converge almost surely to the parameters $\ln(a) + \mathbb{E}(\ln(\epsilon))$ and b .
2. If the bounds are adequate, i.e. $\ln(M) - 2 \cdot (\ln(N) - \ln(M)) > \ln(p_{\min})$, the logarithm of the price converges in the Cesaro sense to the logarithm of the optimal price, which

means that the average logarithm of the price converges to the logarithm of the optimal price.

3. A subset $\mathfrak{A}(\mathbb{N}) = \{n \mid \text{there exists no } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor\}$ of the prices converges to the optimal price almost surely. Moreover, on this subset the expected revenue also converges to the optimal revenue.

Proof. The proof is similar to single product linear case, considering the logarithm of the price. Note that the estimate a_n of the demand parameter a will be biased because of the bias in the noise but b_n is unbiased. However, it has no effect on the price, which is only a function of b_n and c . □

Note that as before, we could consider the constant elasticity demand model with a capacity constraint. For the sake of brevity, we do not present this case, as it is similar to the loglinear case with a capacity constraint.

6.3 Multi-product case

Finally, we consider is the multi-product case. When the demand is nonlinear, we do not know if the tatonnement algorithm works as we do not have simple theoretical conditions to guarantee that the price will converge to the optimal price. Hence, for this setting, we are unable to present a formal proof of the algorithm we proposed. Nevertheless, we provide a heuristic, explain it, and run simulations to test it in practice. This heuristic can be used both for the linear demand, loglinear demand and constant elasticity demand.

Overall, we have $m^2 + m$ unknown demand parameters. For every product, we have $(m + 1)$ parameters to learn and we have m products. As before, we will use linear regression to estimate the demand parameters. However, in the case of more than one product, we were not able to have consistency conditions for the demand parameters that are easy to check. Indeed, for this case, there exists no closed form expressions for the eigenvalues of

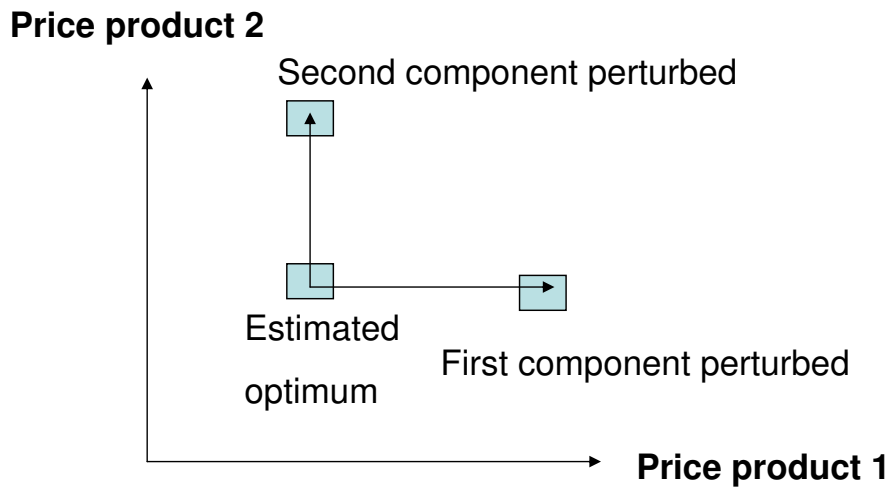


Figure 6-1: Illustration of the perturbation

the covariance matrix, and as a result, the theorem by Lai and Wei [22] is hard to apply.

Nevertheless, we will be able to provide a heuristic method. For each product, in order to learn the $m + 1$ demand parameters, we need $m + 1$ points, which means estimating the exact demand without the noise for $m + 1$ prices. We will learn these price-demand vectors by perturbing the prices as in the affine demand case.

We start with an initial number of price vectors that corresponds to past prices. Using these past prices and their corresponding demand, for each product i , we compute the estimated demand parameters using linear regression after the adequate change of variables. With these estimated demand parameters, we compute the price vector that maximizes the total profit. Then, we use this vector as our price vector for the next time period and iterate the procedure. Additionally, at some periods, instead of pricing at this estimated optimal price, we will perturb one of the price components by a fixed scalar, as shown on Figure 6-1 for the two product case.

Over the whole horizon, the estimated unperturbed optimal price is considered as a fixed price. The idea is that the demand estimates are not going to evolve by a large amount over time hence the price vector that optimizes the profit is not going to change significantly, and will be considered as a fixed price. As we price repeatedly at this “fixed price”, we expect that eventually, we learn the demand function “exactly” for this price. This is our first “exact price-demand vector”. Moreover, we have m other price-demand vectors, which correspond to the estimated optimal price with a perturbation on one of the price components. For these other “fixed prices”, the demand will also be learned “exactly”. Hence, intuitively, we will have $m+1$ price vectors for which the demand is known and the estimated demand parameters should be accurate.

The two questions remaining are:

- When do we need to add a perturbation?
- What are the conditions on the value of the perturbation?

For the single product case, we were able to answer these two questions exactly to guarantee strong consistency of the least square parameters (see Chapter 2).

For the multi product case, we do not have conditions that guarantee consistency anymore. We make the choice to add a perturbation at the same steps as the single product case. We will add the perturbation to every price component alternatively.

The algorithm is described in Algorithm 4 defining as for the single product case:

$$\mathfrak{A}(\mathbb{N}) = \{n \mid \text{there exists no } i \text{ such that } n = \lfloor 2^{\sqrt{i}} \rfloor\}.$$

Algorithm 4 Heuristic Approach for H steps with perturbation γ .

Input: For every product $1 \leq i \leq m$ and $n_0 \geq m + 1$, be input a set of initial prices P_i^k and observed demands D_i^k , with $1 \leq k \leq n_0$.

Output: Set of prices.

$n \leftarrow n_0 + 1$

indicator $\leftarrow 1 \pmod{m}$

repeat

for $i=1,\dots,m$. **do**

 Compute A_i^n, B_{ij}^n for all $1 \leq j \leq m$ using linear regression.

end for

for $i=1,\dots,m$. **do**

 Compute $P_i^{(n)*}$ that maximizes the total profit.

end for

if $n \notin \mathfrak{A}(\mathbb{N})$ **then**

for $i=1..m$ **do**

$P_i^n \leftarrow P_i^{(n)*}$

end for

else

$P_{indicator}^n \leftarrow P_{indicator}^{(n)*} - \gamma$

for $i=1,\dots,m$. **do**

if $i \neq indicator$ **then**

$P_i^n \leftarrow P_i^{(n)*}$

end if

end for

 indicator $\leftarrow indicator+1 \pmod{m}$

end if

$n \leftarrow n+1$

until $n = H$

6.4 Conclusions

In this chapter, we showed how for the single product case, the results extend naturally to nonlinear demand models. For the multi product case, there is no simple way of extending the results of Chapter 4. However, we introduced a heuristic method based on the results of the single product case.

This heuristic is tested in the next chapter through simulations. Moreover, because the heuristic works also for the linear demand case in the multiple product setting, we will compare it to the tatonnement algorithm developed in Chapter 4.

Chapter 7

Simulations: Nonlinear case and heuristic

We test the quality of our algorithms when the demand is nonlinear, as well as the performance of the heuristic.

7.1 Single product case

7.1.1 Loglinear demand

Sensitivity to the standard deviation of the noise

We consider the following demand function

$$d(p) = \varepsilon \cdot \exp\left(6 - \frac{p}{100}\right).$$

Then, $\ln(d(p)) = \left(6 - \frac{p}{100}\right) + \ln(\varepsilon)$.

