

# POLITECNICO DI TORINO

Corso di Laurea Magistrale  
in Ingegneria Matematica



**Politecnico  
di Torino**

Tesi di Laurea Magistrale

## Robust Pricing Optimization Data-Driven Demand Forecasting

**Relatore**

Prof. Paolo Brandimarte

---

**Candidato**

Milena Maio

---

Anno Accademico 2023-2024

*"Mi domando se le stelle sono illuminate perchè ognuno possa un giorno trovare la sua"*

*Il Piccolo Principe*

*Alla mia splendida famiglia*

## Sommario

Nel contesto odierno, per raggiungere il successo in un mercato sempre più competitivo ed esigente come quello del marketing e del business aziendale è necessario adottare decisioni strategiche anche analizzando dati provenienti dall'esterno (non a caso si sente sempre più spesso parlare di approccio data-driven). Essi, infatti, costituiscono una fonte preziosa di informazione per conoscere il cliente e il suo comportamento d'acquisto, dunque, uno strumento molto utile alla stima dei parametri delle funzioni di domanda che modellano le vendite dei prodotti in relazione ai diversi fattori che le influenzano.

Uno dei driver principali che si andrà ad approfondire è il prezzo e, attraverso un esempio pratico, verrà analizzato il suo rapporto di inversa proporzionalità con la domanda lineare. I coefficienti saranno stimati attraverso una regressione ai minimi quadrati sulla base dei dati raccolti e l'incertezza verrà gestita utilizzando tecniche di ottimizzazione robusta. Spesso, però, non è possibile accedere ad un numero di dati sufficiente ad individuare una buona stima delle variabili non note, quindi, si può adottare un approccio differente, il cosiddetto *online learning*, ovvero un tipo di apprendimento step by step che consente di accumulare i dati e aggiornare il proprio modello predittivo man mano che le informazioni diventano disponibili.



# Indice

0.1	Introduzione . . . . .	1
0.2	Revenue Management e Marketing . . . . .	2
0.3	Il pricing e gli obiettivi aziendali . . . . .	4
<b>1</b>	<b>Pricing e Revenue Management</b>	<b>5</b>
1.1	Metodi di Pricing . . . . .	5
1.2	Elasticità della domanda . . . . .	6
1.3	Funzioni di domanda . . . . .	7
<b>2</b>	<b>Ottimizzazione</b>	<b>11</b>
2.1	Introduzione ai problemi di ottimizzazione . . . . .	11
2.2	Ottimizzazione convessa . . . . .	12
2.3	Analisi del caso lineare semplice non vincolato . . . . .	14
2.4	Problema di pricing deterministico nel continuo . . . . .	16
2.5	Problema di pricing deterministico misto intero . . . . .	17
<b>3</b>	<b>Approccio data-driven e ottimizzazione robusta</b>	<b>19</b>
3.1	Approccio data-driven . . . . .	19
3.2	Stima dei coefficienti: regressione lineare . . . . .	20
3.3	Ottimizzazione robusta . . . . .	21
3.4	Modelli di incertezza . . . . .	22
3.5	Programmazione robusta . . . . .	24
<b>4</b>	<b>Modelli implementati</b>	<b>25</b>
4.1	Costruzione dell'uncertainty set . . . . .	25

4.2	Problema di pricing robusto nel continuo con prodotto singolo . . . . .	28
4.3	Problema di pricing robusto misto intero due prodotti . . . . .	30
4.4	Problema di pricing robusto misto intero due prodotti - online learning . .	33
<b>Appendice: Concetti teorici</b>		<b>37</b>
4.5	Dimostrazione di concavità . . . . .	37
4.6	Condizione di ottimalità . . . . .	38
4.7	Stime coefficienti di regressione . . . . .	39
4.8	Formulazione duale del vincolo robusto . . . . .	42
<b>Appendice: Descrizione codici</b>		<b>45</b>

## 0.1 Introduzione

Nella presente trattazione si illustrerà una breve panoramica del ruolo del Revenue Management nel contesto del business aziendale e il suo legame con il marketing. Si procederà con la definizione dei metodi di pricing, dell'elasticità e delle curve di domanda che modellano l'andamento delle vendite sulla base del prezzo, con un focus particolare sulla funzione di domanda lineare ed esponenziale (capitolo 1).

A seguire saranno presentate le caratteristiche principali dei problemi di ottimizzazione approfondendo il concetto di convessità, in dettaglio sarà oggetto di analisi il problema deterministico di massimizzazione dei ricavi con variabili decisionali (vettore dei prezzi) nel continuo. Verranno mostrate anche le formulazioni delle varianti miste-intero che risultano più aderenti ad un contesto reale di applicazione del modello, in quanto, solitamente, non si ha un numero di candidati illimitati per l'assegnazione dei prezzi ai beni prodotti (capitolo 2).

Nelle applicazioni reali, tuttavia, la domanda è affetta da incertezza, dunque, non possiamo utilizzare i modelli deterministici, bensì ci serviamo dell'ottimizzazione robusta e dell'approccio data-driven in modo tale da stimare i parametri sulla base dei dati raccolti in precedenza e individuare una soluzione ottima anche nello scenario peggiore (capitolo 3).

Per concludere presentiamo i risultati numerici e alcune considerazioni dei modelli implementati in *Python*, (versione 3.9 supportata da Jupyter Notebook), e ottimizzati con il software *Gurobi* (versione 11.0.1) [4]. Per i primi due casi (singolo prodotto - prezzo continuo e due prodotti - prezzo intero) abbiamo caratterizzato l'insieme di incertezza con il budget of uncertainty,  $\Delta$  variabile, e riformulato il problema di ottimizzazione grazie alla dualità lagrangiana per un LP. Mentre nel terzo caso, apprendimento online, l'insieme di incertezza selezionato è del tipo *box uncertainty*, regione n-dimensionale definita dalla norma  $l_\infty$ , che va a restringersi con l'aumentare dei dati raccolti in quanto le stime puntuali dei coefficienti diventano gradualmente più affidabili. Gli ottimi robusti saranno valutati operando il confronto con le simulazioni Monte Carlo e la soluzione deterministica (capitolo 4).

## 0.2 Revenue Management e Marketing

Il presente elaborato si propone di approfondire i metodi per la risoluzione robusta di problemi di ottimizzazione di Revenue Management (RM), con particolare enfasi sulle strategie di prezzo applicate per prodotti e/o servizi. Innanzitutto, si procederà con la presentazione di cenni storici sul revenue management e, successivamente, si illustrerà il ruolo centrale del prezzo nell'economia odierna tramite strumenti come il marketing mix.

L'RM, noto anche come Yield Management, è il campo di studio di tecniche di gestione strategica volte alla massimizzazione dei ricavi di un'azienda. Nasce alla fine degli anni '70 per offrire un supporto decisionale alle compagnie aeree tradizionali che dovevano fronteggiare la concorrenza spudorata delle imprese low cost in seguito alla deregolamentazione del mercato del trasporto aereo. La nuova normativa, infatti, aveva l'obiettivo di disincentivare l'oligopolio permettendo l'entrata di nuovi concorrenti nel settore, essi effettuarono una competizione prevalentemente incentrata sui prezzi poiché il prodotto/servizio offerto era altamente sostituibile.

In generale il successo delle grandi imprese, in termini di profittabilità, dipende dalla loro abilità nell'individuare correttamente i bisogni e le esigenze dei consumatori realizzando prodotti che possano incontrare il loro gusto soggettivo, minimizzando i costi di produzione e massimizzando i ricavi derivanti dalla loro vendita.

In un'economia dinamica e competitiva, il prezzo rappresenta un fattore chiave per la sostenibilità delle imprese, infatti, le aziende che riescono a misurare correttamente il valore monetario dei propri prodotti conquistano quote di mercato e fidelizzano la clientela. Pertanto, il prezzo non rappresenta più una mera quantità numerica, ma diventa l'elemento che condiziona la percezione dei consumatori, la loro disponibilità all'acquisto e la valutazione del prodotto nel suo insieme.

Per implementare una strategia di pricing efficace e di successo è necessario considerare alcuni fattori importanti come la conoscenza approfondita dell'ambiente circostante in cui opera l'azienda, del settore e di tutti i suoi concorrenti, i costi di produzione e gli obiettivi aziendali di lungo periodo a cui è necessario allinearsi.

Il comitato decisionale ricorre al marketing mix come strumento essenziale per sviluppare i punti di forza e limitare le debolezze del proprio business. McCarthy negli anni



‘60 ha individuato le cosiddette 4P [5]:

1. **Product** → il prodotto rappresenta il core, l'elemento centrale intorno al quale ruotano tutte le altre leve strategiche. Si identifica con il bene offerto dall'azienda all'utente finale, delineandosi come il risultato di un processo di ideazione, lavorazione e produzione che sarà introdotto nel mercato e rivolto al segmento di clientela rilevante per l'impresa. Dal prodotto dipendono tutti gli altri fattori del Marketing Mix (prezzo, promozione e distribuzione), la sua gestione comprende numerosi aspetti, dall'ideazione del concept alla definizione delle sue caratteristiche tecniche. Ogni sua componente deve essere curata nei minimi dettagli per rispondere alle esigenze e alle aspettative del target di riferimento. Il prodotto, inoltre, attraversa un ciclo di vita caratterizzato da fase di lancio, crescita, maturità o declino (obsolescenza) e le strategie di marketing dovranno adeguarsi a ciascuna di esse ricorrendo a politiche coerenti con il posizionamento del prodotto nel suo ciclo di vita.
2. **Price** → il prezzo rappresenta il valore monetario che il consumatore è disposto a pagare per ottenere in cambio il prodotto/servizio e viene influenzato da molti fattori sia interni che esterni all'azienda. Tra i primi si possono riscontrare i volumi di vendita (quantità di acquisti prevista), i costi di produzione, l'intensità della concorrenza e le caratteristiche distintive del prodotto che potrebbero giustificare un premium price. Mentre tra i secondi si individuano le condizioni geo-politiche del territorio (guerre e pandemie) che possono determinare fluttuazioni significative dei prezzi e l'andamento di indici economico-finanziari come l'inflazione. Oltre ai fattori appena citati, nella definizione del prezzo si include anche l'implementazione di eventuali politiche commerciali volte a stimolare la domanda, tra queste, sconti o promozioni che riescono ad incentivare l'acquisto e metodi di pagamento dilazionati per modificare la percezione del prezzo del bene offerto.
3. **Place** → la distribuzione consiste nell'insieme delle attività create per rendere il prodotto accessibile al consumatore finale, definendo i canali e i luoghi attraverso i quali il prodotto sarà commercializzato. Oltre ai negozi fisici, che risultavano il canale distributivo per eccellenza fino a qualche anno fa, bisogna considerare anche gli shop online che ormai hanno monopolizzato il panorama distributivo

permettendo alle aziende di raggiungere un pubblico più ampio, abbattendo, spesso, i costi associati ai negozi fisici.

4. Promotion → la promozione si riferisce alle attività di comunicazione del prodotto al fine di accrescere la sua notorietà e il valore percepito dal consumatore. Rientrano in questo ambito la pubblicità, i media tradizionali (giornali, televisione e cartellonistica) e gli strumenti più digitali come i social network e siti e-commerce.

### 0.3 Il pricing e gli obiettivi aziendali

Come già evidenziato, la definizione di una strategia di prezzo efficace non può prescindere dagli obiettivi aziendali che si intendono perseguire, infatti, in base ad essi, le azioni da intraprendere avranno caratteristiche differenti. Di seguito si illustrano i tre principali macro-obiettivi aziendali e le relative decisioni in ambito pricing:

- **Massimizzare il profitto**, scopo prioritario di molte imprese è l'incremento della profittabilità, in questo caso il management è portato a definire un prezzo che consenta di aumentare le quantità vendute e di coprire i costi sostenuti per la produzione (fissi e variabili);
- **Gestire la concorrenza**, la differenziazione dai competitors è il motivo che spinge l'azienda a creare prodotti con caratteristiche qualitative superiori, garantendo un valore aggiunto (associato ad un premium price). Altrimenti si può competere sul prezzo, enfatizzando il vantaggio offerto in termini di convenienza d'acquisto, prevedendo di posizionare un prodotto ad un prezzo minore rispetto a quello deciso dai rivali;
- **Entrare in un nuovo mercato**, in questo scenario è fondamentale l'acquisto di quote di mercato, ma ciò implica sottrarre fette di ricavi alla concorrenza, quindi, il prezzo potrebbe essere l'elemento chiave per far conoscere il proprio prodotto ai clienti. Tuttavia, scatenare una guerra dei prezzi potrebbe non essere sostenibile nel lungo periodo in quanto avrebbe un impatto importante sulla redditività dell'azienda stessa.

# Capitolo 1

## Pricing e Revenue Management

### 1.1 Metodi di Pricing

Esistono diversi metodi per determinare il prezzo, ciascuno con i suoi punti di forza e debolezza, sono elencati di seguito:

1. *Cost-based*, politiche di pricing più comuni che riducono il margine di incertezza in quanto si basano sul calcolo dei costi sostenuti dall'azienda per la produzione del bene, sia fissi sia variabili (dati facilmente accessibili), ad essi verrà sommato un mark-up, ovvero il margine di profitto desiderato. Questo approccio risulta piuttosto semplicistico dal momento che non considera la disponibilità economica del target di riferimento e le decisioni adottate dalla concorrenza;
2. *Demand-based*, metodo che si focalizza sulla percezione del valore del prodotto da parte del cliente. Conoscendo la funzione di domanda che modella il comportamento d'acquisto, sulla base del prezzo massimo che i consumatori sono disposti a pagare (willingness to pay), l'impresa può selezionare il prezzo che le consente di massimizzare l'area sottesa alla curva associata ai ricavi (nel caso di domanda lineare è l'area di un rettangolo). La problematica risiede nell'individuazione della corretta curva di domanda, nell'avere le risorse necessarie per un'analisi di mercato volta alla segmentazione della clientela in base alla willingness to pay e nella disponibilità di dati storici affidabili per la stima dei parametri ignoti;

3. *Market-based*, piano di tariffazione realizzato considerando il prezzo fissato dai concorrenti, è la strategia più comune nel momento in cui l'azienda entra in un nuovo mercato e, quindi, ancora non conosce a fondo la domanda. Oppure questa tattica è spesso utilizzata nel settore delle commodities dove il prodotto è fortemente standardizzato e, pertanto, sostituibile. Tuttavia, questo modo di agire potrebbe facilmente comportare una guerra dei prezzi perchè anche gli altri attori coinvolti sarebbero costretti ad una lotta al ribasso, dunque, si otterrebbero conseguenze negative per tutto il settore in termini di redditività (competizione alla Bertrand).

