

SAPIENZA
UNIVERSITÀ DI ROMA

Modellazione PNL

Modelli di programmazione non lineare

Dispensa didattica con modelli, esempi numerici svolti, casi di studio, implementazioni complete in Python/Gurobi e analisi di sensitività. Programmazione quadratica e non lineare: convessità e condizioni KKT, casi gestionali (portafoglio, prezzi, budget, localizzazione, code, supply chain), varianti quadratiche di modelli lineari e ottimizzazione per il machine learning (SVM, minimi quadrati).

Fabio Furini
fabiofurini.github.io
Dipartimento di Ingegneria informatica, automatica e gestionale — Sapienza Università di Roma

Versione online e materiale scaricabile:
fabiofurini.github.io/modellazione-pnl/
github.com/fabiofurini/modellazione-pnl

Anno accademico 2026–2027

Indice

I Strumenti **4**

1 Introduzione al corso **5**
1.1 Che cosa faremo . . . 5
1.2 Le classi di modelli e le sigle usate . . . 5
1.3 Mappa dei modelli . . . 6
1.4 Organizzazione del corso . . . 6
1.5 In sintesi . . . 6

2 Richiami: convessità, QP e condizioni KKT **7**
2.1 Convessità e programmazione quadratica . . . 8
2.2 Le condizioni di ottimalità: le condizioni KKT . . . 9
2.3 In sintesi . . . 10

3 Il solver per i modelli non lineari **11**
3.1 Vincoli funzionali e termini bilineari . . . 11
3.2 Tolleranze e scalatura . . . 12
3.3 I moltiplicatori nei QP convessi . . . 12
3.4 Quando i duali non ci sono . . . 12
3.5 In sintesi . . . 12

II Modelli deterministici non lineari **14**

4 Produzione con smoothing quadratico **15**
4.1 Che cosa cambia rispetto al modello lineare . . . 15
4.2 Il modello . . . 16
4.3 Esempio svolto a mano . . . 17
4.4 Implementazione . . . 17
4.5 Risultati . . . 17
4.6 Analisi di sensitività . . . 18
4.7 In sintesi . . . 19
4.8 Esercizi . . . 20

5 Supply chain con congestione e sostenibilità **21**
5.1 Motivazione gestionale . . . 21
5.2 Costruzione guidata del modello . . . 22
5.3 Esempio svolto a mano . . . 23

5.4 Caso di studio . . . 23
5.5 Risultati . . . 24
5.6 Analisi di sensitività: il prezzo della CO_2 . . . 25
5.7 In sintesi . . . 26
5.8 Esercizi . . . 26

6 Portafoglio di Markowitz **27**
6.1 Motivazione gestionale . . . 27
6.2 Costruzione guidata del modello . . . 28
6.3 Esempio svolto a mano . . . 29
6.4 Caso di studio . . . 29
6.5 Risultati . . . 30
6.6 Analisi di sensitività . . . 31
6.7 In sintesi . . . 32
6.8 Esercizi . . . 32

7 Pricing e revenue management **33**
7.1 Motivazione gestionale . . . 33
7.2 Costruzione guidata del modello . . . 34
7.3 Esempio svolto a mano . . . 35
7.4 Caso di studio e implementazione . . . 35
7.5 Versione multiprodotto . . . 37
7.6 In sintesi . . . 37
7.7 Esercizi . . . 37

8 Allocazione del budget pubblicitario **39**
8.1 Motivazione gestionale . . . 39
8.2 Costruzione guidata del modello . . . 40
8.3 Esempio svolto a mano . . . 41
8.4 Caso di studio e implementazione . . . 41
8.5 Risultati e verifica delle KKT . . . 42
8.6 Analisi di sensitività . . . 43
8.7 In sintesi . . . 43
8.8 Esercizi . . . 43

9 Localizzazione continua di un servizio **45**
9.1 Motivazione gestionale . . . 45
9.2 Costruzione guidata del modello . . . 46
9.3 Esempio svolto a mano . . . 47
9.4 Caso di studio . . . 47
9.5 Risultati . . . 48
9.6 Analisi di sensitività: la frontiera efficienza-equità . . . 49
9.7 In sintesi . . . 49
9.8 Esercizi . . . 50

10 Ricarica di veicoli elettrici con profilo regolare **51**
10.1 Che cosa cambia rispetto al modello lineare . . . 51
10.2 Il modello . . . 51
10.3 Esempio svolto a mano . . . 53
10.4 Implementazione . . . 53

10.5 Risultati . . . 53
10.6 Analisi di sensitività . . . 54
10.7 In sintesi . . . 55
10.8 Esercizi . . . 55

11 Capacità di servizio e tempi di attesa **57**
11.1 Motivazione gestionale . . . 57
11.2 Costruzione guidata del modello . . . 58
11.3 Esempio svolto a mano . . . 59
11.4 Caso di studio e risultati . . . 59
11.5 Analisi di sensitività: il prezzo di una promessa . . . 60
11.6 In sintesi . . . 61
11.7 Esercizi . . . 61