The optimal price is $p^* = 100$ and the optimal expected profit is

$$\mathbb{E}\Pi(p^*) = p^* \cdot \mathbb{E}d(p) = 14,841.$$

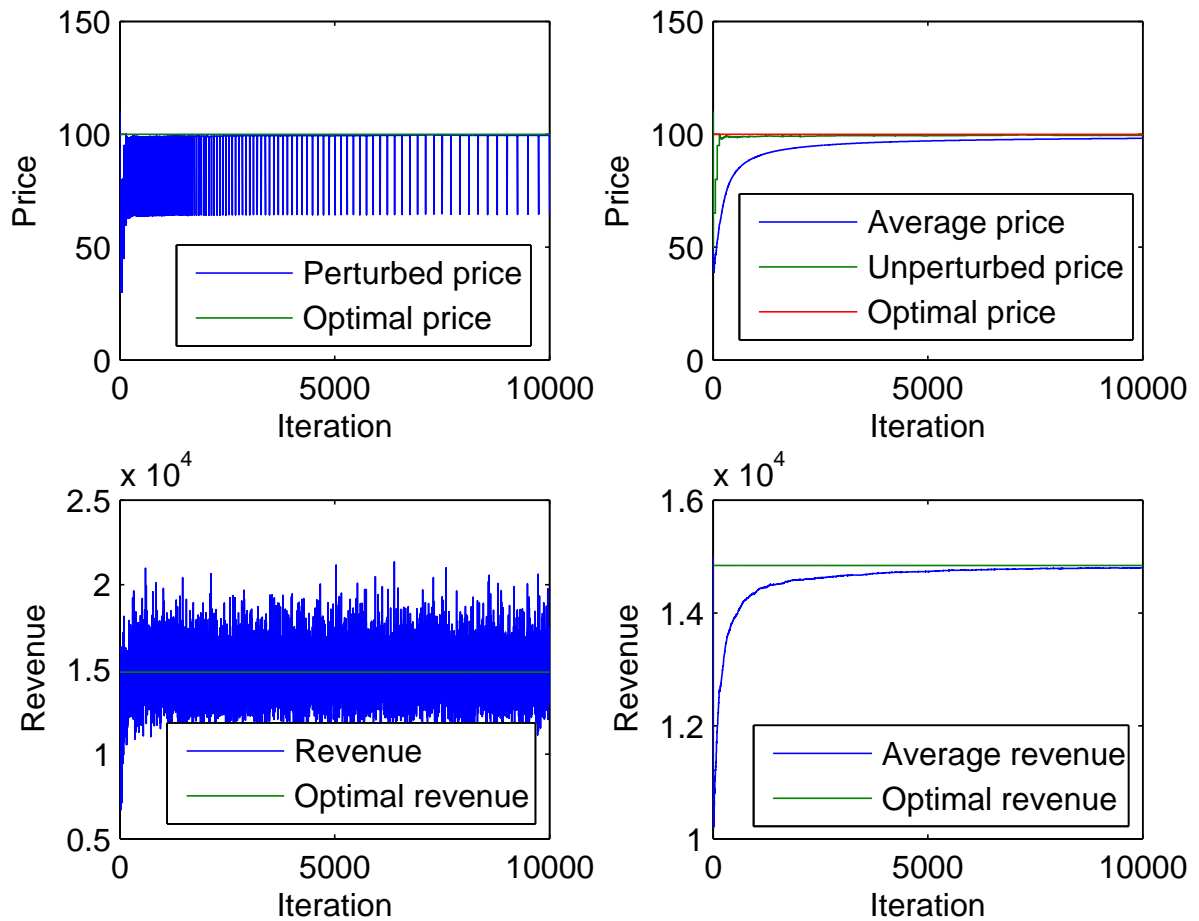


Figure 7-1: Output for the loglinear demand case

Since we may not know precise bounds on the optimal price, we use the improved version of the algorithm described in Chapter 3. We assume that ε has a lognormal distribution with mean 1 and standard deviation 0.1. On Figure 7-1, we plot a typical output of the algorithm. It is very similar to the output for the affine single product case.

In what follows, we investigate the sensitivity of the algorithm with respect to changes in the standard deviation of the noise. The demand noise ε has a lognormal distribution with

Std. dev	0.05	0.1	0.15	0.2	0.25
Mean(Price)	99.84273	99.56549	97.81713	99.82887	102.4261
STD(Price)	1.396352	2.453313	5.147222	3.501324	4.597305
Mean(Exp. revenue)	14839.99	14837.11	14819.38	14833.2	14823.4
STD(Exp. revenue)	1.395046	4.946095	16.30882	8.597226	14.93737

Table 7.1: Loglinear demand: sensitivity to the standard deviation of the noise

mean 1 and standard deviation std , where

$$std = 0.05, 0.1, 0.15, 0.2 \text{ and } 0.25.$$

We run the algorithm 10,000 steps . Table 7.1 represents the price returned and expected revenue at the end of the algorithm over 10 runs.

As expected, the accuracy worsens significantly when the standard deviation of the noise increases. Note that even if the standard deviation might seem small, it is actually significant compared to the dependence between the expected demand and price, because $\mathbb{E}[\ln(d(p))] = (6 - \frac{p}{100})$. This means that if the price increases by one unit, the expected demand will only increase by $\frac{1}{100}$ of a unit.

Sensitivity to the parameters of the demand

Remember that if the demand function is $d(p) = \varepsilon \cdot \exp(a - b \cdot p)$, the optimal price is a function of only parameter b . We test the sensitivity to this parameter b when ε follows a lognormal distribution with mean 1 and standard deviation 0.1, and $a = 6$.

We consider the following values for b :

$$b = \frac{1}{60}, b = \frac{1}{90}, b = \frac{1}{120}, b = \frac{1}{150} \text{ and } b = \frac{1}{180}.$$

As in Chapter 3 for the linear case, we use the improved version of the algorithm with the same size for all the subintervals. We represent the distance from the unperturbed price to the optimal price $|\frac{P-P^*}{P^*}| \cdot 100$, and from the expected revenue to the optimal revenue $|\frac{\mathbb{E}\Pi(P) - \mathbb{E}\Pi(P^*)}{\mathbb{E}\Pi(P^*)}| \cdot 100$ as a function of the number of iterations after 10 runs.