In questa trattazione ci concentreremo sulla determinazione del prezzo in base alla domanda osservata e ai costi di produzione come valori soglia inferiori da non superare.

## 1.2 Elasticità della domanda

Per una grande quantità di prodotti o servizi si può affermare intuitivamente che la relazione tra domanda e prezzo sia negativa, difatti, solitamente se il prezzo aumenta meno persone saranno disposte a comprare quel bene soprattutto se non rientra tra gli acquisti di prima necessità.

Un indice normalizzato che modella questo legame è l'elasticità, ovvero il rapporto tra variazione relativa di domanda e variazione relativa di prezzo

$$\epsilon = \frac{\Delta d}{\Delta p} \frac{p}{d}$$

Considerando  $\Delta p \rightarrow 0$  si ottiene

$$\epsilon = - \frac{d'(p)p}{d(p)}$$

Il segno meno viene inserito per avere un valore di elasticità positivo, dal momento che la derivata spesso è negativa.

Se  $\epsilon > 1$  la domanda è molto elastica, quindi, molto sensibile alle perturbazioni di prezzo (prodotti comuni).

Se  $\epsilon < 1$  la domanda è inelastica cioè rigida rispetto alle variazioni di prezzo (beni di lusso o di prima necessità come i farmaci).

Se  $\epsilon = 1$  l'elasticità è unitaria.

## 1.3 Funzioni di domanda

Le curve di domanda o anche dette *price response functions* soddisfano tre proprietà principali:

- (i) Non negatività, la domanda deve essere sempre maggiore o uguale a zero per ogni valore di prezzo;
- (ii) Monotonia decrescente o strettamente decrescente, all'aumentare del prezzo assumiamo che la domanda diminuisca o, almeno, non sia soggetta ad un incremento.
- (iii) Continuità, assenza di salti, un'assunzione importante per poter trattare in modo più semplice il modello da un punto di vista matematico. Dobbiamo essere consapevoli, però, che la domanda ha una natura discreta perchè i prezzi non sono continui, solitamente variano fino all'ordine dei centesimi.

Le funzioni più utilizzate per modellare la relazione tra domanda e prezzo sono tre: Lineare, Esponenziale e Logistica/Sigmoide. Per ciascuna di esse si provvederà a fornire una descrizione generale e un ambito di applicazione. [3]

Si illustra la curva di **domanda lineare** che sarà oggetto del nostro modello di ottimizzazione robusta. Risulta la funzione più semplice ed è caratterizzata da due parametri: l'intercetta  $\alpha$  e il coefficiente angolare  $\beta$

$$d(p) = \alpha - \beta p \text{ con } \beta > 0$$

o equivalentemente

$$d(p) = \alpha + \beta p \text{ con } \beta < 0$$

l'intercetta indica quanti individui sono disposti a comprare il prodotto al prezzo minimo possibile che, solitamente, coincide con il costo di produzione del bene o di erogazione del servizio, mentre il coefficiente angolare rappresenta la variazione della domanda quando si verifica un incremento unitario del prezzo.

Per soddisfare la condizione di non negatività della domanda, bisogna fissare un valore soglia oltre il quale non ha senso analizzare il fenomeno, questo upper bound è

detto prezzo limite o di saturazione e si calcola ricavando il valore assunto da  $p$  quando la domanda è nulla.

$$\alpha + \beta p = 0 \implies p_{lim} = -\frac{\alpha}{\beta} \text{ con } \beta < 0$$

Pertanto, il prezzo deve essere selezionato nell'intervallo  $[p_{low}, p_{lim}]$ , ovvero  $[c_{prod}, -\frac{\alpha}{\beta}]$ .

L'elasticità della domanda lineare è variabile da elastica a inelastica a seconda se si è a destra o a sinistra di un prezzo soglia  $p_0$  (che vedremo successivamente essere uguale al punto di ottimo nella massimizzazione dei ricavi).

$$\epsilon_{lin} = -\frac{\beta p}{\alpha + \beta p}$$

Quindi, la domanda è inelastica nel momento in cui

$$\epsilon_{lin} < 1 \implies -\alpha - 2\beta p < 0 \implies p < -\frac{\alpha}{2\beta}$$

denotando il valore soglia  $p_0 = -\frac{\alpha}{2\beta}$ , la domanda è elastica quando  $p > p_0$ .

Se  $p \rightarrow p_{lim}$  l'elasticità tende a  $+\infty$ , ovvero la domanda diventa perfettamente elastica.

Questo tipo di funzione è molto semplice da trattare nei problemi di ottimizzazione (proviene da una willingness to pay distribuita come un'uniforme) e può essere applicata nei casi di monopolio in cui la domanda di un prodotto/servizio dipende solo dal prezzo. Purtroppo presenta dei limiti perchè suppone che stesse variazioni di prezzo implicino uguale riduzione della domanda. Ciò è irrealistico quando un concorrente sta offrendo lo stesso bene, o comunque un prodotto sostituto, ad un prezzo molto vicino a quello da noi fissato, nel concreto la variazione di domanda sarà notevolmente maggiore.

In secondo luogo si presenta la curva di **domanda esponenziale** definita come segue:

$$d(p) = De^{-bp}$$

$D > 0$  è il valore della domanda quando il prezzo è nullo, mentre  $b > 0$  è il parametro che determina quanto rapidamente si riduce la domanda all'aumentare del prezzo (la willingness to pay ha distribuzione esponenziale). In questo caso non c'è un upper bound per il prezzo, come nella situazione precedente, in quanto la funzione esponenziale è sempre positiva.

L'elasticità è lineare rispetto a  $p$  perchè

$$d'(p) = -Dbe^{-bp} \implies \epsilon_{exp} = \frac{Dbpe^{-bp}}{De^{-bp}} = bp$$

Dunque la domanda è rigida se  $p < \frac{1}{b}$  ed è sensibile al prezzo quando  $p > \frac{1}{b}$ , di conseguenza, sarà perfettamente elastica solo per  $p \rightarrow \infty$ .

Il punto di forza di questa funzione è la proprietà di linearità del logaritmo della domanda rispetto al prezzo,

$$\log(d(p)) = \log(D) - bp$$

pertanto, la stima dei parametri  $D$  e  $b$  può essere effettuata attraverso il metodo di regressione ai minimi quadrati e, successivamente, si applicano le dovute trasformazioni all'intercetta e al coefficiente angolare della retta trovata.

$$\alpha = \log(D) \implies D = e^\alpha \quad e \quad \beta = -b \implies b = -\beta.$$

In terzo luogo si descrive la curva di **domanda logistica** sulla base di una willingness to pay che è distribuita secondo una forma a campana, ovvero concentrata su un determinato prezzo (maggiore massa di probabilità) che coincide con la disponibilità a pagare più gettonata.

$$d(p) = \frac{Ce^{a-bp}}{1 + e^{a-bp}}$$

$C > 0$  è il parametro che riflette la grandezza del mercato di riferimento,  $b > 0$  è relativo alla sensibilità al prezzo, più è grande, maggiore sarà l'influenza delle variazioni di prezzo. La funzione presenta una forte inclinazione della curva in corrispondenza di  $p = \frac{a}{b}$ , solitamente il prezzo di mercato. Al contrario delle due precedenti (utili per analizzare localmente la risposta al prezzo), questa curva modella meglio il comportamento del consumatore in un mercato competitivo con grandi variazioni di prezzi, tuttavia risulta più difficile da gestire nei modelli di ottimizzazione data-driven perchè la stima dei parametri diventa complessa.





## Capitolo 2

# Ottimizzazione

### 2.1 Introduzione ai problemi di ottimizzazione

L'ottimizzazione è un potente strumento che può essere utilizzato in diversi contesti per ottenere soluzioni o proiezioni di scelte future, basti pensare alla finanza con modelli di minimizzazione del rischio, all'ingegneria per minimizzare l'energia dei sistemi e perfino al machine learning se si vuole stimare un set di parametri riducendo al minimo l'errore. Un problema di ottimizzazione ha l'obiettivo di trovare la soluzione "migliore" in base a determinati criteri. Migliore può voler dire massimizzare una funzione come il profitto/utilità o minimizzare una funzione di costo condizionatamente ad alcuni vincoli.

I problemi di ottimizzazione presentano tre caratteristiche principali:

1. **Funzione obiettivo** è la funzione che vogliamo massimizzare o minimizzare. Viene definita da un'espressione matematica in grado di mappare il fenomeno di interesse in una quantità numerica.
2. **Variabili decisionali** sono gli elementi da selezionare in un determinato dominio che fungono da leva per modificare il valore della funzione obiettivo (ridurlo o incrementarlo a seconda dell'orientamento dell'ottimizzazione). Possono essere di diverso tipo: reali se indicano, ad esempio, il prezzo di un prodotto, quantità interesse se parliamo di unità fisiche vendute o ancora booleane se specificano una condizione logica 1-0 come nel caso in cui bisogna decidere se attivare o meno una campagna pubblicitaria.

3. **Vincoli** determinano l'insieme delle alternative disponibili, noto anche come regione di ammissibilità. Sono le condizioni che devono essere soddisfatte dalle variabili decisionali per ottenere una soluzione praticabile nella realtà, essi, infatti, indicano limiti fisici (capacità produttiva o disponibilità di materie prime), restrizioni economiche (budget assegnato alle diverse business unit) o condizioni dirette per le variabili decisionali (ad esempio prezzo strettamente positivo).

Nella formulazione standard un problema di ottimizzazione si scrive nel seguente modo:

$$\begin{aligned} \min_{x \in X} f(x) \\ \text{s.t. } g_i(x) \leq 0, \quad i = 1, \dots, n \\ h_j(x) = 0, \quad j = 1, \dots, m \end{aligned} \tag{2.1}$$

I vincoli sono di due tipi, uguaglianza o disuguaglianza, ma i primi possono essere riformulati come combinazione di due vincoli di disuguaglianza aventi segno opposto.

## 2.2 Ottimizzazione convessa

Una sottoclasse particolare di problemi di ottimizzazione è costituita dai modelli convessi, hanno il vantaggio di poter essere risolti in maniera efficiente dal punto di vista computazionale in quanto consentono l'applicazione di alcuni concetti teorici di ottimalità globale, tuttavia devono soddisfare due condizioni:

- (i) Convessità della regione ammissibile
- (ii) Convessità della funzione obiettivo

Per il punto (i) un insieme si dice convesso se per ogni coppia di punti al suo interno anche il segmento che li collega è ancora incluso nell'insieme stesso.

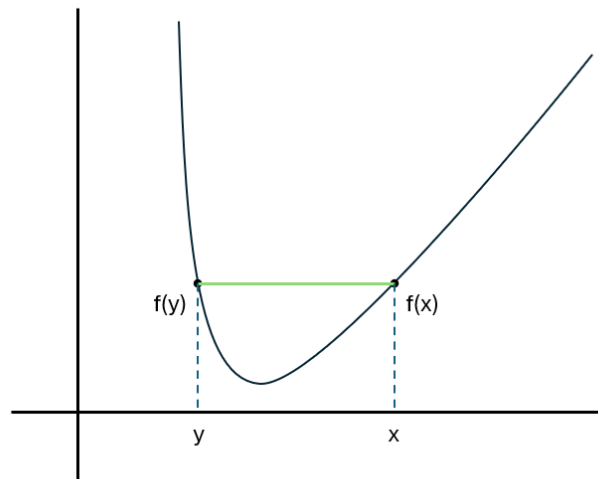
$$\forall x, y \in S \quad z = \lambda x + (1 - \lambda)y \in S \quad \forall \lambda \in [0, 1]$$

Per il punto (ii) una funzione si dice convessa se il suo dominio di appartenenza è un insieme convesso e se verifica la seguente proprietà

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \quad \forall x, y \in \text{dom}_f$$

Da un punto di vista geometrico ciò significa che la funzione valutata in una combinazione di punti del dominio si trova sempre al di sotto del segmento che unisce le valutazioni della funzione nei singoli punti.

Figura 2.1: Grafico di una funzione convessa, disuguaglianza da soddisfare



La formulazione standard dei problemi di ottimizzazione convessa è

$$\begin{aligned} \min_{x \in \mathbb{R}^n} f_0(x) \\ \text{s.t. } f_i(x) \leq 0, \quad i = 1, \dots, m \\ h_j(x) = 0, \quad j = 1, \dots, q \end{aligned} \quad (2.2)$$

dove i vincoli di disuguaglianza  $f_i(x)$  sono espressi da funzioni convesse e i vincoli di uguaglianza  $h_j(x)$  da funzioni affini.

Senza perdita di generalità un problema di massimo può essere facilmente convertito in uno di minimo

$$\max f(x) \implies -\min -f(x)$$

e risulta convesso se la funzione obiettivo è concava e la regione di ammissibilità è convessa.