III Ottimizzazione e machine learning **62**

12 Support Vector Machine **63**
12.1 Motivazione gestionale . . . 63
12.2 Hard margin: la geometria . . . 64
12.3 Soft margin . . . 65
12.4 La formulazione duale e i support vector . . . 67
12.5 Kernel: frontiere non lineari da un QP convesso . . . 68
12.6 Classi sbilanciate e costi asimmetrici . . . 69
12.7 Support Vector Regression . . . 69
12.8 In sintesi . . . 70
12.9 Esercizi . . . 70

13 Regressione ai minimi quadrati **71**
13.1 Che cosa cambia rispetto al modello lineare . . . 71
13.2 Il modello . . . 72
13.3 Le KKT sono le equazioni normali . . . 73
13.4 Caso di studio: minimi quadrati contro scarti assoluti . . . 73
13.5 Previsione per il newsvendor: media più scorta o quantile . . . 74
13.6 Analisi di sensitività . . . 74
13.7 In sintesi . . . 75
13.8 Esercizi . . . 76

IV Organizzazione del corso **77**

14 Organizzazione del corso **78**
14.1 Percorso essenziale in quattro laboratori . . . 78
14.2 Errori ricorrenti da evitare . . . 78
14.3 Materiali del corso . . . 79

Parte I

Strumenti

Capitolo 1

Introduzione al corso

Prerequisiti

- Dal modulo *Modellazione PL*, Capitolo 2 «Richiami: programmazione lineare e analisi di sensitività»: i modelli lineari, la dualità e il protocollo di sensitività.
- Dal modulo *Ottimizzazione convessa*, Capitolo 6 «Problemi di ottimizzazione convessi e concavi»: insiemi e funzioni convessi, e i problemi di ottimizzazione convessi.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- descrivere il percorso di ogni capitolo: problema gestionale, modello, Python e solver, interpretazione;
- distinguere le classi QP e NLP, e dire quando un modello è convesso;
- orientarsi nella mappa dei modelli del corso.

1.1 Che cosa faremo

Questo corso insegna a usare la programmazione non lineare come strumento di *decisione*, non solo di calcolo. Ogni capitolo parte da un problema gestionale concreto — quanto rischio accettare, quale prezzo fissare, dove localizzare un servizio, quanto regolare tenere un piano — e lo trasforma in un modello di ottimizzazione, che viene implementato in Python, risolto con un solver e, soprattutto, *interrogato*: quanto vale un'unità di risorsa in più? Quanto costa un piano più regolare? La soluzione resiste se i dati cambiano?

> **La domanda giusta**
>
> Alla fine di ogni esercitazione la domanda non è soltanto *"qual è l'ottimo?"*, ma *"quale decisione suggeriamo e quanto è robusta?"*. Il valore di un modello sta nella qualità della raccomandazione che permette di formulare.

1.2 Le classi di modelli e le sigle usate

In tutta la dispensa i modelli appartengono a due classi, indicate con le sigle standard internazionali:

- **QP** (*Quadratic Programming*, programmazione quadratica): obiettivo quadratico, vincoli lineari;
- **NLP** (*Nonlinear Programming*, programmazione non lineare): obiettivo o vincoli non lineari generali.

La sigla **LP** (*Linear Programming*) indica i modelli lineari, che qui compaiono solo come termine di confronto. Un problema è *convesso* quando ogni minimo locale è anche globale (definizione precisa nel capitolo 2): per QP e NLP dipende dalle funzioni, e lo segnaleremo ogni volta.

Notazione, formato dei modelli e scaletta dei capitoli (motivazione, modello, esempio a mano, caso di studio, codice, risultati, sensitività, esercizi) sono gli stessi di Modellazione PL. I numeri di ogni capitolo sono generati dagli script della cartella `python/` (`python3 esegui_tutti.py`).

1.3 Mappa dei modelli

Cap.	Applicazione	Classe	Concetto centrale
4	Produzione con smoothing quadratico	QP conv.	regolarità del piano
5	Supply chain e sostenibilità	NLP conv.	congestione, CO_2
6	Portafoglio di Markowitz	QP conv.	rischio-rendimento
7	Pricing e revenue management	NLP	elasticità e capacità
8	Budget pubblicitario	NLP conv.	rendimenti decrescenti
9	Localizzazione continua	NLP conv.	efficienza-equità
10	Ricarica EV con profilo regolare	QP conv.	nervosismo del prelievo
11	Capacità e tempi di attesa	NLP conv.	capacità-attesa
12	Support Vector Machine	QP conv.	ottimizzazione per il ML
13	Regressione ai minimi quadrati	QP conv.	media, anomali, ridge

I capitoli 4, 10 e 13 sono le varianti quadratiche di tre casi che in Modellazione PL restano lineari: stessi dati, obiettivo diverso.