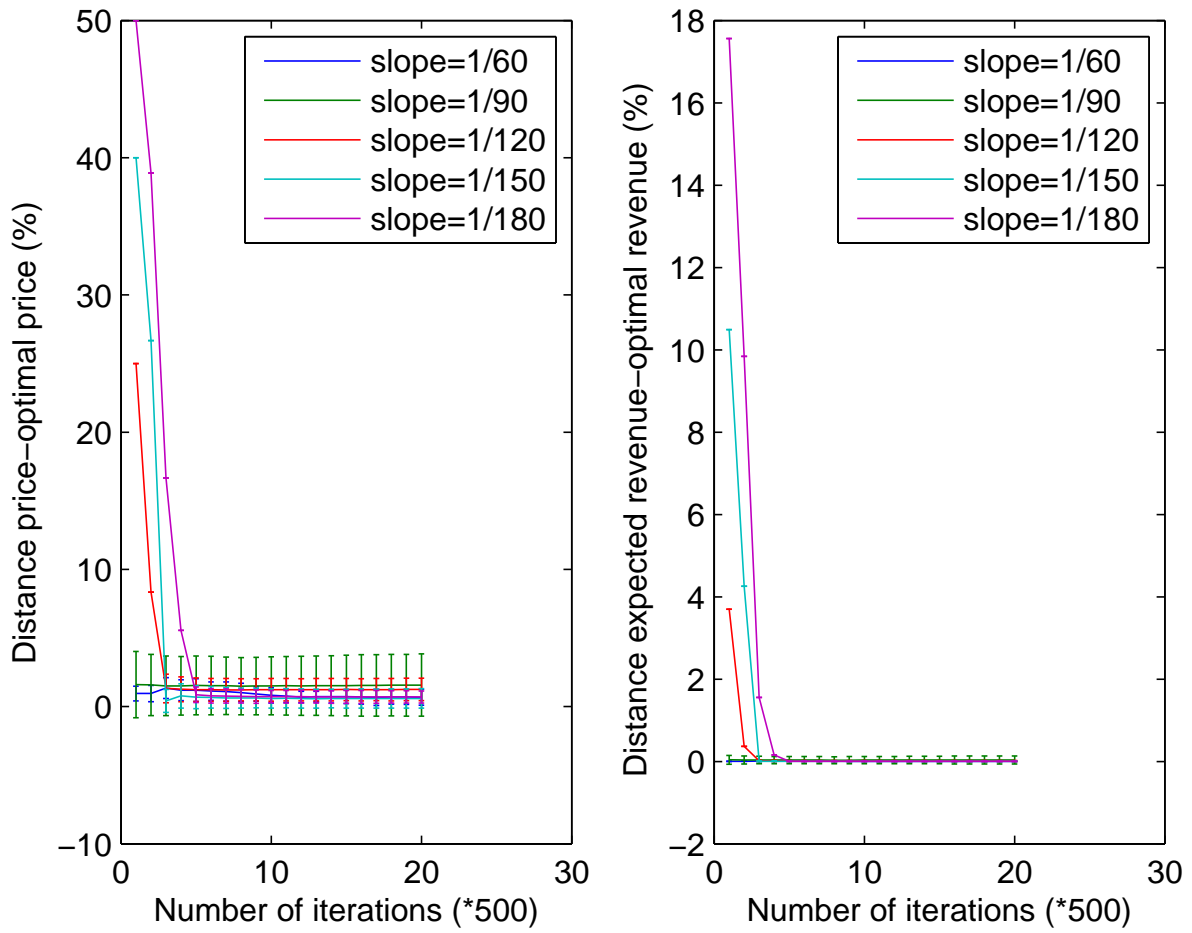


Figure 7-2: Loglinear demand: sensitivity to the slope

We see on Figure 7-2 that the transient phase gets longer when the slope decreases. This is due to the fact that the optimal price gets farther away from the original interval. Moreover, we notice in Table 7.2 that the longer the transient phase, the more accurate the final solution is. It is particularly visible for the revenue and confirms the results of Chapter 3.

	Slope	$\frac{1}{60}$	$\frac{1}{90}$	$\frac{1}{120}$	$\frac{1}{150}$	$\frac{1}{180}$
Mean(Dist price-o p)^a(%)		1.671014	1.56363	1.249828	0.594863	0.696977
STD(Dist price-o p)(%)		0.577562	2.269714	0.822869	0.706619	0.442096
Mean(Dist e r-o r)^b(%)		0.003787	0.037056	0.010744	0.00402	0.003327
STD(Dist e r-o r)(%)		0.006923	0.098728	0.012318	0.007396	0.003862

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 7.2: Loglinear demand: sensitivity to the slope

	Capacity	exp(4.5)	exp(5)	exp(5.5)	exp(6)
Mean(Dist price-opt. price)^a(%)		0.029197	0.035375	0.435281	1.124411
STD(Dist price-opt. price)(%)		0.047317	0.058265	0.650029	0.918003
Mean(Dist exp. rev.-opt. rev.)^b(%)		0.029197	0.017671	0.002815	0.010055
STD(Dist Exp. rev.-opt. rev.)(%)		0.047316	0.029105	0.006811	0.010493

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 7.3: Loglinear demand with capacity

7.1.2 Loglinear demand with capacity

The next case we consider is when there is a demand capacity for the loglinear demand case. Remember that if the expected demand function is $\mathbb{E}(d(p)) = \exp(a - b \cdot p)$, and the capacity is d_{max} , the optimal price will be

$$p^* = \max\left(\frac{1}{2 \cdot b}, \frac{a - \ln(d_{max})}{b}\right).$$

As before $a = 6$, $b = \frac{1}{100}$. The noise ε is lognormal with mean 1 and standard deviation 0.05. We consider the following capacities:

$$capacity = \exp(4), \exp(4.5), \exp(5.5) \text{ and } \exp(6).$$

The last two capacity constraints are not binding. As before, for every capacity, we run the algorithm 10,000 iterations. Then, we repeat the procedure 10 times.

Standard deviation	0.025	0.05	0.075	0.1	0.125
Mean(Price)(%)	98.50543	107.7798	102.2443	110.4749	115.7419
STD(Price)(%)	5.333876	17.56673	18.31435	22.27893	33.75063
Mean(Exp. rev.)^a(%)	4984.601	4906.455	4908.712	4829.528	4632.056
STD(Ex. rev.)(%)	18.1275	165.9394	171.3542	173.3315	199.6612

^aExpected revenue

Table 7.4: Constant-elasticity demand: sensitivity to the standard deviation of the noise

We see in Table 7.3 that the errors are typically of the same order. Surprisingly, the accuracy seems to be better when the capacity is binding whereas, in this case, we were expecting a bias on the price (see Chapter 6) but this bias must be pretty insignificant.

7.1.3 Constant elasticity demand

Sensitivity to the standard deviation of the noise

We consider the following demand form:

$$d(p) = \varepsilon \cdot 100 \cdot \left[\frac{100}{p} \right]^2.$$

Remember that for this case we assume that there is a fixed per unit cost $c = 50$. Then the optimal price is $p^* = 100$ and the optimal expected profit is

$$\mathbb{E}(\Pi(P^*)) = (p^* - c) \cdot \mathbb{E}(d(p^*)) = 50,000.$$

The demand noise ε has a log normal distribution with mean 1 and standard deviation std , where

$$std = 0.025, 0.05, 0.075, 0.01 \text{ and } 0.125.$$

As in the loglinear demand case, for every standard deviation of the noise, we run the algorithm 10,000 iterations and repeat the procedure 10 times. We see in Table 7.4 the results for the price as well as the expected revenue.

	Slope	2	3	4	5	6
Mean(Dist price-opt p)^a(%)	0.055541	0.053392	0.081149	0.247475	0.25073	
STD(Dist price-opt p)(%)	0.043052	0.031305	0.054312	0.064022	0.13004	
Mean(Dist e r-o r)^b(%)	0.151005	0.194772	0.213248	0.223403	0.277234	
STD(Dist e r-o r) (%)	0.130805	0.127968	0.162896	0.125098	0.214487	

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 7.5: Constant elasticity demand: sensitivity to the demand parameters

The accuracy is worse compared to the loglinear demand, even for smaller noises. This is probably due to the fact that ,for constant elasticity demand, the regression is performed for variables $y = \ln d(p)$ and $x = \ln(p)$. Then we have to consider the exponent of x to get the actual price, hence the errors inflate.

Sensitivity to the parameters of the demand

Assuming a constant elasticity demand function $d(p) = \varepsilon \cdot a \cdot p^{-b}$ and a per unit cost c , the optimal price is $p^* = \frac{c \cdot b}{b-1}$. Hence, it depends on only the parameter b . We test the sensitivity to this parameter when ε is lognormal with mean 1 and standard deviation 0.025 and $a = 100^3$. The parameter b takes the following values

$$b = 2, b = 3, b = 4, b = 5 \text{ and } b = 6.$$

Table 7.5 represents the accuracy of the final price and the final expected revenue for different values of the parameter b . We see that when b increases, the accuracy decreases. As in the loglinear demand case, this is probably due to the fact that when b increases, the transient phase is getting shorter because the optimal price decreases. This confirms that our observations for the linear demand model extends to other nonparametric models.

7.2 Performance of the heuristic

We will conclude by testing the performance of our heuristic method on two demand models for the multiproduct setting: the loglinear demand model as well as the linear demand model. Note that this heuristic can be used for a constant elasticity demand as well, but we do not consider this case for the sake of brevity.

7.2.1 Loglinear case

We consider the following demand functions for two products:

$$D_1(p) = \varepsilon_1 \cdot \exp\left(2 - \frac{1}{100} \cdot p_1 + \frac{1}{1000} \cdot p_2\right)$$

$$D_2(p) = \varepsilon_2 \cdot \exp\left(2 - \frac{1}{100} \cdot p_2 + \frac{1}{1000} \cdot p_1\right)$$

We will restrict the price domain to $\Omega = [0, 300]^2$. On Figure 7-3, we can see the corresponding total profit on this domain. We find numerically that the optimal price is

$$P^* = [111.11, 111.11]$$

We assume that both ε_1 and ε_2 follow the lognormal distribution with mean 1 and standard deviation 0.1.

Figure 7-4 shows the output. Note that the perturbation is added alternatively on P_1 and P_2 .