In questa trattazione saranno utilizzati i problemi di ottimizzazione convessa LS (Least Squares) per la stima dei coefficienti di regressione lineare volti a delineare la funzione

di domanda in base ai dati disponibili, e QP (Quadratic Programming) per ottenere il prezzo ottimo che massimizza i ricavi dell'azienda.

I problemi non convessi, invece, sono, ad esempio, quelli che presentano variabili binarie o intere e necessitano di euristiche per la risoluzione.

### 2.3 Analisi del caso lineare semplice non vincolato

Assumendo di avere una domanda lineare, la massimizzazione dei ricavi si ottiene andando a selezionare il prezzo  $p$  tale che il valore della funzione  $r(p) = p \cdot d(p)$  sia ottimo

$$r(p) = p(\alpha + \beta p) = \alpha p + \beta p^2$$

La funzione è concava rispetto alla variabile  $p$  (Appendice 1- 4.5), quindi il punto di ottimo sarà il punto di massimo globale assumendo che i costi di produzione siano trascurabili o affondati ( $p$  non è soggetto a vincoli).

Si calcola  $p$  ponendo la derivata prima uguale a zero (Appendice 1-4.6)

$$\alpha + 2\beta p = 0 \implies p^* = -\frac{\alpha}{2\beta}$$

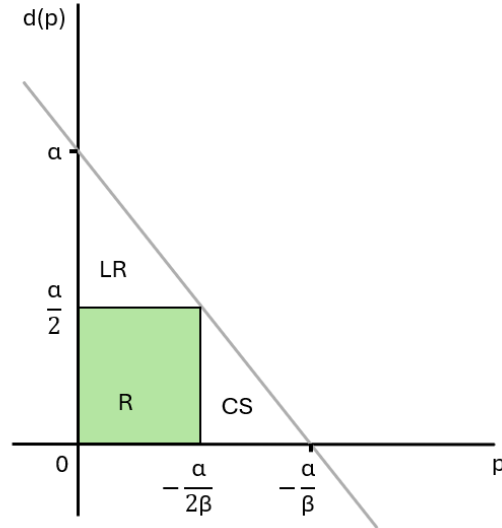
$$r(p^*) = -\frac{\alpha^2}{4\beta}$$

Il ricavo ottimo è, quindi, l'area del rettangolo con base  $p^* = -\frac{\alpha}{2\beta}$  e altezza  $d(p^*) = \alpha + \beta p = \frac{\alpha}{2}$ . La situazione ideale sarebbe quella in cui si ottiene l'area dell'intero triangolo, ciò implicherebbe l'uso di discriminazioni di prezzo in relazione ai diversi segmenti di clientela.

Infatti in figura 2.2, il triangolo superiore denotato con LR indica il *Lost Revenue* ovvero il ricavo perso perchè il prezzo era al di sopra della disponibilità a pagare di alcuni consumatori, mentre il triangolo inferiore targato CS rappresenta il *Customer Surplus* cioè il beneficio del cliente derivante dall'aver pagato un prezzo minore della propria willingness-to-pay, dunque, la quantità di denaro risparmiata.

Nella pratica si presentano alcuni limiti perchè i prezzi non sono quantità continue e per ragioni di marketing solo alcune configurazioni sono psicologicamente appetibili per

Figura 2.2: Funzione di domanda lineare (ricavo ottimo, lost revenue e surplus del consumatore)



il consumatore (ad esempio 9,99 euro invece di 10). Anche il tempo ha un ruolo fondamentale in quanto la sperimentazione deve essere effettuata in periodi circoscritti, non affetti da stagionalità, altrimenti si potrebbe incorrere nel malcontento dei consumatori o in un atteggiamento strategico dovuto al troppo dinamismo dei prezzi.

Infine un aspetto importante è la stima dei parametri, ovvero l'intercetta e il coefficiente angolare ( $\alpha$  e  $\beta$ ), che va effettuata sulla base di dati storici con la regressione lineare oppure, quando non si ha alcuna informazione, si adotta un tipo di apprendimento step by step detto *online learning*. Si procede, quindi, per esperimenti fissando diversi valori per la variabile prezzo e osservando la realizzazione della domanda ad esso associata, aggiornando il modello ad ogni iterazione. Naturalmente più è esteso il range di prezzi selezionati e minore sarà l'errore della stima, tuttavia è necessario trovare un compromesso tra l'accuratezza della funzione che modella la domanda e il costo di apprendimento che ne deriva.

## 2.4 Problema di pricing deterministico nel continuo

In questa sezione sarà formulato il problema di ottimizzazione di prezzo deterministico multi-prodotto, assumendo una funzione di domanda lineare. L'espressione della funzione obiettivo è una generalizzazione della funzione dei ricavi vista nel paragrafo precedente e, in aggiunta, saranno introdotti dei vincoli sulle variabili decisionali.[1]

$$d_j(p) = \alpha_j + \beta_j p_j \quad \forall j = 1, \dots, m$$

$m$  è il numero di prodotti di cui vogliamo stabilire il prezzo ottimo. Il problema di ottimizzazione risulta essere il seguente

$$\begin{aligned} \max_{p \in \mathbb{R}^m} \quad & \sum_{j=1}^m p_j (\alpha_j + \beta_j p_j) \\ \text{s.t.} \quad & l \leq p \leq u \end{aligned} \tag{2.3}$$

$l$  e  $u$  sono rispettivamente lower bound e upper bound dei prezzi dei diversi prodotti in esame, sono vettori appartenenti a  $\mathbb{R}_+^m$  in quanto devono essere compatibili con la dimensione della variabile decisionale  $p$ .

Si può selezionare come limite inferiore il costo di produzione unitario del bene in modo tale da consentire all'azienda di coprire almeno le spese operative legate alle materie prime utilizzate e alla manodopera necessaria al prodotto oppure i costi derivanti dall'erogazione del servizio richiesto. Il limite superiore, invece, è dato dal valore soglia oltre il quale la domanda diventa negativa  $u_j = -\frac{\alpha_j}{\beta_j} \quad \forall j = 1, \dots, m$ .

Per modellare una relazione negativa tra prezzi e domande associate, prendiamo

$$\beta_j < 0 \quad \forall j = 1, \dots, m$$

la funzione obiettivo risulta separabile e concava perchè somma positiva di termini concavi, dunque, il massimo globale può essere individuato facilmente nel rispetto dei vincoli imposti.

$$\alpha_j + 2\beta_j p_j = 0 \implies p_j^* = -\frac{\alpha_j}{2\beta_j} \quad \forall j = 1, \dots, m$$

$$\begin{cases} p_j^* = -\frac{\alpha_j}{2\beta_j} & \text{se } l_j \leq -\frac{\alpha_j}{2\beta_j} \\ p_j^* = l_j & \text{altrimenti} \end{cases} \quad \forall j = 1, \dots, m$$

## 2.5 Problema di pricing deterministico misto intero

Costituisce una leggera variazione della precedente formulazione, in quanto si aggiunge il vincolo intero (2.4) oppure la condizione di appartenenza ad un determinato set discreto di valori (2.5) per la variabile decisionale  $p$ .

$$\begin{aligned} \max_p \quad & \sum_{j=1}^m p_j(\alpha_j + \beta_j p_j) \\ \text{s.t.} \quad & l \leq p \leq u \\ & p_j \in \mathbb{Z} \quad \forall j = 1, \dots, m \end{aligned} \tag{2.4}$$

oppure

$$\begin{aligned} \max_p \quad & \sum_{j=1}^m p_j(\alpha_j + \beta_j p_j) \\ \text{s.t.} \quad & p_j \in S_j \quad \forall j = 1, \dots, m \end{aligned} \tag{2.5}$$

dove gli  $S_j$  sono insiemi discreti contenenti i prezzi candidati per ciascun tipo di prodotto.





## Capitolo 3

# Approccio data-driven e ottimizzazione robusta

### 3.1 Approccio data-driven

I problemi di ottimizzazione deterministici analizzati in precedenza si basano su un'assunzione alquanto discutibile: i parametri della funzione di domanda sono noti. Tuttavia, nella realtà i coefficienti della curva di domanda non sono disponibili a priori ma devono essere stimati sulla base dei dati storici disponibili[2].

Anche la tecnica di simulazione degli scenari e, di conseguenza, l'ottimizzazione stocastica del valore atteso dei ricavi determinerebbe un risultato distorto, in quanto non si conosce la vera distribuzione di probabilità della domanda.

Difatti il modello di ottimizzazione implementato si articolerà in due fasi distinte:

1. **Stima dei coefficienti** si individueranno i parametri propri della funzione di domanda che modella meglio la risposta al prezzo dei consumatori, riducendo il margine di errore al minimo. Le stime ottenute grazie agli algoritmi di machine learning saranno, ovviamente, affette dal rumore presente nei dati raccolti, dunque, verrà definito per ciascuna di esse un uncertainty set.
2. **Ottimizzazione Worst-case** Per costruire un modello stabile rispetto alle perturbazioni dei dati, in questo caso l'incertezza della domanda, adottiamo un approccio

robusto in modo tale da ottenere il vettore di prezzi ottimo che massimizzi i ricavi nel peggiore scenario possibile.

### 3.2 Stima dei coefficienti: regressione lineare

Avendo ipotizzato una curva di domanda lineare, l'algoritmo che ci permette di individuare le migliori stime per i parametri  $\alpha$  e  $\beta$  della funzione

$$d(p) = \alpha + \beta p$$

a partire dai dati delle coppie (prezzo-domanda)  $\{(p_i, d_i)_{i=1}^n\}$  raccolti in passato è la regressione lineare, utilizzando il metodo dei minimi quadrati (Ordinary Least Squares).

Il principio dei minimi quadrati non è nient'altro che la risoluzione di un problema di ottimizzazione convesso non vincolato

$$\min_{\alpha, \beta} \sum_{i=1}^n (d_i - d(p_i))^2 = \min_{\alpha, \beta} \sum_{i=1}^n (d_i - (\alpha + \beta p_i))^2$$

Dal punto di vista geometrico questo algoritmo consiste nella minimizzazione dei residui ovvero le distanze al quadrato tra il dato di domanda osservato realmente e il valore assunto dalla funzione in corrispondenza dello stesso prezzo (variabile indipendente). Più piccola sarà la somma degli errori e più accurata sarà la funzione che modella i dati disponibili.

Applicando le condizioni di ottimalità

$$\nabla f(\alpha, \beta) = 0 \implies \left( \frac{\delta}{\delta \alpha} f(\alpha, \beta), \frac{\delta}{\delta \beta} f(\alpha, \beta) \right) = (0, 0)$$

si ottengono le seguenti stime (Appendice 1 - 4.7):

$$\alpha = \hat{d} - \beta \hat{p} \quad \text{con} \quad \hat{d} = \sum_{i=1}^n \frac{d_i}{n} \quad \text{e} \quad \hat{p} = \sum_{i=1}^n \frac{p_i}{n}$$

$$\beta = \frac{\sum_{i=1}^n \frac{d_i p_i}{n} - \hat{d} \hat{p}}{\sum_{i=1}^n \frac{p_i^2}{n} - \hat{p}^2}$$

$$\implies \beta = \frac{Cov(d, p)}{Var(p)}$$

Per la costruzione dell'uncertainty set, si necessita degli intervalli di confidenza delle stime ottenute (Appendice 1 - 4.7). Naturalmente non conosciamo la vera deviazione standard dei coefficienti, dunque, si utilizzeranno le statistiche campionarie per la media e la varianza e la distribuzione di riferimento sarà la t di Student con n-2 gradi di libertà (ricordiamo che n indica la cardinalità del campione di dati osservati).

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} - t_{n-2, \frac{\delta}{2}} \cdot \begin{bmatrix} \sigma_\alpha \\ \sigma_\beta \end{bmatrix} \leq \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} \leq \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + t_{n-2, \frac{\delta}{2}} \cdot \begin{bmatrix} \sigma_\alpha \\ \sigma_\beta \end{bmatrix}$$

### 3.3 Ottimizzazione robusta

L'ottimizzazione robusta è un ramo della programmazione matematica che si occupa di trovare soluzioni ottimali per problemi decisionali caratterizzati da incertezza. A differenza dell'ottimizzazione classica, che presuppone la conoscenza dei parametri coinvolti nei vincoli e nella funzione obiettivo del problema, l'ottimizzazione robusta riesce a gestire l'incertezza adottando un approccio *worst-case*, ovvero costruendo un uncertainty set chiuso e limitato in cui i fattori di rischio sono circoscritti e ricercando la soluzione migliore nello scenario peggiore che possa verificarsi.

Pertanto, l'ottimizzazione risulta più conservativa ed è adatta a diversi contesti tra cui i problemi di natura fisica e ingegneristica come la minimizzazione dei costi legati all'erogazione di energia elettrica dai generatori disponibili, si va a considerare infatti il vincolo di soddisfacimento della domanda durante i picchi e non in media.

Nel caso del pricing otteniamo la seguente formulazione:

$$\begin{aligned} \max_p \left\{ \min_{\alpha, \beta \in U} \sum_{j=1}^m p_j (\alpha_j + \beta_j p_j) \right\} \\ \text{s.t. } l \leq p \leq u \end{aligned} \quad (3.1)$$

dove  $\alpha$  è il vettore delle intercette  $(\alpha_1, \alpha_2, \dots, \alpha_m)$  e  $\beta$  è il vettore dei coefficienti angolari  $(\beta_1, \beta_2, \dots, \beta_m)$  delle funzioni di domanda associate a ciascun tipo di prodotto  $j$ . Mentre  $U$  è l'uncertainty set relativo ai due parametri affetti da incertezza.

Lo scopo è quello di individuare la migliore decisione di prezzo che rimanga ottima per qualsiasi realizzazione di  $\alpha$  e  $\beta$  all'interno dei propri intervalli di incertezza.

Spesso, in letteratura, i problemi di ottimizzazione robusta sono riformulati attraverso variabili ausiliarie generando modelli equivalenti per facilitarne la risoluzione e ridurre il costo computazionale dei software utilizzati. Tuttavia, la difficoltà di individuazione della soluzione ottima non dipende solo da come viene imbastito il modello, ma anche dal tipo di regione di incertezza.

### 3.4 Modelli di incertezza

Tipicamente gli insiemi di incertezza si distinguono in cinque categorie principali[6]:

- **Insieme finito di scenari**, è il caso particolare dell'ottimizzazione con approccio data-driven in quanto si va ad individuare una soluzione robusta in base alla realizzazione dei fattori di rischio osservati precedentemente (dati di domanda e dei prezzi corrispondenti registrati nel corso dell'attività dell'azienda per ciascun tipo di prodotto immesso sul mercato). Purtroppo questa tecnica non è applicabile in tutti i casi, ad esempio per la stima della domanda relativa ad nuovo prodotto, i cui dati, naturalmente, ancora non sono disponibili, si rende necessario guardare ad una curva di domanda esterna per un bene già in commercio che presenti caratteristiche molto simili alle nostre.
- **Intervallo cartesiano** Detto anche *box uncertainty set*, è una regione di incertezza costruita come prodotto cartesiano degli intervalli di ciascun fattore di rischio/variabile casuale coinvolta nel problema di ottimizzazione. La caratterizzazione matematica può essere effettuata attraverso la norma  $l_\infty$

$$U_a = \{a \mid a = a_0 + P\xi, \|\xi\|_\infty \leq 1\}$$

dove per  $\alpha_0$  si intende il valore nominale del coefficiente e  $P$  è una matrice diagonale poichè non si considera alcun tipo di relazione tra i parametri incerti. Talvolta,

questo tipo di insieme potrebbe risultare troppo conservativo, in quanto si vanno a prendere in esame anche gli spigoli degli iper rettangoli ovvero gli scenari in cui tutti i fattori di rischio hanno realizzazioni estreme.

- **Insieme poliedrale** Insieme di incertezza identificato da un poliedro limitato, può essere visto come la ricopertura convessa di un set finito di punti, detti vertici del politopo.

$$U_a = \text{conv}(a_1, a_2, \dots, a_k)$$

Possiamo pensare a questo insieme come una generalizzazione della prima categoria di uncertainty set, in quanto consentirebbe di estendere l'ottimizzazione con approccio data-driven anche alle combinazioni dei vari scenari osservati e non più solo alle loro realizzazioni disgiunte.

- **Insieme ellissoide** Così come nel box uncertainty set avevamo utilizzato la norma  $l_\infty$ , questo tipo di insieme di incertezza è caratterizzato dalla norma  $l_2$ , quindi, in modo analogo scriviamo

$$U_a = \{a \mid a = a_0 + P\xi, \|\xi\|_2 \leq 1\}$$

In questo caso la matrice  $P$  non è necessariamente diagonale perchè si contemplan correlazioni tra i fattori di rischio. La forma dell'insieme risultante è un ellissoide che richiama alla mente la configurazione delle regioni di confidenza proprie della statistica multivariata.

- **Budget di incertezza** Costituisce una versione meno conservativa rispetto alla *box uncertainty*, in quanto riesce a smussare gli spigoli della 'scatola' n-dimensionale grazie all'introduzione di un vincolo in norma  $l_1$

$$U_a = \{a \mid a = a_0 + P\xi, \|\xi\|_\infty \leq \rho, \|\xi\|_1 \leq \delta\}$$

### 3.5 Programmazione robusta

La formulazione robusta per un problema di ottimizzazione standard con incertezza solo nella funzione obiettivo  $f$  causata dalla presenza del fattore di rischio  $\xi$  è la seguente

$$\begin{aligned} \min_x \left\{ \max_{\xi \in U_\xi} f(x, \xi) \right\} \\ \text{s.t. } g(x) \leq 0 \end{aligned} \quad (3.2)$$

Questo modello può essere riscritto in modo più semplice grazie all'introduzione di una variabile ausiliaria  $z$

$$\begin{aligned} \min_{x,z} z & \qquad \qquad \qquad \min_{x,z} z \\ \text{s.t. } g(x) \leq 0 & \qquad \qquad \implies \text{s.t. } g(x) \leq 0 \\ \max_{\xi \in U_\xi} f(x, \xi) \leq z & \qquad \qquad f(x, \xi) \leq z \quad \forall \xi \in U_\xi \end{aligned} \quad (3.3)$$

Si può notare che nei modelli di programmazione robusta i vincoli di uguaglianza non sono contemplati, questo deriva dal fatto che, se l'insieme di incertezza è continuo come il *box uncertainty set* e la condizione imposta deve essere soddisfatta per ciascun elemento appartenente ad esso, il numero di vincoli sarebbe infinito e, di conseguenza, la regione di ammissibilità sarebbe sempre vuota  $\implies$  nessun problema di ottimizzazione avrebbe soluzione.

Il modello di pricing (3.1), usando il trucco della variabile ausiliaria, diventa

$$\begin{aligned} \max_{p,z} z \\ \text{s.t. } l \leq p \leq u \\ \min_{\alpha, \beta \in U} \sum_{j=1}^m p_j (\alpha_j + \beta_j p_j) \geq z \end{aligned} \quad (3.4)$$

$$\begin{aligned} \max_{p,z} z \\ \implies \text{s.t. } l \leq p \leq u \\ \sum_{j=1}^m p_j (\alpha_j + \beta_j p_j) \geq z \quad \forall \alpha, \beta \in U \end{aligned} \quad (3.5)$$

A seconda di come viene caratterizzata l'incertezza e, quindi, l'insieme  $U$ , otteniamo risoluzioni differenti. Nel prossimo capitolo presenteremo i modelli implementati in *Python* utilizzando il solver *Gurobi*.

## Capitolo 4

# Modelli implementati

I dati per addestrare il modello di regressione lineare per la funzione di domanda sono stati generati indipendentemente scegliendo un valore specifico per  $\alpha_r$  e per  $\beta_r$ , fissando i limiti inferiore e superiore del prezzo  $(l, u)$  e aggiungendo del rumore  $\varepsilon$  avente distribuzione normale con media 0 e varianza  $\sigma_\varepsilon$ .

I prezzi sono stati campionati secondo una distribuzione uniforme nell'intervallo  $[l, u]$  e le domande sono state ottenute attraverso una trasformazione lineare dei prezzi.

$$d_i = \alpha_r + \beta_r p_i + \varepsilon_i \quad i = 1, \dots, n$$

con  $n$  cardinalità del campione generato.  $n$ , in genere, non assume un valore molto grande perchè si suppone che l'impresa abbia a disposizione pochi esperimenti, altrimenti le frequenti variazioni di prezzo potrebbero generare un malcontento dei consumatori o addirittura lo sviluppo di un atteggiamento strategico (aspettare che il prezzo scenda per effettuare l'acquisto).

Come accennato nei capitoli precedenti, il lower bound può coincidere con il costo di produzione del bene, mentre l'upper bound è il valore soglia oltre il quale la domanda diventa negativa  $(-\frac{\alpha_r}{\beta_r})$ .

### 4.1 Costruzione dell'uncertainty set

Assumiamo che le stime dei parametri affetti da incertezza possano variare tra un estremo inferiore e un estremo superiore, rispettivamente il lower bound e l'upper bound

dell'intervallo di confidenza ottenuto in precedenza.

Denotiamo, inoltre, con  $S^{RO}$  il numero di realizzazioni uniformemente campionate dal suddetto intervallo e definiamo alcune caratteristiche della regione di incertezza per ciascun prodotto  $j = 1, \dots, m$ , dato il set di scenari estratti  $\{\hat{\beta}_{js}\}_{s=1}^{S^{RO}}$  e  $\{\hat{\alpha}_{js}\}_{s=1}^{S^{RO}}$ :

- Media degli scenari

$$\bar{\beta}_j = \frac{\sum_{s=1}^{S^{RO}} \hat{\beta}_{js}}{S^{RO}} \quad e \quad \bar{\alpha}_j = \frac{\sum_{s=1}^{S^{RO}} \hat{\alpha}_{js}}{S^{RO}}$$

- Deviazione standard degli scenari

$$\tilde{\beta}_j = \sqrt{\frac{\sum_{s=1}^{S^{RO}} (\hat{\beta}_{js} - \bar{\beta}_j)^2}{S^{RO} - 1}} \quad e \quad \tilde{\alpha}_j = \sqrt{\frac{\sum_{s=1}^{S^{RO}} (\hat{\alpha}_{js} - \bar{\alpha}_j)^2}{S^{RO} - 1}}$$

- Minimo degli scenari

$$\hat{\beta}_j^{min} = \min_{s=1, \dots, S^{RO}} \{\hat{\beta}_{js}\} \quad e \quad \hat{\alpha}_j^{min} = \min_{s=1, \dots, S^{RO}} \{\hat{\alpha}_{js}\}$$

- Massimo degli scenari

$$\hat{\beta}_j^{max} = \max_{s=1, \dots, S^{RO}} \{\hat{\beta}_{js}\} \quad e \quad \hat{\alpha}_j^{max} = \max_{s=1, \dots, S^{RO}} \{\hat{\alpha}_{js}\}$$

- Budget di incertezza  $\Delta$

Ora possiamo definire l'uncertainty set  $U_\Delta$  nel seguente modo:

$$U_\Delta = \{[\alpha, \beta]^T \in \mathbb{R}^{2m} : \max_{j=1, \dots, m} \frac{|\beta_j - \bar{\beta}_j|}{\tilde{\beta}_j} \leq \Delta,$$

$$\max_{j=1, \dots, m} \frac{|\alpha_j - \bar{\alpha}_j|}{\tilde{\alpha}_j} \leq \Delta,$$

$$\hat{\beta}_j^{min} \leq \beta_j \leq \hat{\beta}_j^{max},$$

$$\hat{\alpha}_j^{min} \leq \alpha_j \leq \hat{\alpha}_j^{max}$$

$$\forall j = 1, \dots, m\}$$

Proviamo, adesso, a scomporre i vincoli non lineari per ottenere un insieme di incertezza più facile da gestire nella risoluzione del problema di pricing robusto.



Le condizioni con il massimo possono essere così trasformate

$$\begin{aligned} \max_{j=1,\dots,m} \frac{|\beta_j - \bar{\beta}_j|}{\tilde{\beta}_j} \leq \Delta &\implies \frac{|\beta_j - \bar{\beta}_j|}{\tilde{\beta}_j} \leq \Delta \quad \forall j = 1, \dots, m \\ &\implies |\beta_j - \bar{\beta}_j| \leq \Delta \cdot \tilde{\beta}_j \quad \forall j = 1, \dots, m \\ &\implies \beta_j - \bar{\beta}_j \leq \Delta \cdot \tilde{\beta}_j \quad e \quad \bar{\beta}_j - \beta_j \leq \Delta \cdot \tilde{\beta}_j \quad \forall j = 1, \dots, m \end{aligned}$$

Pertanto la regione di incertezza diventa

$$\begin{aligned} U_\Delta = \{[\alpha, \beta]^T \in \mathbb{R}^{2m} : &\beta_j - \bar{\beta}_j \leq \Delta \cdot \tilde{\beta}_j, \\ &\bar{\beta}_j - \beta_j \leq \Delta \cdot \tilde{\beta}_j, \\ &\alpha_j - \bar{\alpha}_j \leq \Delta \cdot \tilde{\alpha}_j, \\ &\bar{\alpha}_j - \alpha_j \leq \Delta \cdot \tilde{\alpha}_j, \\ &\hat{\beta}_j^{min} \leq \beta_j, \\ &\beta_j \leq \hat{\beta}_j^{max}, \\ &\hat{\alpha}_j^{min} \leq \alpha_j, \\ &\alpha_j \leq \hat{\alpha}_j^{max} \\ &\forall j = 1, \dots, m\} \end{aligned}$$

Esplicitiamo la tabella dei parametri usati nei modelli a seguire: prodotto singolo prezzo continuo (1p-c), due prodotti prezzo intero (2p-i), due prodotti prezzo intero online learning (2p-i-ol)

Parametri 1p-c	Valori	Parametri 2p-i	Valori	Parametri 2p-i-ol	Valori
seed	42	seed	42	seed	42
$\alpha_{real}$	100	$\alpha_{real}$	[200, 350]	$\alpha_{real}$	[200, 350]
$\beta_{real}$	-4	$\beta_{real}$	[-5, -7]	$\beta_{real}$	[-5, -7]
$\sigma_\varepsilon$	1	$\sigma_\varepsilon$	1	$\sigma_\varepsilon$	1
lower <sub>p</sub>	10	lower <sub>p</sub>	[10, 15]	lower <sub>p</sub>	[10, 15]
S <sup>RO</sup>	50	S <sup>RO</sup>	50	S <sup>RO</sup>	20
n	20	n	20	n <sub>initial</sub>	3
n <sub>scen</sub>	50	n <sub>scen</sub>	50	n <sub>iterations</sub>	10
conf-level	0.05	conf-level	0.05	conf-level	0.05