Next we study the accuracy of the output with respect to the value of the perturbation. We consider the following values for the perturbation

$$\text{perturbation} = 10, 20, 30, 40 \text{ and } 50.$$

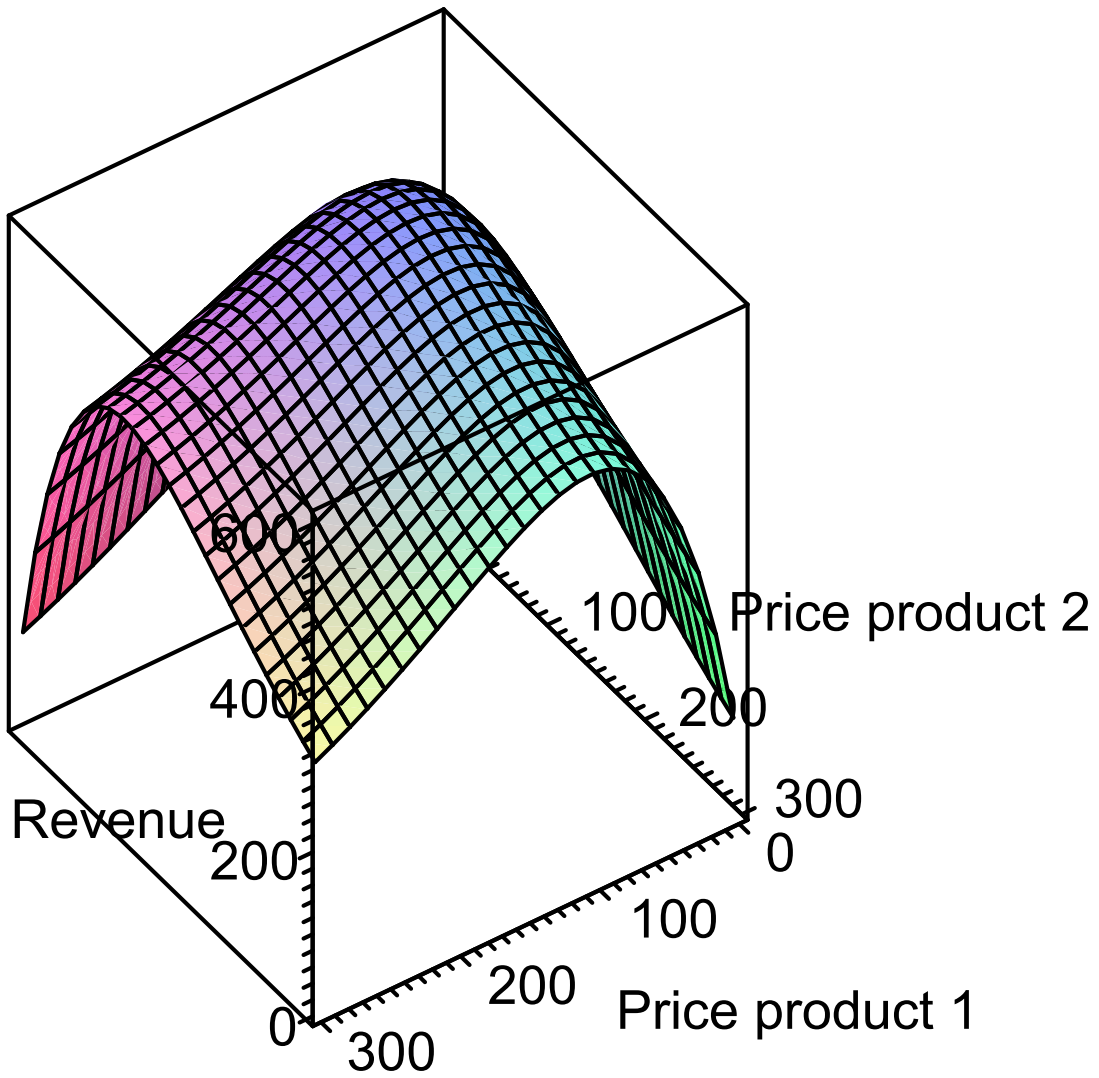


Figure 7-3: Revenue for the loglinear demand case

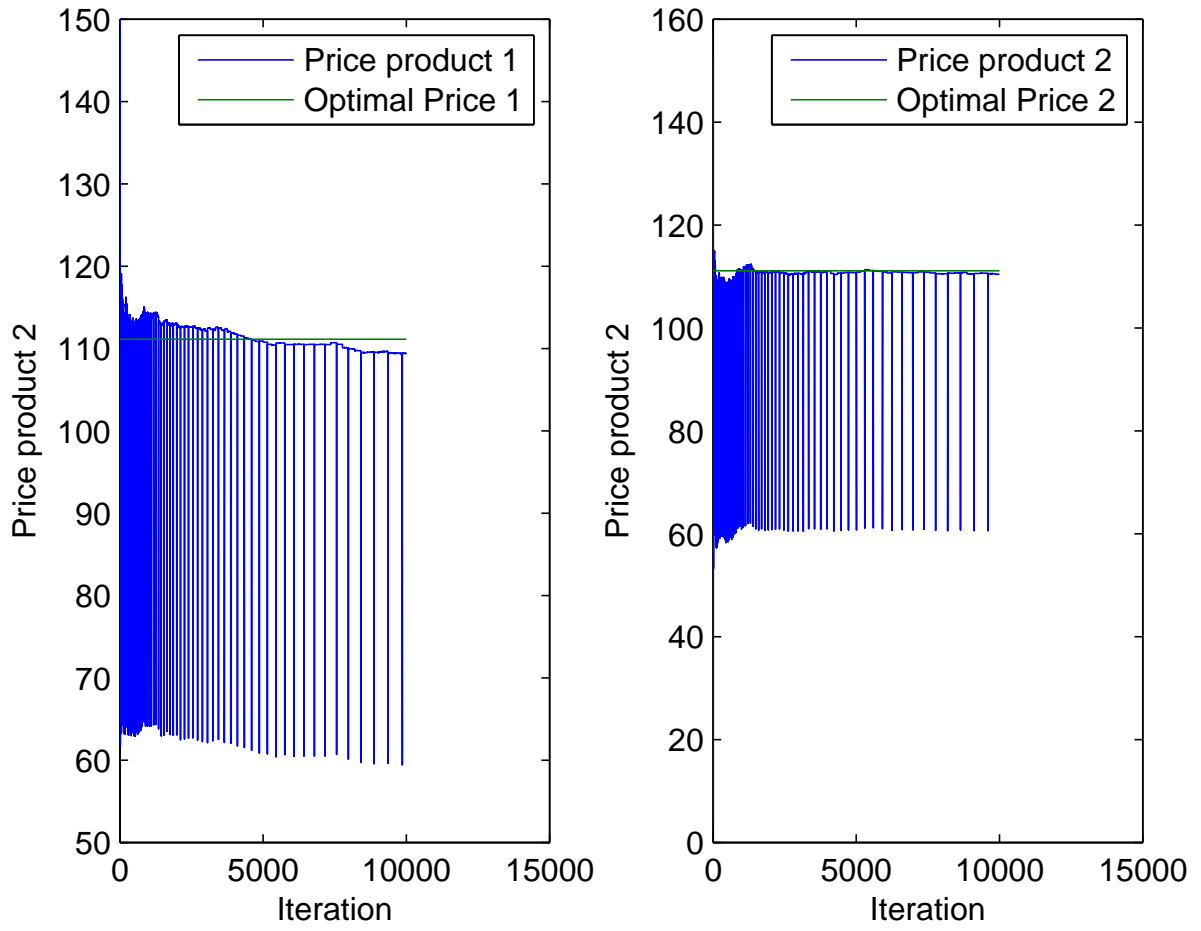


Figure 7-4: Heuristic: loglinear demand, 2 products

Perturbation	10	20	30	40	50
Mean(Dist price-o p) ^a (%)	2.47269	1.047856	0.63113	2.165485	1.497494
STD(Dist price-o p)(%)	1.541931	0.573952	0.593555	0.760837	1.123673
Mean(Dist e r-o r) ^b (%)	0.075047	0.016801	0.010816	0.021392	0.013573
STD(Dist e r-o r)(%)	0.076452	0.011883	0.013466	0.010691	0.011658

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 7.6: Sensitivity to the value of the perturbation

We see in Table 7.6 that when we increase the value of the perturbation, the accuracy increases. But surprisingly, after a threshold at $\text{perturbation} = 30$, the accuracy does not improve anymore and seems to worsen for the price.

Last but not least, we show the significance of adding a perturbation. Using the notations of Chapter 2 (see Theorem 2.2), we plot λ_{min} and $\frac{\log(\lambda_{max})}{\lambda_{min}}$. Note that in order to satisfy the consistency conditions, λ_{min} needs to converge to ∞ and $\frac{\log(\lambda_{max})}{\lambda_{min}}$ to zero. Figure 7-5 plots these quantities when there is no perturbation and Figure 7-6 when the perturbation is 30. We see that $\frac{\log(\lambda_{max})}{\lambda_{min}}$ seems numerically to converge to zero when there is a perturbation whereas it keeps increasing under no perturbation. This seems to justify the heuristic we consider.

7.2.2 Linear demand case

Comparison of heuristic versus tatonnement algorithm

For the multi-product linear case, we can apply the heuristic described in Chapter 6 or the tatonnement method described in Chapter 4. We will compare the performances of these two pricing algorithms and the strategies they induce.

We consider a setting with two products. We consider the following demand functions:

$$D_1(P_1, P_2) = 100 - P_1 + 0.5P_2 + \varepsilon_1,$$

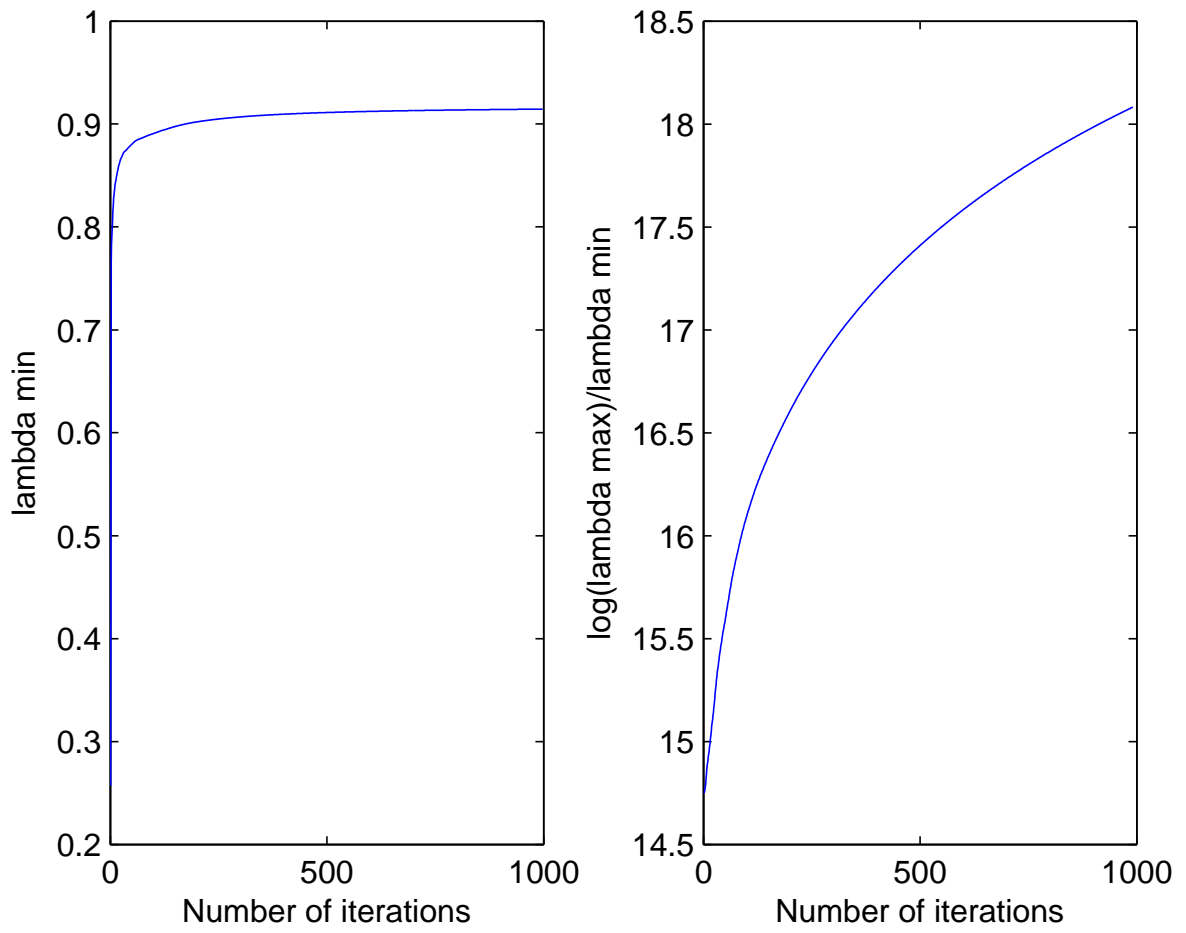


Figure 7-5: Conditions by Lai and Wei without a perturbation

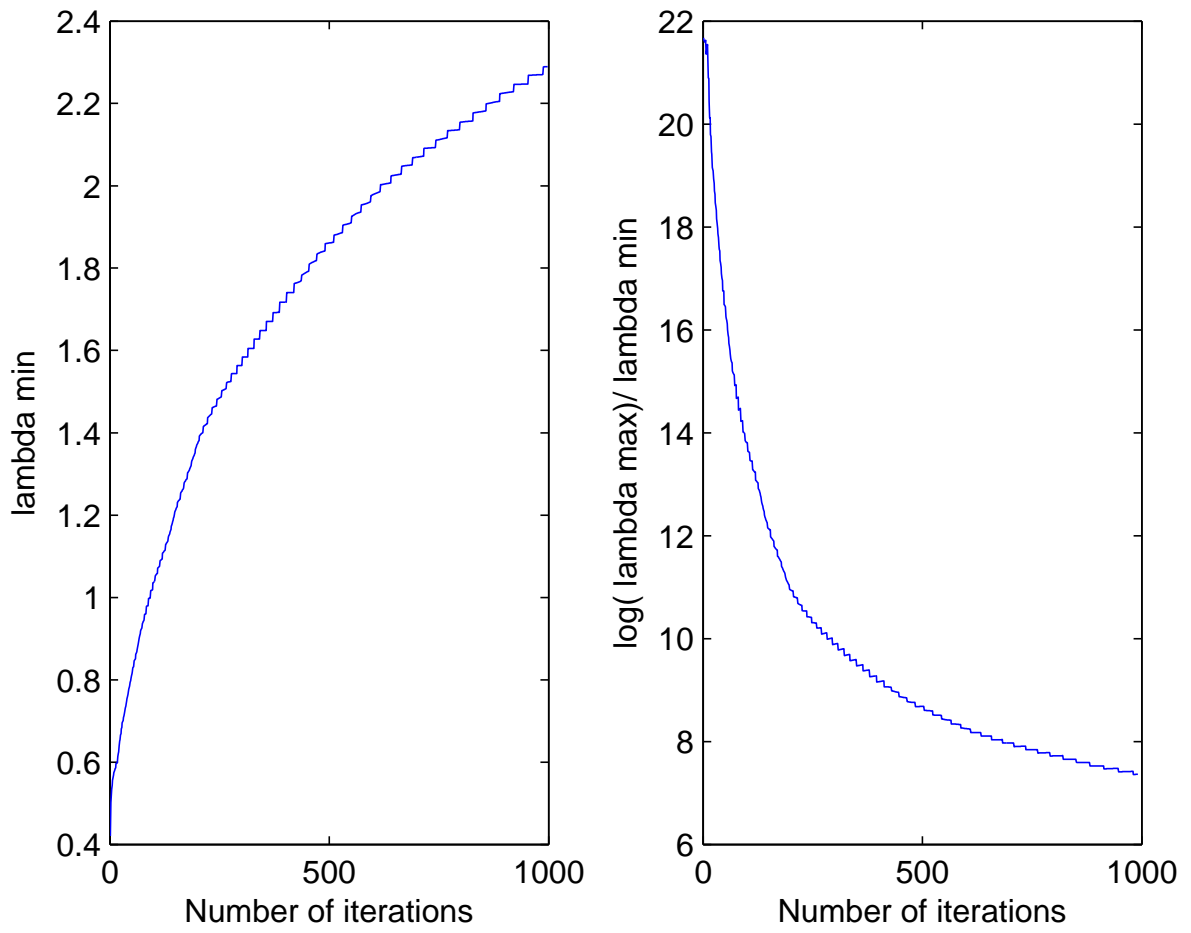


Figure 7-6: Conditions by Lai and Wei with a perturbation

$$D_2(P_1, P_2) = 100 + 0.5P_1 - P_2 + \varepsilon_2.$$

We use $\mathbf{P}_{\min} = [100; 100]$ and $\mathbf{P}_{\max} = [250; 250]$. The matrix is diagonal dominant with $r = 0.5$. The optimal price is the vector: $P_1^* = 100$ and $P_2^* = 100$. The optimal expected profit is $\Pi(P_1, P_2) = 10,000$.