Per testare diversi livelli di conservatività del modello robusto, risolviamo il problema di ottimizzazione con valori di  $\Delta$  differenti, misura fondamentale del budget di incertezza: selezioniamo un insieme di valori (10) corrispondenti agli estremi dei sottointervalli risultanti dalla divisione in parti uguali del range  $[\Delta_{min}, \Delta_{max}]$ , dove

$$\Delta_{min} = 0 \quad e \quad \Delta_{max} = \max\{\Delta_{max}^{\alpha}, \Delta_{max}^{\beta}\}$$

con

$$\Delta_{max}^{\alpha} = \max_{j=1, \dots, m} \left\{ \max \left\{ \frac{|\hat{\alpha}_j^{min} - \bar{\alpha}_j|}{\tilde{\alpha}_j}, \frac{|\hat{\alpha}_j^{max} - \bar{\alpha}_j|}{\tilde{\alpha}_j} \right\} \right\}$$

$$\Delta_{max}^{\beta} = \max_{j=1, \dots, m} \left\{ \max \left\{ \frac{|\hat{\beta}_j^{min} - \bar{\beta}_j|}{\tilde{\beta}_j}, \frac{|\hat{\beta}_j^{max} - \bar{\beta}_j|}{\tilde{\beta}_j} \right\} \right\}$$

## 4.2 Problema di pricing robusto nel continuo con prodotto singolo

$$\begin{aligned} & \max_{p, z} z \\ & \text{s.t. } l \leq p \leq u \\ & \sum_{j=1}^m p_j (\alpha_j + \beta_j p_j) \geq z \quad \forall \alpha, \beta \in U_{\Delta} \end{aligned} \tag{4.1}$$

In questo caso m numero di prodotti è pari a 1, dunque, anche il vettore dei prezzi è unidimensionale così come i coefficienti di regressione  $\alpha$  e  $\beta$ . Allora otteniamo

$$\begin{aligned} & \max_{p, z} z \\ & \text{s.t. } l \leq p \leq u \\ & p(\alpha + \beta p) \geq z \quad \forall \alpha, \beta \in U_{\Delta} \end{aligned} \tag{4.2}$$

Possiamo utilizzare la dualità per il vincolo robusto (Appendice 1 - 4.8) ottenendo

$$\begin{aligned} & \max_{p, z} z \\ & \text{s.t. } l \leq p \leq u \\ & -\gamma_1 \cdot (\bar{\beta} + \Delta \cdot \tilde{\beta}) + \gamma_2 \cdot (\bar{\beta} - \Delta \cdot \tilde{\beta}) \\ & -\mu_1 \cdot (\bar{\alpha} + \Delta \cdot \tilde{\alpha}) + \mu_2 \cdot (\bar{\alpha} - \Delta \cdot \tilde{\alpha}) \end{aligned}$$

$$+\lambda_1 \cdot \hat{\beta}^{min} - \lambda_2 \cdot \hat{\beta}^{max} + \theta_1 \cdot \hat{\alpha}^{min} - \theta_2 \cdot \hat{\alpha}^{max}) \geq z$$

$$p + \mu_1 - \mu_2 - \theta_1 + \theta_2 = 0$$

$$p^2 + \gamma_1 - \gamma_2 - \lambda_1 + \lambda_2 = 0$$

$$\mu_1, \mu_2, \theta_1, \theta_2, \gamma_1, \gamma_2, \lambda_1, \lambda_2 \geq 0$$

### Risultati numerici

Confrontiamo l'ottimo del modello robusto con quello del problema deterministico e validiamo l'approccio worst-case con una simulazione Monte Carlo, campionando uniformemente dall'insieme di incertezza definito sopra e risolvendo il problema come fosse deterministico per ogni scenario  $(\alpha_s, \beta_s)$ . Il valore della simulazione sarà la media delle funzioni obiettivo ottime per ciascuna replica.

Figura 4.1: Simulazione Monte Carlo prodotto singolo - prezzo continuo

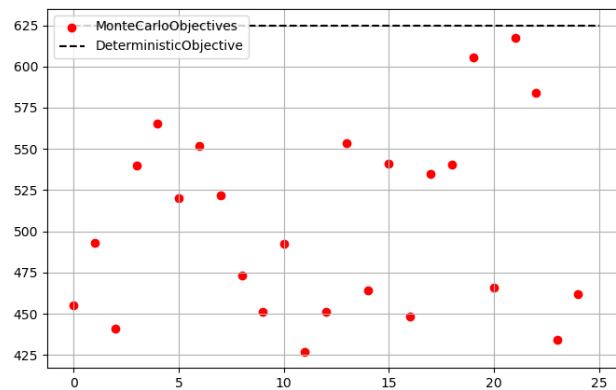


Tabella 4.1: Risultati ottimi 1p - c

Problema	Valore Ottimo
Deterministico	625
Robusto $\Delta_{min}$	618.98
Robusto $\Delta_{max}$	424.76
Simulazione Monte Carlo	505.42

Vediamo ora le variazioni in percentuale (VP) minima, media e massima dell'ottimo worst-case rispetto al valore della simulazione Monte Carlo

$$VP_{\Delta_i} = \frac{MC^* - RO_{\Delta_i}^*}{RO_{\Delta_i}^*}$$

$$\implies VP_{\Delta_{min}} = -0.18347, \quad VP_{avg} = 0.01617, \quad VP_{\Delta_{max}} = 0.18987$$

Pertanto, possiamo affermare che con l'incremento del budget di incertezza otteniamo un valore della funzione obiettivo più basso in quanto stiamo aumentando il livello di conservatività della soluzione.

Vogliamo osservare, inoltre, come reagisce l'ottimo robusto per i diversi valori di  $\Delta$  quando modifichiamo la varianza del rumore aggiunto durante la generazione dei dati ( $\sigma_\varepsilon$ )

Tabella 4.2: Risultati robusti al variare di  $\sigma_\varepsilon$  modello 1p - c

$\sigma_\varepsilon$	$\Delta_{min}$	$\Delta_{max}$	MC	$VP_{\Delta_{min}}$	$VP_{\Delta_{max}}$	$VP_{avg}$
0.5	621.99	512.76	558.75	-0.10167	0.08968	0.00437
1	618.98	424.76	505.42	-0.18347	0.18987	0.01617
2	613.01	249.53	405.40	-0.33867	0.62466	0.10048
3	607.07	74.29	306.85	-0.49453	3.13025	0.67821

Naturalmente, con l'aumentare dell'incertezza il valore della simulazione Monte Carlo diminuisce considerevolmente, mentre l'ottimo robusto presenta una riduzione più circoscritta, quindi una performance migliore (quando il *budget of uncertainty* risulta piccolo). Dunque, è necessario effettuare un trade-off tra il livello di conservatività desiderato e la massimizzazione dei ricavi robusti, questo compromesso sarà certamente influenzato dalla predisposizione o avversione al rischio dell'azienda.

### 4.3 Problema di pricing robusto misto intero due prodotti

Analogamente formuliamo il problema di ottimizzazione robusto con due prodotti e il prezzo appartenente ad un insieme discreto, gli interi tra il lower bound e l'upper bound fissati precedentemente

$$\begin{aligned}
 & \max_{p,z} z \\
 & \text{s.t. } l_j \leq p_j \leq u_j, p_j \in \mathbb{Z} \quad \forall j = 1, 2 \\
 & \sum_{j=1}^2 -\gamma_{j1} \cdot (\bar{\beta}_j + \Delta \cdot \tilde{\beta}_j) + \gamma_{j2} \cdot (\bar{\beta}_j - \Delta \cdot \tilde{\beta}_j) \\
 & -\mu_{j1} \cdot (\bar{\alpha}_j + \Delta \cdot \tilde{\alpha}_j) + \mu_{j2} \cdot (\bar{\alpha}_j - \Delta \cdot \tilde{\alpha}_j) \\
 & + \lambda_{j1} \cdot \hat{\beta}^{min} - \lambda_{j2} \cdot \hat{\beta}^{max} + \theta_{j1} \cdot \hat{\alpha}^{min} - \theta_{j2} \cdot \hat{\alpha}^{max} \geq z \\
 & \{p_j + \mu_{j1} - \mu_{j2} - \theta_{j1} + \theta_{j2} = 0 \\
 & p_j^2 + \gamma_{j1} - \gamma_{j2} - \lambda_{j1} + \lambda_{j2} = 0 \\
 & \mu_{j1}, \mu_{j2}, \theta_{j1}, \theta_{j2}, \gamma_{j1}, \gamma_{j2}, \lambda_{j1}, \lambda_{j2} \geq 0\} \quad \forall j = 1, 2
 \end{aligned}$$

### Risultati numerici

Confrontiamo la soluzione robusta con quella deterministica (caso benchmark perchè conosciamo la vera curva di domanda) e validiamo la tecnica worst-case con la simulazione Monte Carlo, campionando uniformemente dall'insieme di incertezza definito in precedenza e risolvendo il problema come fosse deterministico per ogni scenario  $\{\alpha_{js}, \beta_{js}\}_{j=1}^m$ . Il valore della simulazione sarà calcolato come la media delle soluzioni ottime per ciascuna simulazione.

Figura 4.2: Simulazione Monte Carlo due prodotti - prezzo intero

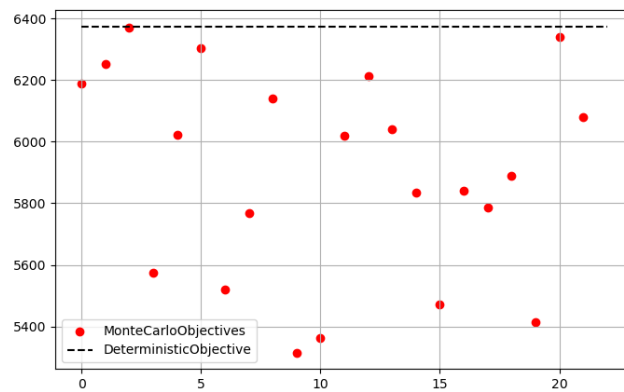


Tabella 4.3: Risultati ottimi 2p - i

<b>Problema</b>	<b>Valore Ottimo</b>
Deterministico	6375
Robusto $\Delta_{min}$	6376.62
Robusto $\Delta_{max}$	5213.39
Simulazione Monte Carlo	5897.85

Illustriamo di seguito le variazioni in percentuale (VP) minima, media e massima della soluzione robusta rispetto al valore Monte Carlo

$$\implies VP_{\Delta_{min}} = -0.07508, \quad VP_{avg} = 0.03449, \quad VP_{\Delta_{max}} = 0.13129$$

Studiamo, come nella sezione precedente, il comportamento dell'ottimo robusto e delle simulazioni Monte Carlo al variare dell'incertezza (perturbazioni della varianza del termine di rumore  $\varepsilon$ )

Tabella 4.4: Risultati robusti al variare di  $\sigma_\varepsilon$  modello 2p - i

$\sigma_\varepsilon$	$\Delta_{min}$	$\Delta_{max}$	MC	$VP_{\Delta_{min}}$	$VP_{\Delta_{max}}$	$VP_{avg}$
0.5	6375.81	5736.75	6123.23	-0.03962	0.06737	0.01725
1	6376.62	5213.39	5897.85	-0.07508	0.13129	0.03449
2	6378.25	4406.17	5443.98	-0.14648	0.23553	0.05620
3	6381.03	3814.04	5104.21	-0.20010	0.33827	0.08586

Dai risultati si può notare un comportamento anomalo per la soluzione robusta con minimo livello di conservatività, in quanto cresce con l'incertezza e supera anche il valore nominale. Questo è dovuto alla scelta dei prezzi in un set discreto mentre i coefficienti affetti da incertezza appartengono ad un intervallo continuo in cui è possibile trovare anche una stima più vantaggiosa di quella reale, mentre nel caso deterministico i valori sono fissati a priori. Le altre soluzioni robuste, invece, confermano il comportamento precedente, ovvero risultano più stabili all'aumentare dell'incertezza al contrario delle simulazioni Monte Carlo che mostrano un'ampia variabilità.

## 4.4 Problema di pricing robusto misto intero due prodotti - online learning

Nel caso dell'online learning la procedura dell'algoritmo risolutivo presenta caratteristiche differenti. L'uncertainty set non è più caratterizzato da un budget di incertezza ma da una regione a scatola (*box uncertainty*) ottenuta dal prodotto cartesiano degli intervalli di confidenza dei coefficienti di regressione  $\alpha$  e  $\beta$  stimati.

Questo approccio di apprendimento continuo prevede un modello di ottimizzazione robusto sequenziale che si aggiorna all'affiorare di nuovi dati, le fasi di learning e decision-making diventano interattive (bilanciando *exploration* ed *exploitation*). Esso necessita, dunque, di un'inizializzazione, ovvero la prima iterazione dell'algoritmo sarà eseguita con le poche informazioni a disposizione (nel nostro caso 3 record) da cui si otterrà il prezzo da sperimentare.

Una volta deciso il valore monetario del prodotto, si osserva la realizzazione della domanda, si aggiunge questa coppia di dati al campione iniziale e si reitera il processo. In questo modo il modello migliora step by step: più dati sono registrati e più accurata sarà la determinazione del prezzo e, quindi, ricavi maggiori.