We assume that ε_1 and ε_2 follow the Normal distribution with mean zero and standard deviation *std*, where

$$std = 10, 20, 30, 40 \text{ and } 50.$$

For each standard deviation, we run the heuristic 10,000 times and repeat the procedure 10 times. We take *perturbation* = 30. The reason for this choice is that it was shown to be the best one for the loglinear demand case. For the algorithm described in Chapter 4, assuming that the demand parameter \mathbf{A} is known, we will run 5 subroutine calls on every product (10 in total) and for each subroutine, we will have 1,000 iterations. Hence, the total number of iterations for both algorithms is the same. As in the heuristic method, we repeat the algorithm 10 times. However, one might argue that the comparison is not fair because \mathbf{A} is assumed to be known for the tatonnement algorithm but not for the heuristic. So we also run the heuristic assuming that \mathbf{A} is known.

In Tables 7.7, 7.8 and 7.9 we give the results for the accuracy of the prices and expected revenue for the two algorithms at different steps.

We notice that after 10,000 iterations the heuristic performs generally worse than the algorithm developed in Chapter 4. However, at 2000 iterations, for most of the noises, the accuracy of the heuristic is better. In the short run, the heuristic will have prices closer to the optimal than the tatonnement algorithm. In the long run, however, the heuristic will eventually be outperformed by the algorithm of Chapter 4. Moreover, the heuristic with \mathbf{A} known performs extremely well, far better than the heuristic without this additional assumption.

Standard deviation	1	11	21	31	41
Mean(D pr-opt p ^a (%))(2000)	9.994586	9.98242	10.00699	9.460415	6.838934
STD(D pr-opt p(%))(2000)	0.034854	0.269049	0.91717	2.007376	3.730468
Mean(D pr-opt p(%))(5000)	1.26638	1.380156	2.203203	3.380276	3.392044
STD(D pr-opt p(%))(5000)	0.027355	0.482791	1.275962	1.009709	1.288211
Mean(D pr-opt p(%))(10,000)	0.054601	0.323809	2.098615	3.341647	4.424708
STD(D pr-opt p(%))(10,000)	0.032931	0.238552	1.660539	1.113023	2.767441
Mean(D ex r-o r ^b (%))(2000)	0.749206	0.751008	0.767109	0.70798	0.604736
STD(D ex r-o r(%))(2000)	0.005239	0.040443	0.149809	0.226238	0.356941
Mean(D ex r-o r(%))(5000)	0.012042	0.01719	0.052773	0.120857	0.126987
STD(D ex r-o r(%))(5000)	0.000516	0.009555	0.057981	0.064146	0.065791
Mean(D ex r-o r(%))(10,000)	3.44E-05	0.001585	0.064639	0.119724	0.210804
STD(D ex r-o r(%))(10,000)	3.2E-05	0.001816	0.074721	0.060797	0.308474

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 7.7: Accuracy of the tatonnement algorithm

Standard deviation	1	11	21	31	41
Mean(D pr-opt p ^a (%))(2000)	0.482279	4.811901	7.450708	10.40315	10.18355
STD(D pr-opt p(%))(2000)	0.283737	1.590703	6.710295	4.71741	7.864455
Mean(D pr-opt p(%))(5000)	0.498971	4.052615	5.589561	10.55027	7.188381
STD(D pr-opt p(%))(5000)	0.334517	1.806168	2.720864	5.998122	7.028261
Mean(D pr-opt p(%))(10,000)	0.435673	3.024379	5.131065	6.739073	6.138962
STD(D pr-opt p(%))(10,000)	0.305311	1.133411	3.234801	4.504129	5.2444
Mean(D ex r-o r ^b (%))(2000)	0.002533	0.237137	0.826589	1.252055	1.331634
STD(D ex r-o r(%))(2000)	0.002356	0.140808	1.083411	0.952839	1.889347
Mean(D ex r-o r(%))(5000)	0.00301	0.173088	0.339848	1.248374	0.795701
STD(D ex r-o r(%))(5000)	0.003703	0.137481	0.286312	1.190789	1.269215
Mean(D ex r-o r(%))(10,000)	0.002385	0.103254	0.304522	0.575972	0.510685
STD(D ex r-o r(%))(10,000)	0.003571	0.046	0.32051	0.618626	0.873158

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 7.8: Accuracy of the heuristic

Standard deviation	1	11	21	31	41
Mean(D pr-opt p ^a (%))(2000)	0.048485	0.483078	0.865686	0.93888	1.55489
STD(D pr-opt p(%))(2000)	0.035611	0.281482	0.705084	1.125651	0.936511
Mean(D pr-opt p(%))(5000)	0.019479	0.238601	0.531142	0.451335	1.285791
STD(D pr-opt p(%))(5000)	0.009289	0.097713	0.262493	0.248246	0.593023
Mean(D pr-opt p(%))(10,000)	0.011718	0.116695	0.251371	0.669098	0.790115
STD(D pr-opt p(%))(10,000)	0.008005	0.10144	0.173874	0.192253	0.44517
Mean(D ex r-o r ^b (%))(2000)	2.82E-05	0.00259	0.010715	0.016294	0.028307
STD(D ex r-o r(%))(2000)	3.43E-05	0.002553	0.019176	0.031884	0.031625
Mean(D ex r-o r(%))(5000)	4.74E-06	0.000566	0.002962	0.002223	0.017266
STD(D ex r-o r(%))(5000)	3.75E-06	0.000421	0.002666	0.001958	0.013144
Mean(D ex r-o r(%))(10,000)	1.93E-06	0.000202	0.000945	0.003932	0.007187
STD(D ex r-o r(%))(10,000)	1.83E-06	0.000288	0.001019	0.001916	0.00604

^aDistance price-optimal price

^bDistance expected revenue-optimal revenue

Table 7.9: Accuracy of the heuristic with \mathbf{A} known

This is particularly visible in Figure 7-7, which represents the accuracy of the price returned when the standard deviation is 11 for the three algorithms. It is clear that the geometric convergence of the tatonnement algorithm is much faster. The heuristic has a better estimate at the beginning but a slower convergence. Overall, this confirms our theoretical results.

Sensitivity to the number of products

We saw that knowing vector \mathbf{A} was enhancing significantly the performance of the heuristic in the two product case. We now check if the same performance enhancement occurs with a higher number of products. As in Chapter 5, dim is the number of products and the demand is of the following form

$$\mathbf{D}(\mathbf{P}) = \mathbf{A} - \mathbf{B} \cdot \mathbf{P} + \epsilon \text{ with,}$$

$$A = \begin{bmatrix} 5000 \\ 5000 \\ \dots \\ 5000 \end{bmatrix}, B = \begin{bmatrix} dim - 1 & -0.2 & -0.2 & \dots & -0.2 \\ -0.2 & dim - 1 & \dots & -0.2 & -0.2 \\ & & \dots & & \\ -0.2 & -0.2 & \dots & -0.2 & dim - 1 \end{bmatrix} \text{ and } P = \begin{bmatrix} P_1 \\ P_2 \\ \dots \\ P_{dim} \end{bmatrix}.$$

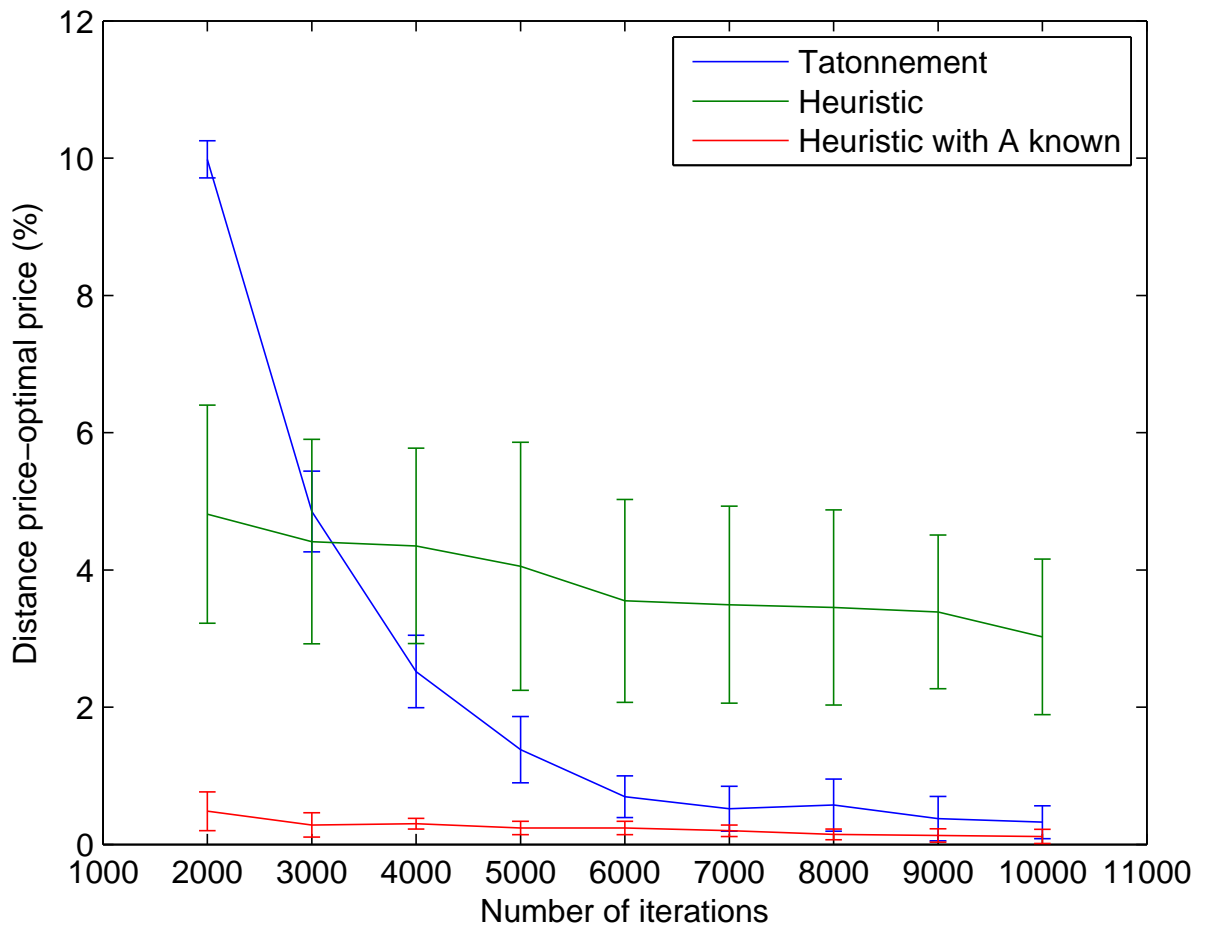


Figure 7-7: Comparison Heuristic-Tatonnement algorithm

Num. products	10	15	30	40	50
HEURISTIC					
Mean(Running time)	2.268708	3.872018	18.28688	41.27315	88.49795
STD(Running time)	0.006075	0.010141	0.275396	0.582921	9.450284
Mean(dist. price-opt \mathbf{p})(%)	0.224526	1.067138	0.066204	0.035095	0.023785
STD(dist. price-opt \mathbf{p})(%)	0.052457	0.333036	0.008484	0.007171	0.00309
HEURISTIC (A KNOWN)					
Mean(Running time)	2.279288	3.776987	17.00401	38.66978	78.66963
STD(Running Time)	0.240355	0.028699	0.566754	1.000758	1.865845
Mean(dist. price-opt \mathbf{p})(%)	0.016265	0.024463	0.022395	0.024676	0.021732
STD(dist. price-opt \mathbf{p})(%)	0.005652	0.002858	0.005514	0.004463	0.002507

Table 7.10: Sensitivity of the heuristic to the number of products

As before, all the noises follow a Normal distribution with mean 0 and standard deviation 10. We take

$$dim = 10, 15, 20, 25, 30, 35, 40, 45, 50.$$

For each value of dim , we run the two heuristics 1000 times and repeat the procedure 10 times. In Table 7.10 and Figure 7-8, we plot the running time and accuracy of the final price. We notice that knowing the value of the vector \mathbf{A} is of little importance for a big number of products but does make a significant difference for $dim \leq 25$. This is intuitive, because when the number of demand parameters increase, the interest of knowing one parameter beforehand decreases.

7.3 Insights

The simulations for the nonlinear demand models in the single product case confirmed our observations drawn from Chapter 3 on the importance of the length of the transient phase and the standard deviation of the noise of the demand function. The heuristic method was tested for the loglinear demand case and seemed numerically to check the consistency conditions by Lai and Wei [22]. The comparison with the tatonnement algorithm for the linear demand case was particularly insightful. The tatonnement algorithm has a better convergence rate but performs poorly at the beginning of the selling horizon. Moreover,