Il modello di ottimizzazione robusta è lo stesso della sezione precedente eccetto per  $\Delta$  posto pari a 0.

$$\begin{aligned}
& \max_{p,z} z \\
& \text{s.t. } l_j \leq p_j \leq u_j, p_j \in \mathbb{Z} \quad \forall j = 1, 2 \\
& \sum_{j=1}^2 -\gamma_{j1} \cdot \bar{\beta}_j + \gamma_{j2} \cdot \bar{\beta}_j - \mu_{j1} \cdot \bar{\alpha}_j + \mu_{j2} \cdot \bar{\alpha}_j \\
& + \lambda_{j1} \cdot \hat{\beta}^{min} - \lambda_{j2} \cdot \hat{\beta}^{max} + \theta_{j1} \cdot \hat{\alpha}^{min} - \theta_{j2} \cdot \hat{\alpha}^{max} \geq z \\
& \{p_j + \mu_{j1} - \mu_{j2} - \theta_{j1} + \theta_{j2} = 0 \\
& p_j^2 + \gamma_{j1} - \gamma_{j2} - \lambda_{j1} + \lambda_{j2} = 0 \\
& \mu_{j1}, \mu_{j2}, \theta_{j1}, \theta_{j2}, \gamma_{j1}, \gamma_{j2}, \lambda_{j1}, \lambda_{j2} \geq 0\} \quad \forall j = 1, 2
\end{aligned}$$

### Risultati numerici

Osseviamo ora i risultati numerici del modello a confronto con la soluzione deterministica che fornisce un valore benchmark. Per semplicità abbiamo supposto un numero massimo di iterazioni oltre il quale l'apprendimento termina, si potrebbe alternativamete fissare una costante di tolleranza che indichi una soglia percentuale di miglioramento da rispettare, altrimenti l'algoritmo si ferma in quanto si è raggiunta una sorta di convergenza.

Figura 4.3: Sequenza ottimizzazione robusta online learning

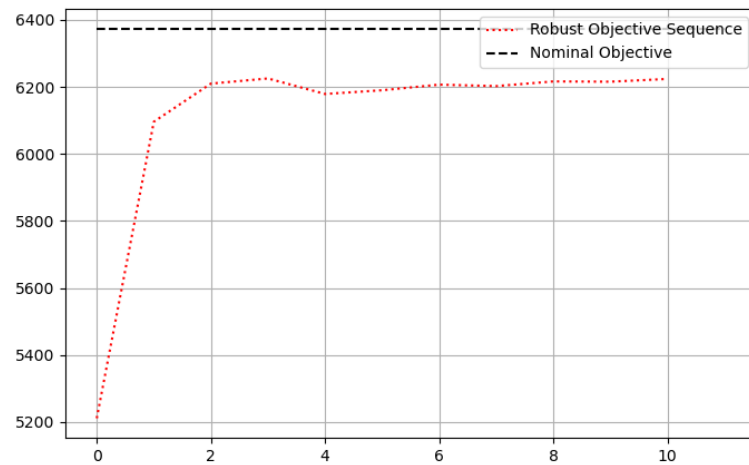


Tabella 4.5: Risultati ottimi 2p - i - ol

Problema	Valore Ottimo
Deterministico	6375
Robusto iterazione <sub>1</sub>	5210.98
Robusto iterazione <sub>10</sub>	6223.92

Illustriamo di seguito l'errore in percentuale (EP) minimo, medio e massimo della soluzione robusta rispetto al valore deterministico (DV)

$$EP_{ol_{iter_i}} = \frac{RO_{iter_i}^* - DV}{DV}$$

$$\implies EP_{ol_{iter_1}} = -0.18259, \quad EP_{ol_{avg}} = -0.04204, \quad EP_{ol_{iter_{10}}} = -0.02370$$



Dunque, con un numero di iterazioni massimo pari a 10 otteniamo il 2% di distanza dal valore deterministico.

Ora analizziamo il comportamento dell'apprendimento robusto al variare dell'incertezza dei dati, mantenendo costante il numero di iterazioni.

Tabella 4.6: Risultati robusti al variare di  $\sigma_\varepsilon$  modello 2p - i - ol

$\sigma_\varepsilon$	iterazione <sub>1</sub>	iterazione <sub>10</sub>	EP <sub>ol<sub>iter</sub>1</sub>	EP <sub>ol<sub>iter</sub>10</sub>	EP <sub>ol<sub>avg</sub></sub>
0.5	5712.24	6293.80	-0.10396	-0.01274	-0.02320
1	5210.98	6223.92	-0.18259	-0.02370	-0.02967
2	4487.88	6104.32	-0.29602	-0.04246	-0.07174
3	3904.32	5986.58	-0.38756	-0.06410	-0.09935

Come intuibile, la tabella mostra che l'errore cresce con l'aumentare della varianza del fattore di rumore  $\varepsilon$ , quindi l'algoritmo necessiterebbe di un numero maggiore di iterazioni per raggiungere una soluzione robusta migliore.

### Eventuali approfondimenti

I modelli di ottimizzazione robusta implementati e descritti in precedenza avrebbero applicazione anche nel caso di una curva di domanda non lineare come quella esponenziale. Difatti il metodo dei minimi quadrati potrebbe ancora essere utilizzato per la stima dei coefficienti che, però, assumeranno un nuovo significato.

$$\log(d(p)) = \log(D) - bp \implies \log(d(p)) = \alpha + \beta p$$

con

$$\alpha = \log(D) \quad e \quad \beta = -b$$

Inoltre, nella funzione di domanda lineare potrebbero essere introdotti termini che modellano una relazione con la politica di pricing adottata dalla concorrenza, dunque, il prezzo di un bene sostituto (stesse caratteristiche) o complementare (se si attua una strategia di bundling o tying con altri prodotti).

In alternativa, per prevedere i volumi di domanda relativi ad un prodotto o servizio ci si può avvalere di algoritmi di machine learning diversi dalla regressione lineare quali Decision Tree, Random Forest e Regressione Polinomiale che restituiscano le stime dei

coefficienti della price-response function. Le reti neurali, invece, non sono consigliate in un contesto in cui il campione di training è limitato, in quanto il numero degli esperimenti di pricing praticabili è circoscritto, ma per ottenere una discreta accuratezza è necessaria una grande mole di dati disponibili.

Per migliorare l'interpretabilità dei metodi di tipo ensemble, come il Random Forest, che offre maggiore robustezza e stabilità rispetto ai singoli alberi decisionali (grazie alle tecniche di bagging e boosting), si potrebbero applicare concetti di *Explainable AI*, un nuovo ramo dell'intelligenza artificiale che si è posto l'obiettivo di rendere più accessibili e intelligibili le procedure logiche e decisionali degli algoritmi.

# Appendice: Concetti teorici

## 4.5 Dimostrazione di concavità

Mostriamo come la funzione dei ricavi con domanda lineare sia concava rispetto alla variabile di prezzo, ricordiamo di seguito la sua espressione:

$$r(p) = \alpha p + \beta p^2 \quad \text{con } \beta < 0$$

In base alla definizione di concavità, dobbiamo verificare che

$$r(\lambda p_1 + (1 - \lambda)p_2) \geq \lambda r(p_1) + (1 - \lambda)r(p_2) \quad \forall p_1, p_2 \in \text{dom}_r$$

Esplicitiamo il termine a sinistra della disuguaglianza

$$r(\lambda p_1 + (1 - \lambda)p_2) = \alpha[\lambda p_1 + (1 - \lambda)p_2] + \beta[\lambda p_1 + (1 - \lambda)p_2]^2$$

Mentre a destra otteniamo

$$\begin{aligned} & \lambda(\alpha p_1 + \beta p_1^2) + (1 - \lambda)(\alpha p_2 + \beta p_2^2) \\ \implies & \alpha[\lambda p_1 + (1 - \lambda)p_2] + \beta[\lambda p_1 + (1 - \lambda)p_2]^2 \geq \lambda(\alpha p_1 + \beta p_1^2) + (1 - \lambda)(\alpha p_2 + \beta p_2^2) \end{aligned}$$

Possiamo elidere i termini uguali con  $\alpha$  in entrambi i membri

$$\beta[\lambda p_1 + (1 - \lambda)p_2]^2 \geq \lambda\beta p_1^2 + (1 - \lambda)\beta p_2^2$$

Semplifichiamo  $\beta$  ricordando che ha segno negativo

$$\begin{aligned} & [\lambda p_1 + (1 - \lambda)p_2]^2 \leq \lambda p_1^2 + (1 - \lambda)p_2^2 \\ \implies & \lambda^2 p_1^2 + (1 + \lambda^2 - 2\lambda)p_2^2 + 2\lambda(1 - \lambda)p_1 p_2 \leq \lambda p_1^2 + (1 - \lambda)p_2^2 \end{aligned}$$

$$\implies \lambda^2(p_1^2 + p_2^2 - 2p_1p_2) \leq \lambda(p_1^2 + p_2^2 - 2p_1p_2)$$

$$\implies (\lambda^2 - \lambda)(p_1 - p_2)^2 \leq 0$$

Il termine elevato al quadrato è sempre positivo quindi può essere semplificato

$$\lambda^2 - \lambda \leq 0$$

Da ciò otteniamo

$$0 \leq \lambda \leq 1$$

La disuguaglianza, dunque, è valida per ogni valore di  $\lambda$  dal momento che per definizione è un parametro appartenente all'intervallo chiuso  $[0, 1] \implies$  concavità verificata.

## 4.6 Condizione di ottimalità

Data  $f$  funzione concava e differenziabile al primo ordine,  $x$  è un punto di massimo se e solo se  $\nabla f(x) = 0$ . La dimostrazione di questa proposizione deriva parzialmente dalla condizione di ottimalità del primo ordine soddisfatta dalle funzioni concave

( $\rightarrow$ )  $x$  è punto di ottimo, supponiamo per assurdo che  $\nabla f(x)$  sia diverso da zero, allora possiamo facilmente individuare  $d = \nabla f(x)$  tale che

$$d^T \nabla f(x) = \|\nabla f(x)\|^2 > 0$$

direzione di salita. Quindi in ogni intorno di  $x$  di raggio arbitrariamente piccolo esiste  $y = x + \gamma \nabla f(x)$  con  $\gamma > 0$  tale che

$$f(y) = f(x + \gamma \nabla f(x)) \approx f(x) + \gamma \|\nabla f(x)\|^2 \implies f(y) > f(x)$$

Ma questo genera una contraddizione dal momento che  $x$  è un punto di massimo per ipotesi, quindi soddisfa  $f(x) \geq f(y) \forall y \in \mathbb{R}^n$ .

( $\leftarrow$ ) Una funzione concava differenziabile soddisfa la seguente disuguaglianza

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) \quad \forall y \in \mathbb{R}^n$$

Pertanto se  $\nabla f(x) = 0$  si ottiene

$$f(y) \leq f(x) \quad \forall y \in \mathbb{R}^n$$

ovvero la definizione di punto di massimo globale.

## 4.7 Stime coefficienti di regressione

Per individuare le stime dell'intercetta e il coefficiente angolare di regressione si utilizza la condizione di ottimalità ponendo il gradiente dei residui uguale a zero.

$$\begin{cases} \frac{\delta}{\delta\alpha}(\sum_{i=1}^n (d_i - \alpha - \beta p_i)^2) = 0 \\ \frac{\delta}{\delta\beta}(\sum_{i=1}^n (d_i - \alpha - \beta p_i)^2) = 0 \end{cases}$$

Ci concentriamo sulla prima equazione

$$\begin{aligned} \sum_{i=1}^n 2(d_i - \alpha - \beta p_i)(-1) &= 0 \\ \implies \sum_{i=1}^n \alpha + \beta p_i - d_i &= 0 \\ \implies \alpha n + \sum_{i=1}^n \beta p_i - d_i &= 0 \\ \implies \alpha &= \frac{1}{n} \sum_{i=1}^n d_i - \beta p_i \\ \implies \alpha &= \sum_{i=1}^n \frac{d_i}{n} - \beta \sum_{i=1}^n \frac{p_i}{n} \\ \implies \alpha &= \hat{d} - \beta \hat{p} \end{aligned}$$

Ora sostituiamo l'espressione di  $\alpha$  nella seconda equazione e ricaviamo  $\beta$

$$\begin{aligned} \sum_{i=1}^n 2(d_i - \alpha - \beta p_i)(-p_i) &= 0 \\ \implies \sum_{i=1}^n -d_i p_i + \left(\sum_{i=1}^n \frac{d_i}{n}\right) p_i - \beta \left(\sum_{i=1}^n \frac{p_i}{n}\right) p_i + \beta p_i^2 &= 0 \end{aligned}$$

$$\begin{aligned}
&\implies \sum_{i=1}^n -d_i p_i + \sum_{i=1}^n \frac{d_i}{n} \sum_{i=1}^n p_i - \beta \sum_{i=1}^n \frac{p_i}{n} \sum_{i=1}^n p_i + \beta \sum_{i=1}^n p_i^2 = 0 \\
&\implies \sum_{i=1}^n -d_i p_i + \sum_{i=1}^n d_i \hat{p} - \frac{\beta}{n} \left( \sum_{i=1}^n p_i \right)^2 + \beta \sum_{i=1}^n p_i^2 = 0 \\
&\implies \beta \left( \sum_{i=1}^n p_i^2 - \frac{(\sum_{i=1}^n p_i)^2}{n} \right) = \sum_{i=1}^n d_i (p_i - \hat{p}) \\
&\implies \beta = \frac{\sum_{i=1}^n d_i (p_i - \hat{p})}{\sum_{i=1}^n p_i^2 - \frac{(\sum_{i=1}^n p_i)^2}{n}} \\
&\implies \beta = \frac{n \left( \sum_{i=1}^n \frac{d_i p_i}{n} - \hat{d} \hat{p} \right)}{\sum_{i=1}^n p_i^2 - \frac{(\sum_{i=1}^n p_i)^2}{n}} \\
&\implies \beta = \frac{\sum_{i=1}^n \frac{d_i p_i}{n} - \hat{d} \hat{p}}{\sum_{i=1}^n \frac{p_i^2}{n} - \frac{(\sum_{i=1}^n p_i)^2}{n^2}} \\
&\implies \beta = \frac{\sum_{i=1}^n \frac{d_i p_i}{n} - \hat{d} \hat{p}}{\sum_{i=1}^n \frac{p_i^2}{n} - \hat{p}^2} \\
&\implies \beta = \frac{E[dp] - E[d]E[p]}{E[p^2] - E[p]^2} \\
&\implies \beta = \frac{Cov(d, p)}{Var(p)}
\end{aligned}$$

Alternativamente il vettore delle stime dei coefficienti può essere scritto come

$$b = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = (X^T X)^{-1} X^T y$$

con

$$X = \begin{bmatrix} 1 & p_1 \\ 1 & p_2 \\ \vdots & \vdots \\ 1 & p_n \end{bmatrix} \in \mathbb{R}^{n,2} \quad e \quad y = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} \in \mathbb{R}^n$$

Ora individuiamo l'intervallo di confidenza delle stime ottenute associato al livello di significatività  $\delta$  (i valori più comuni sono 1%, 5% e 10%).

Dal momento che le stime puntuali dei parametri provengono da una regressione ai minimi quadrati su un campione di osservazioni, la loro deviazione standard non è nota a priori. Dunque, la regione di confidenza sarà individuata attraverso il valore critico della distribuzione t di Student con n-2 gradi di libertà e una confidenza del  $(1 - \delta)\%$ .

Si calcola innanzitutto la somma dei residui al quadrato

$$RSS = \sum_{i=1}^n (d_i - (\alpha + \beta p_i))^2 = \|y - Xb\|^2$$

dove b è il vettore dei coefficienti precedentemente ricavati.

A seguire ricaviamo la matrice delle covarianze

$$\Sigma = MSE \cdot (X^T X)^{-1} \quad \text{con} \quad MSE = \frac{RSS}{n-2}$$

Per ottenere le varianze campionarie di ciascun coefficiente, selezioniamo solo gli elementi sulla diagonale della matrice  $\Sigma$ . Le deviazioni standard (*standard errors*) sono le radici quadrate delle varianze campionarie.

$$\sigma_b = \begin{bmatrix} \sigma_\alpha \\ \sigma_\beta \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{RSS}{n-2} \cdot ((X^T X)^{-1})_{1,1}} \\ \sqrt{\frac{RSS}{n-2} \cdot ((X^T X)^{-1})_{2,2}} \end{bmatrix}$$

Dunque, una volta calcolato anche il valore soglia della distribuzione t di Student con n-2 gradi di libertà

$$t_{critico} = t_{n-2, \frac{\delta}{2}}$$

a cui corrisponde un'area di probabilità cumulata pari a  $1 - \frac{\delta}{2}$  (data la simmetria), possiamo definire gli intervalli di confidenza:

$$\begin{aligned} b - t_{n-2, \frac{\delta}{2}} \cdot \sigma_b &\leq \hat{b} \leq b + t_{n-2, \frac{\delta}{2}} \cdot \sigma_b \\ \implies \begin{bmatrix} \alpha \\ \beta \end{bmatrix} - t_{n-2, \frac{\delta}{2}} \cdot \begin{bmatrix} \sigma_\alpha \\ \sigma_\beta \end{bmatrix} &\leq \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} \leq \begin{bmatrix} \alpha \\ \beta \end{bmatrix} + t_{n-2, \frac{\delta}{2}} \cdot \begin{bmatrix} \sigma_\alpha \\ \sigma_\beta \end{bmatrix} \end{aligned}$$

dove  $\alpha$  e  $\beta$  sono le stime puntuali ottenute dai dati, mentre  $\hat{\alpha}$  e  $\hat{\beta}$  sono gli stimatori delle medie.

## 4.8 Formulazione duale del vincolo robusto

$$\begin{aligned}
& \max_{p,z} z \\
& \text{s.t. } l \leq p \leq u \\
& \min_{\alpha, \beta \in U_\Delta} \sum_{j=1}^m p_j (\alpha_j + \beta_j p_j) \geq z
\end{aligned} \tag{4.3}$$

Concentriamoci sulla problema robusto

$$\min_{\alpha, \beta \in U_\Delta} \sum_{j=1}^m p_j (\alpha_j + \beta_j p_j)$$

Esplicitiamo i vincoli dell'uncertainty set  $U_\Delta$

$$\begin{aligned}
& \min_{\alpha, \beta} \sum_{j=1}^m p_j (\alpha_j + \beta_j p_j) \\
& \text{s.t. } \{ \beta_j - \bar{\beta}_j \leq \Delta \cdot \tilde{\beta}_j \\
& \quad \bar{\beta}_j - \beta_j \leq \Delta \cdot \tilde{\beta}_j \\
& \implies \alpha_j - \bar{\alpha}_j \leq \Delta \cdot \tilde{\alpha}_j \\
& \quad \bar{\alpha}_j - \alpha_j \leq \Delta \cdot \tilde{\alpha}_j \\
& \quad \hat{\beta}_j^{\min} \leq \beta_j, \quad \beta_j \leq \hat{\beta}_j^{\max} \\
& \quad \hat{\alpha}_j^{\min} \leq \alpha_j \quad \alpha_j \leq \hat{\alpha}_j^{\max} \quad \forall j = 1, \dots, m \}
\end{aligned} \tag{4.4}$$

Associamo a ciascuna condizione una variabile duale  $(\gamma_{j1}, \gamma_{j2}, \mu_{j1}, \mu_{j2}, \lambda_{j1}, \lambda_{j2}, \theta_{j1}, \theta_{j2})$  e scriviamo la funzione lagrangiana

$$\begin{aligned}
\mathcal{L}(\alpha, \beta, \gamma, \mu, \lambda, \theta) = & \sum_{j=1}^m (p_j (\alpha_j + \beta_j p_j) + \gamma_{j1} \cdot (\beta_j - \bar{\beta}_j - \Delta \cdot \tilde{\beta}_j) + \gamma_{j2} \cdot (\bar{\beta}_j - \beta_j - \Delta \cdot \tilde{\beta}_j) + \\
& \mu_{j1} \cdot (\alpha_j - \bar{\alpha}_j - \Delta \cdot \tilde{\alpha}_j) + \mu_{j2} \cdot (\bar{\alpha}_j - \alpha_j - \Delta \cdot \tilde{\alpha}_j) + \\
& \lambda_{j1} \cdot (\hat{\beta}_j^{\min} - \beta_j) + \lambda_{j2} \cdot (\beta_j - \hat{\beta}_j^{\max}) + \\
& \theta_{j1} \cdot (\hat{\alpha}_j^{\min} - \alpha_j) + \theta_{j2} \cdot (\alpha_j - \hat{\alpha}_j^{\max}))
\end{aligned}$$

Il minimo della lagrangiana rispetto a  $\alpha$  e  $\beta$ , data la linearità, si ottiene annullando i rispettivi coefficienti altrimenti il problema diventa *unbounded below* con l'ottimo che va



a  $-\infty$

$$\implies \begin{cases} p_j + \mu_{j1} - \mu_{j2} - \theta_{j1} + \theta_{j2} = 0 \\ p_j^2 + \gamma_{j1} - \gamma_{j2} - \lambda_{j1} + \lambda_{j2} = 0 \\ \mu_{j1}, \mu_{j2}, \theta_{j1}, \theta_{j2}, \gamma_{j1}, \gamma_{j2}, \lambda_{j1}, \lambda_{j2} \geq 0 \end{cases} \quad \forall j = 1, \dots, m$$

La funzione obiettivo del problema duale risulta

$$g(\gamma, \mu, \lambda, \theta) = \sum_{j=1}^m (-\gamma_{j1} \cdot (\bar{\beta}_j + \Delta \cdot \tilde{\beta}_j) + \gamma_{j2} \cdot (\bar{\beta}_j - \Delta \cdot \tilde{\beta}_j) - \mu_{j1} \cdot (\bar{\alpha}_j + \Delta \cdot \tilde{\alpha}_j) + \mu_{j2} \cdot (\bar{\alpha}_j - \Delta \cdot \tilde{\alpha}_j) + \lambda_{j1} \cdot \hat{\beta}_j^{\min} - \lambda_{j2} \cdot \hat{\beta}_j^{\max} + \theta_{j1} \cdot \hat{\alpha}_j^{\min} - \theta_{j2} \cdot \hat{\alpha}_j^{\max})$$

Secondo la dualità debole qualsiasi soluzione ammissibile del problema duale fornisce un lower bound per l'ottimo del primale, quindi, otteniamo la seguente configurazione finale:

$$\begin{aligned} & \max_{p,z} z \\ & \text{s.t. } l \leq p \leq u \\ & \sum_{j=1}^m (-\gamma_{j1} \cdot (\bar{\beta}_j + \Delta \cdot \tilde{\beta}_j) + \gamma_{j2} \cdot (\bar{\beta}_j - \Delta \cdot \tilde{\beta}_j) - \\ & \quad \mu_{j1} \cdot (\bar{\alpha}_j + \Delta \cdot \tilde{\alpha}_j) + \mu_{j2} \cdot (\bar{\alpha}_j - \Delta \cdot \tilde{\alpha}_j) + \\ & \quad \lambda_{j1} \cdot \hat{\beta}_j^{\min} - \lambda_{j2} \cdot \hat{\beta}_j^{\max} + \theta_{j1} \cdot \hat{\alpha}_j^{\min} - \theta_{j2} \cdot \hat{\alpha}_j^{\max}) \geq z \\ & \quad \{p_j + \mu_{j1} - \mu_{j2} - \theta_{j1} + \theta_{j2} = 0 \\ & \quad \quad p_j^2 + \gamma_{j1} - \gamma_{j2} - \lambda_{j1} + \lambda_{j2} = 0 \\ & \quad \quad \mu_{j1}, \mu_{j2}, \theta_{j1}, \theta_{j2}, \gamma_{j1}, \gamma_{j2}, \lambda_{j1}, \lambda_{j2} \geq 0\} \quad \forall j = 1, \dots, m \end{aligned}$$



# Appendice: Descrizione codici

In questo capitolo andremo a descrivere le caratteristiche principali dei codici implementati per i problemi deterministici e i relativi modelli di ottimizzazione robusta presentati nel Capitolo 4.

Partiamo dai modelli deterministico e robusto con prodotto singolo e prezzo nel continuo, i modelli con 2 prodotti e prezzo intero presentano le stesse caratteristiche (vedere file allegati).

Per definire un problema di ottimizzazione bisogna innanzitutto creare l'*objectModel*, in seguito andranno dichiarate le variabili decisionali con rispettivi lower e upper bound (con il comando *addVar*, se unidimensionali e *addVars* se n-dimensionali) e la funzione obiettivo specificando il senso dell'ottimizzazione (max o min). Per la risoluzione scriviamo l'istruzione *optimize()* e registriamo il valore ottimo nella variabile *obj\_nominal*.

```
[3]: try:
      # create the model
      model = grb.Model()
      # create decision variables = price
      p = model.addVar(ub=upper_p, lb=lower_p, vtype='C', name='p')
      # set objective function
      expr = (alpha_real + beta_real*p)*p
      model.setObjective(expr, sense=grb.GRB.MAXIMIZE)
      # solve the problem
      model.optimize()
      for v in model.getVars():
          print('%s %g' % (v.varName, v.x))
      print('Obj: %g' % model.objVal)
  except grb.GurobiError as e:
      print('Error code' + str(e))
  except AttributeError:
      print('Encountered an attribute error')
  obj_nominal = model.objVal
```

In modo analogo, nella versione robusta dichiariamo le variabili decisionali, quelle ottenute dalla formulazione del problema duale e aggiungiamo i vincoli attraverso il comando `addConstr`, infine risolviamo il problema per tutti i valori nel vettore  $\Delta$  del budget of uncertainty. Fissiamo il parametro di non convessità uguale a due perchè i vincoli di uguaglianza presentano termini quadratici, quindi non affini.

```
[14]: # ROBUST PROBLEM FORMULATION THROUGH STRONG DUALITY
rob_obj_val = []
for i in range(len(delta_test)):
    robust_model = grb.Model()
    z_r = robust_model.addVar(vtype='C', name='z_r')
    price_r = robust_model.addVar(lb=lower_p, ub=upper_p, vtype='C',
    ↪name='price_r')
    mu_r = robust_model.addVars(2, lb=0, vtype='C', name='mu_r')
    lambda_r = robust_model.addVars(2, lb=0, vtype='C', name='lambda_r')
    gamma_r = robust_model.addVars(2, lb=0, vtype='C', name='gamma_r')
    teta_r = robust_model.addVars(2, lb=0, vtype='C', name='teta_r')
    robust_model.setObjective(z_r, sense=grb.GRB.MAXIMIZE)
    robust_model.addConstr(mu_r[0]*alpha_min - mu_r[1]*alpha_max +
    ↪gamma_r[0]*beta_min - gamma_r[1]*beta_max -
        lambda_r[0]*(beta_mean + delta_test[i]*beta_std) +
    ↪lambda_r[1]*(beta_mean - delta_test[i]*beta_std) -
        teta_r[0]*(alpha_mean + delta_test[i]*alpha_std) +
    ↪teta_r[1]*(alpha_mean - delta_test[i]*alpha_std) >= z_r, name='c1')
    robust_model.addConstr(price_r - mu_r[0] + mu_r[1] + teta_r[0] - teta_r[1]
    ↪== 0, name='c2')
    robust_model.addConstr(price_r**2 - gamma_r[0] + gamma_r[1] + lambda_r[0] -
    ↪lambda_r[1] == 0, name='c3')
    # set non convexity parameter to 2 in such a way that gurobi can accept our
    ↪constraints formulation
    robust_model.params.NonConvex = 2
    robust_model.optimize()
    rob_obj_val.append(robust_model.objVal)
```