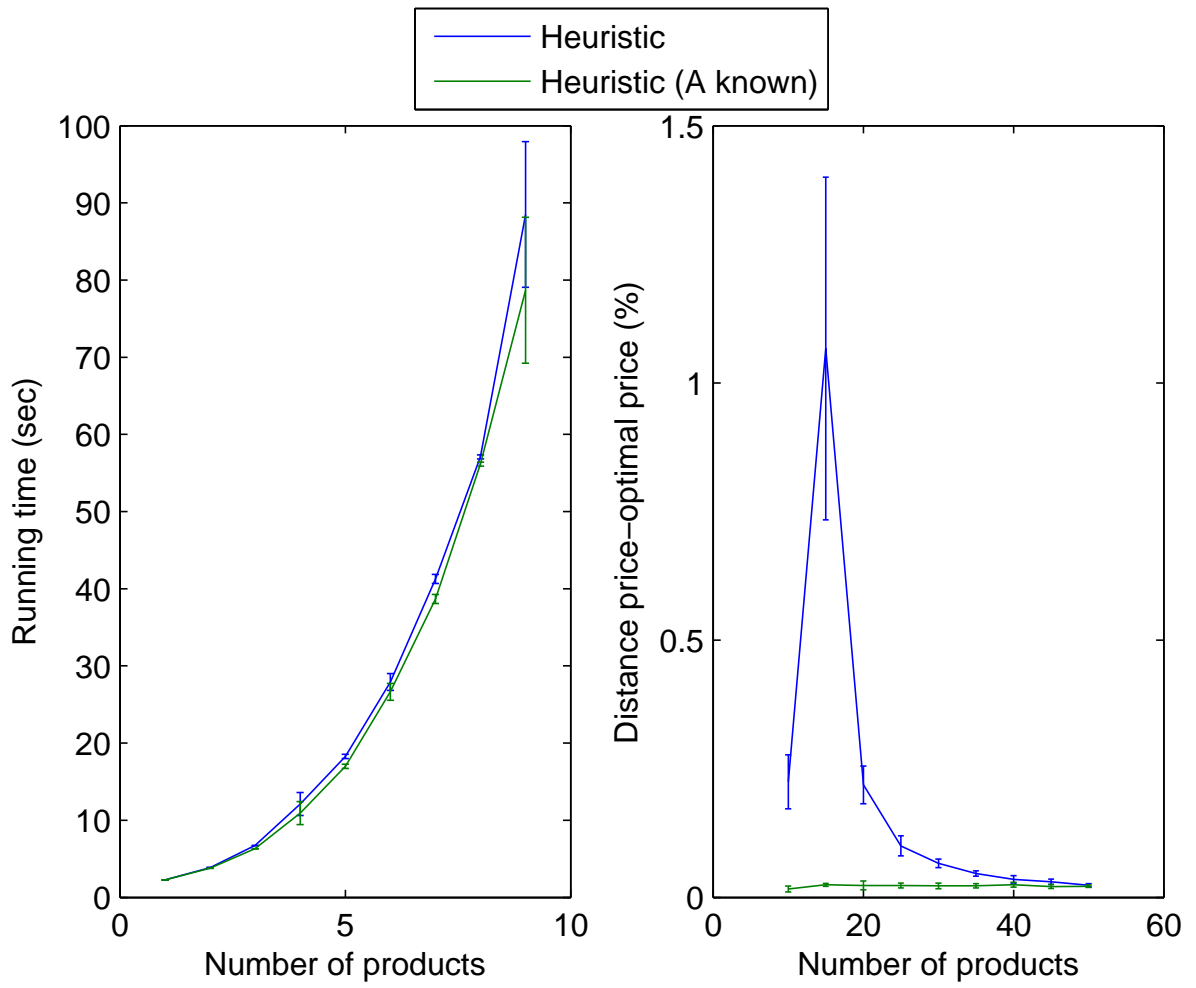


Figure 7-8: Sensitivity of the heuristic to the number of products

assuming that the demand intercept was known enhanced the performance of the heuristic. This was particularly significant for a small number of products.

Chapter 8

Conclusions

In this thesis, we developed a joint pricing and learning method for different parametric forms of the demand with unknown parameters. We focused on a monopoly setting.

In the first part, we considered a linear demand model for a single product. By using the framework of least-squares techniques, we presented a pricing algorithm that guarantees almost sure convergence of the estimated parameters to the true parameters. Moreover, this algorithm prices optimally in some sense.

In the second part, we showed how this algorithm can be extended to the multi-product setting assuming that the demand is affine. We established that a key parameter was the degree of diagonal dominance of the demand sensitivity matrix, and illustrated this point through simulations.

Finally, we considered some more general nonlinear parametric forms of demand for the single and the multi-product case. We introduced a heuristic for the multi product setting and compared its performance to the algorithm introduced in the second part for the linear demand case.

We could consider several extensions of this model as future research directions. These include:

1. Extending the algorithm to other parametric forms of demand or to nonparametric

forms using linear approximations.

2. Including capacity constraints for the multi-product case.
3. Analyzing the effect of competition on the pricing strategy in a Cournot setting (quantity competition).
4. Making the model more complex by adding purchase costs, as well as backloging costs.

Bibliography

- [1] Y. Aviv and A. Pazgal. A partially observed markov decision process for dynamic pricing. *Management science*, 51(9):1400-1416, 2005.
- [2] Y. Aviv and A. Pazgal. Pricing of short-life cycle products through active learning. *Working paper*, 2007.
- [3] A. Barto and R. Sutton. *Reinforced learning:an introduction*. Cambridge, MIT Press, 1998.
- [4] D. Bertsimas and G. Perakis. *Mathematical and Regional Models for Congestion Charging*, chapter Dynamic pricing: a learning approach. G. Sprinkler, Regionalwissenschaften, 2006.
- [5] O. Besbes and A. Zeevi. Dynamic pricing without knowing the demand function:Risk bounds and near optimal pricing algorithms. *Working paper*, 2006.
- [6] O. Besbes and A. Zeevi. Blind parametric revenue management: asymptotic normality of a joint pricing and learning approach. *Working paper*, 2006.
- [7] G. R Bitran and Caldentey. An overview of pricing models for revenue management. *Manufacturing and Service Operations Management*, 5:27, 2003.
- [8] G. R. Bitran, R. Caldentey, and S. V. Monschein. Coordinating clearance markdown sales of seasonal products in retail chains. *Operations Research*, 46(5), 1998.
- [9] A. Bjorck. *Numerical Methods for Least Squares Problems*. SIAM, 1996.

- [10] S. Boyd and M. S Lobo. Pricing and learning under uncertain demand. *Working paper*, 2003.
- [11] A. X. Carvalho and L. Puterman. Dynamic pricing and reinforcement learning. *Neural networks*, 2003.
- [12] A. X. Carvalho and L. Puterman. Learning and pricing in an internet environment with binomial demands. *Journal of Revenue and Pricing*, 3:320-336, 2005.
- [13] W.-C. Chiang, J. C. H. Chen, X. Xu. An overview of research on revenue management: current issues and future research. *International Journal of Revenue Management*, 1(1):97-128, 2007.
- [14] K. L. Chung. A Course in Probability Theory. *Academic Press*, Second Edition, 2000.
- [15] E. Cope. Revenue maximization in a stochastic environment. *Technical report*, Stanford University, 2002.
- [16] W. Elmagraby, A. Guleu and P. Keskinocak. Optimal markdown mechanisms in the presence of rational customers with multi-unit demand. *Working paper*, 2002.
- [17] W. Elmaghraby and P. Keskinocak. Dynamic pricing in the presence of inventory considerations: Research overview, current practice and future directions. *Management science*, 49(10):1287-1309, 2003.
- [18] V. K. Farias and B. Van Roy. Dynamic pricing with a prior on market response. *Working paper*, 2007.
- [19] G. A. Godfrey and W. B. Powell. An adaptive, distribution-free algorithm for the newsvendor problem with censored demands. *Management science*, 47:1101-1112, 2001.
- [20] S. Kachani, G. Perakis and C. Simon. Modeling the Transient Nature of Dynamic Pricing with Demand Learning in a Competitive Environment. *International Series in Operations Research and Management Science*, 102:223-267, 2007.

- [21] T. Lai, H. Robbins and Z. Wei. Strong consistency of least square estimates in multiple regression. *Proceedings of the National Academy of Science*, 75(7):3034-3036, 1978.
- [22] T. L. Lai and C. Z. Wei. Least squares estimates in stochastic regression models with applications to identification and Control of Dynamic Systems. *The Annal of Statistics*, 10(1):154-166, 1982.
- [23] C. E. Larson, L. J Olson and S. Sharma. Optimal demand policy when the demand distribution is not known. *Journal of economic Theory*, 101:281-300, 2001.
- [24] E. P Lazear. Retail prices and clearance sales. *The American Economic review*, 76(1):14-32, 1986.
- [25] K. Lin. Dynamic pricing with real time demand learning. *European Journal of Operations research*, 2005.
- [26] J. I MacGill and G Van Ryzin. *Revenue management:Research Overview and Prospects*. Transportation Science, 33(2):233-256, 1999.
- [27] *The Mc Kinsey quaterly*. The power of pricing. 1, 2003.
- [28] *The Mc Kinsey quaterly*. Pricing new products. 1, 2004.
- [29] H. Minc. *Nonnegative matrices*. Wiley, New York, 1988.
- [30] M. Nagarajan and G. Sasic. Stable Farsighted Coalitions in Competitive Markets. *Management Science*, 1:29-45, 2007.
- [31] K. Pak. and N. Piersma. Airline revenue management:An overview of OR Techniques 1982-2001. *Econometric Institute Report EI*, 2003.
- [32] N. C. Petruzzi and M. Dada. Dynamic pricing and inventory control with learning. *Naval Research Logistics*, 49(3):303-325, 2002.
- [33] J. Rice. *Mathematical Statistics and Data Analysis*. Second Edition, Duxbury Press, California, 1995.

- [34] A. Rustinici and A. Wolinsky. Learning about variable demand in the long run. *Journal of Economic Dynamics and Control*, 19:1283-1292, 1995.
- [35] N. Singh and X. Vives. Price and quantity competition in a duopoly. *Rand Journal of Economics*, 15(4), 1984.
- [36] S. A. Smith and D. D Achabal. Clearance pricing and inventory policies for retail chains. *Management Science*, 44(3), 1998.
- [37] K. Talluri and G. van Ryzin. The theory and practice of revenue management. *Kluwers Academic Publishers*, 2004.