Le simulazioni Monte Carlo vengono implementate campionando uniformemente i parametri di incertezza dagli intervalli delle stime ottenuti con la regressione e risolvendo il problema come fosse deterministico per ogni scenario caratterizzato da una coppia  $(\alpha\_scen, \beta\_scen)$ .

```
[11]: # Monte Carlo simulations
np.random.seed(42)
n_scenarios = 50
alpha_scen = np.random.uniform(alpha_min, alpha_max, size=n_scenarios)
beta_scen = np.random.uniform(beta_min, beta_max, size=n_scenarios)
obj_val = []
for i in range(n_scenarios):
    model = grb.Model()
    p = model.addVar(ub=upper_p, lb=lower_p, vtype='C', name='p')
    expr = (alpha_scen[i] + beta_scen[i]*p)*p
    model.setObjective(expr, sense=grb.GRB.MAXIMIZE)
    model.optimize()
    obj_val.append(model.objVal)
```

Problema di pricing con 2 prodotti e prezzo intero (quando si dichiara la variabile decisionale si specifica  $vtype=I$ )

```
[4]: try:
# create the model
    model = grb.Model()
# create decision variables
    p = model.addVars(2, ub=[upper_p1, upper_p2], lb=[lower_p1, lower_p2],
    ↪vtype='I', name='p')
# set objective function
    expr = (alpha_real_1 + beta_real_1*p[0])*p[0]+(alpha_real_2 +
    ↪beta_real_2*p[1])*p[1]
    model.setObjective(expr, sense=grb.GRB.MAXIMIZE)
# solve the problem
    model.optimize()
    for v in model.getVars():
        print('%s %g' % (v.varName, v.x))
    print('Obj: %g' % model.objVal)
except grb.GurobiError as e:
    print('Error code' + str(e))
except AttributeError:
    print('Encountered an attribute error')
obj_nominal = model.objVal
```

```
[16]: # WORST CASE SCENARIO - REFORMULATION WITH STRONG DUALITY
rob_obj_val = []
for i in range(len(delta_test)):
    #create the robust model
    robust_model = grb.Model()
    #create decision variables
    z_r = robust_model.addVar(lb=0, vtype='C', name='z_r')
    price_r = robust_model.addVars(2, lb=[lower_p1, lower_p2], ub=[upper_p1,
↳upper_p2], vtype='I', name='price_r')
    mu_ra = robust_model.addVars(4, lb=0, vtype='C', name='mu_ra')
    mu_rb = robust_model.addVars(4, lb=0, vtype='C', name='mu_rb')
    gamma_ra = robust_model.addVars(4, lb=0, vtype='C', name='gamma_ra')
    gamma_rb = robust_model.addVars(4, lb=0, vtype='C', name='gamma_rb')
    #set objective function
    robust_model.setObjective(z_r, sense=grb.GRB.MAXIMIZE)
    #set constraints
    robust_model.addConstr(mu_ra[0]*alpha_min_1 - mu_ra[1]*alpha_max_1 +
↳mu_ra[2]*alpha_min_2 - mu_ra[3]*alpha_max_2 +
        mu_rb[0]*beta_min_1 - mu_rb[1]*beta_max_1 +
↳mu_rb[2]*beta_min_2 - mu_rb[3]*beta_max_2 -
        gamma_ra[0]*(alpha_mean_1 +
↳delta_test[i]*alpha_std_1) + gamma_ra[1]*(alpha_mean_1 -
↳delta_test[i]*alpha_std_1) -
        gamma_ra[2]*(alpha_mean_2 +
↳delta_test[i]*alpha_std_2) + gamma_ra[3]*(alpha_mean_2 -
↳delta_test[i]*alpha_std_2) -
        gamma_rb[0]*(beta_mean_1 + delta_test[i]*beta_std_1)
↳+ gamma_rb[1]*(beta_mean_1 - delta_test[i]*beta_std_1) -
        gamma_rb[2]*(beta_mean_2 + delta_test[i]*beta_std_2)
↳+ gamma_rb[3]*(beta_mean_2 - delta_test[i]*beta_std_2) >= z_r,name='c1')
    robust_model.addConstr(price_r[0] - mu_ra[0] + mu_ra[1] + gamma_ra[0] -
↳gamma_ra[1] == 0, name='c2')
    robust_model.addConstr(price_r[1] - mu_ra[2] + mu_ra[3] + gamma_ra[2] -
↳gamma_ra[3] == 0, name='c3')
    robust_model.addConstr(price_r[0]**2 - mu_rb[0] + mu_rb[1] + gamma_rb[0] -
↳gamma_rb[1] == 0, name='c4')
    robust_model.addConstr(price_r[1]**2 - mu_rb[2] + mu_rb[3] + gamma_rb[2] -
↳gamma_rb[3] == 0, name='c5')
    #solve the problem
    robust_model.params.NonConvex = 2
    robust_model.optimize()
    rob_obj_val.append(robust_model.objVal)
```

```
[13]: # Monte Carlo simulations
np.random.seed(42)
n_scenarios = 50
alpha_scen_1 = np.random.uniform(alpha_min_1, alpha_max_1, size=n_scenarios)
beta_scen_1 = np.random.uniform(beta_min_1, beta_max_1, size=n_scenarios)
alpha_scen_2 = np.random.uniform(alpha_min_2, alpha_max_2, size=n_scenarios)
beta_scen_2 = np.random.uniform(beta_min_2, beta_max_2, size=n_scenarios)
obj_val = []
for i in range(n_scenarios):
    model = grb.Model()
    p = model.addVars(2, ub=[upper_p1, upper_p2], lb=[lower_p1, lower_p2], u
    ↪vtype='I', name='p')
    expr = (alpha_scen_1[i] + beta_scen_1[i]*p[0])*p[0] + (alpha_scen_2[i] +
    ↪beta_scen_2[i]*p[1])*p[1]
    model.setObjective(expr, sense=grb.GRB.MAXIMIZE)
    model.optimize()
    obj_val.append(model.objVal)
```

Per il modello di online learning, invece, abbiamo una struttura leggermente diversa perchè bisogna aggiornare l'algoritmo di apprendimento della funzione di domanda ad ogni iterazione e risolvere il problema robusto con i nuovi dati osservati. Mostriamo la funzione di aggiornamento e di seguito il ciclo for per la risoluzione

```
[7]: def model_updating(p1, p2, d1, d2):
    global price_1, price_2, demand_1, demand_2
    price_1 = np.append(price_1, p1)
    price_2 = np.append(price_2, p2)
    demand_1 = np.append(demand_1, d1)
    demand_2 = np.append(demand_2, d2)
    data_1 = pd.DataFrame({'price': price_1, 'demand': demand_1})
    data_2 = pd.DataFrame({'price': price_2, 'demand': demand_2})
    X_1 = pd.DataFrame({'price': data_1['price'].copy()})
    y_1 = pd.DataFrame({'demand': data_1['demand'].copy()})
    X_1['constant'] = 1
    # LINEAR REGRESSION 1
    np.random.seed(42)
    returns_1 = np.linalg.lstsq(X_1, y_1, rcond=None)
    params_1 = returns_1[0]
    alpha_1 = params_1[1]
    beta_1 = params_1[0]
    sum_sq_residuals_1 = returns_1[1] # sum of squared residuals
    dof_1 = X_1.shape[0] - len(params_1) # degrees of freedom
    mse_1 = sum_sq_residuals_1 / dof_1 # mean squared error
    cov_1 = mse_1 * np.diagonal(np.linalg.inv(X_1.T @ X_1)) # covariance matrix
    se_1 = np.sqrt(cov_1) # standard errors
```





```

# basic statistics
alpha_min_1 = min(alpha_sim_1)
alpha_max_1 = max(alpha_sim_1)
beta_min_1 = min(beta_sim_1)
beta_max_1 = max(beta_sim_1)
alpha_min_2 = min(alpha_sim_2)
alpha_max_2 = max(alpha_sim_2)
beta_min_2 = min(beta_sim_2)
beta_max_2 = max(beta_sim_2)
alpha_mean_1 = sum(alpha_sim_1)/n_samples
alpha_mean_2 = sum(alpha_sim_2)/n_samples
beta_mean_1 = sum(beta_sim_1)/n_samples
beta_mean_2 = sum(beta_sim_2)/n_samples
alpha_std_1 = np.sqrt(sum((alpha_sim_1-alpha_mean_1)**2)/(n_samples-1))
alpha_std_2 = np.sqrt(sum((alpha_sim_2-alpha_mean_2)**2)/(n_samples-1))
beta_std_1 = np.sqrt(sum((beta_sim_1-beta_mean_1)**2)/(n_samples-1))
beta_std_2 = np.sqrt(sum((beta_sim_2-beta_mean_2)**2)/(n_samples-1))
delta_test = 0
try:
    # create the robust model
    robust_model = grb.Model()
    # create decision variables
    z_r = robust_model.addVar(lb=0, vtype='C', name='z_r')
    price_r = robust_model.addVars(2, lb=[lower_p1, lower_p2], ub=[upper_p1, u
↳upper_p2], vtype='I', name='price_r')
    mu_ra = robust_model.addVars(4, lb=0, vtype='C', name='mu_ra')
    mu_rb = robust_model.addVars(4, lb=0, vtype='C', name='mu_rb')
    gamma_ra = robust_model.addVars(4, lb=0, vtype='C', name='gamma_ra')
    gamma_rb = robust_model.addVars(4, lb=0, vtype='C', name='gamma_rb')
    # set objective function
    robust_model.setObjective(z_r, sense=grb.GRB.MAXIMIZE)
    # set constraints
    robust_model.addConstr(mu_ra[0]*alpha_min_1 - mu_ra[1]*alpha_max_1 + u
↳mu_ra[2]*alpha_min_2 - mu_ra[3]*alpha_max_2 +
        mu_rb[0]*beta_min_1 - mu_rb[1]*beta_max_1 + u
↳mu_rb[2]*beta_min_2 - mu_rb[3]*beta_max_2 -
        gamma_ra[0]*(alpha_mean_1 + u
↳delta_test*alpha_std_1) + gamma_ra[1]*(alpha_mean_1 - delta_test*alpha_std_1) -
        gamma_ra[2]*(alpha_mean_2 + u
↳delta_test*alpha_std_2) + gamma_ra[3]*(alpha_mean_2 - delta_test*alpha_std_2) -
        gamma_rb[0]*(beta_mean_1 + delta_test*beta_std_1) u
↳+ gamma_rb[1]*(beta_mean_1 - delta_test*beta_std_1) -
        gamma_rb[2]*(beta_mean_2 + delta_test*beta_std_2) u
↳+
        gamma_rb[3]*(beta_mean_2 - delta_test*beta_std_2) u
↳>= z_r, name='c1')

```

```

robust_model.addConstr(price_r[0] - mu_ra[0] + mu_ra[1] + gamma_ra[0] -
↳gamma_ra[1] == 0, name='c2')
robust_model.addConstr(price_r[1] - mu_ra[2] + mu_ra[3] + gamma_ra[2] -
↳gamma_ra[3] == 0, name='c3')
robust_model.addConstr(price_r[0]**2 - mu_rb[0] + mu_rb[1] + gamma_rb[0] -
↳gamma_rb[1] == 0, name='c4')
robust_model.addConstr(price_r[1]**2 - mu_rb[2] + mu_rb[3] + gamma_rb[2] -
↳gamma_rb[3] == 0, name='c5')
# solve the problem
robust_model.params.NonConvex = 2
robust_model.optimize()
except grb.GurobiError as e:
    print('Error code' + str(e))
except AttributeError:
    print('Encountered an attribute error')
object_val.append(robust_model.objVal)
current_price1 = robust_model.getVars()[1].x
current_price2 = robust_model.getVars()[2].x
obs_demand1 = alpha_real_1 + beta_real_1 * current_price1 + eps[i+n+1, 0]
obs_demand2 = alpha_real_2 + beta_real_2 * current_price2 + eps[i+n+1, 1]

```

# Elenco delle figure

2.1	Grafico di una funzione convessa, disuguaglianza da soddisfare . . . . .	13
2.2	Funzione di domanda lineare (ricavo ottimo, lost revenue e surplus del consumatore) . . . . .	15
4.1	Simulazione Monte Carlo prodotto singolo - prezzo continuo . . . . .	29
4.2	Simulazione Monte Carlo due prodotti - prezzo intero . . . . .	31
4.3	Sequenza ottimizzazione robusta online learning . . . . .	34



# Elenco delle tabelle

4.1	Risultati ottimi 1p - c . . . . .	29
4.2	Risultati robusti al variare di $\sigma_\varepsilon$ modello 1p - c . . . . .	30
4.3	Risultati ottimi 2p - i . . . . .	32
4.4	Risultati robusti al variare di $\sigma_\varepsilon$ modello 2p - i . . . . .	32
4.5	Risultati ottimi 2p - i - ol . . . . .	34
4.6	Risultati robusti al variare di $\sigma_\varepsilon$ modello 2p - i - ol . . . . .	35



# Bibliografia

- [1] Alvin Lim , Mahdi Hamzeei and Jiefeng Xu. *Robust price optimization of multiple products under interval uncertainties*. Journal of Revenue and Pricing Management, Chicago, 2022.
- [2] Dimitris Bertsimas and Phebe Vayanos. *Data-driven learning in dynamic pricing using adaptive optimization*. Operations Research, Cambridge, MA 02139, 2017.
- [3] Van Ryzin Garrett. Models of demand. *The Oxford Handbook of Pricing Management*, 01 2012.
- [4] Gurobi Optimization LLC. Gurobi optimizer reference manual. [https://gurobi.com/wp-content/plugins/hd\\_documentations/documentation/9.0/refman.pdf](https://gurobi.com/wp-content/plugins/hd_documentations/documentation/9.0/refman.pdf), 2020.
- [5] E. Jerome McCarthy. *Basic Marketing: A Managerial Approach*. Richard D. Irwin, Inc., Homewood, 1960.
- [6] Brandimarte Paolo. *An introduction to financial markets, a quantitative approach*. Wiley, Hoboken, NJ 07030, 2018.