1.4 Organizzazione del corso

Il percorso essenziale prevede quattro laboratori (capitolo 14 per il dettaglio): (1) Markowitz e frontiera efficiente; (2) produzione con smoothing e moltiplicatori dei QP; (3) modellazione non lineare (pricing o budget); (4) progetto a scelta con presentazione manageriale.

1.5 In sintesi

In sintesi

- Ogni capitolo traduce un problema gestionale in un modello non lineare, lo risolve con Gurobi da Python e lo interroga.
- QP: obiettivo quadratico e vincoli lineari; NLP: obiettivo o vincoli non lineari qualsiasi.
- La convessità decide se un ottimo locale è globale: per QP e NLP dipende dalle funzioni, e lo si dice ogni volta.
- Notazione, formato dei modelli e schema dei capitoli sono quelli di Modellazione PL.

Capitolo 2

Richiami: convessità, programmazione quadratica e condizioni KKT

Prerequisiti

- Dal modulo *Ottimizzazione convessa*, Capitolo 3 «Funzioni convesse e concave»: funzioni convesse e concave e la loro caratterizzazione al secondo ordine.
- Dal modulo *Ottimizzazione convessa*, Capitolo 4 «Funzioni quadratiche»: le funzioni quadratiche e le matrici semidefinite positive.
- Dal modulo *Modellazione PL*, Capitolo 2 «Richiami: programmazione lineare e analisi di sensitività»: la dualità della PL, gli scarti complementari e i prezzi ombra.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- riconoscere un problema convesso e usare il fatto che ogni minimo locale è globale;
- decidere dalla matrice $\boldsymbol{Q}$ se un QP è convesso;
- scrivere le condizioni KKT di un problema e usarle per verificare una soluzione;
- leggere i moltiplicatori come prezzi ombra e stimarli per perturbazione.

Questo capitolo raccoglie gli strumenti teorici usati in tutta la dispensa: convessità e programmazione quadratica e condizioni KKT. Serve anche da riferimento rapido.

Che cosa si dà per noto

La teoria della programmazione lineare (dualità, scarti complementari, prezzi ombra) è richiamata in Modellazione PL, «Teoria: programmazione lineare» e svolta per intero nel corso Programmazione lineare. Il protocollo di analisi di sensitività seguito in ogni capitolo è descritto in Modellazione PL. Insiemi e funzioni convesse sono trattati nel corso Ottimizzazione convessa.

2.1 Convessità e programmazione quadratica

Un problema $\min f(x_1, x_2, \ldots, x_n)$ su un insieme ammissibile X è **convesso** se f è convessa e X è convesso. La proprietà che useremo di continuo:

> **Perché la convessità conta**
>
> In un problema convesso ogni minimo locale è globale: l'ottimalità si può *certificare*. Nei problemi non convessi (ad esempio un problema di pricing con termine bilineare $p \cdot q$) un metodo locale garantisce solo un ottimo locale; i QP non convessi si risolvono comunque all'ottimo globale con tecniche apposite, ma il costo computazionale cresce.

Un **QP** ha obiettivo $\frac{1}{2}\sum_i \sum_j q_{ij}\, x_i x_j + \sum_j c_j x_j$ e vincoli lineari; è convesso se e solo se la matrice $\boldsymbol{Q} = (q_{ij})$ è semidefinita positiva ($\boldsymbol{Q} \succeq 0$). L'esempio che segue ne mostra uno in piccolo; nella dispensa i QP compaiono ogni volta che un costo o un rischio cresce più che linearmente con la decisione.

Un QP 2×2, svolto per intero

Consideriamo un QP di minimo con obiettivo quadratico separabile (la seconda variabile "costa" il doppio della prima, al margine) e un vincolo di soglia sulla somma:

$$\begin{aligned} \min \quad & x_1^2 + 2\,x_2^2 \\ \text{soggetto a} \quad & x_1 + \quad x_2 \quad \geq \quad 6 \\ & x_1, \qquad x_2 \quad \geq \quad 0 \end{aligned}$$

Qui $\boldsymbol{Q} = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix} \succ 0$: il QP è convesso e l'ottimo trovato è globale.

- **Passo 1 — risolvendo si ottiene:**

$$\tilde{x}_1 = 4, \qquad \tilde{x}_2 = 2, \qquad \tilde{f} = 24, \qquad \lambda = 8.$$

- **Passo 2 — verifica del prezzo ombra.** Il vincolo è attivo ($4 + 2 = 6$: l'obiettivo cresce con le variabili, quindi la somma non supera il necessario) e le derivate parziali coincidono:

$$2\tilde{x}_1 = 8 = 4\tilde{x}_2 = \lambda. \quad \checkmark$$

  Se fossero diverse converrebbe spostare quantità dall'una all'altra variabile; la derivata comune è proprio il moltiplicatore.

- **Passo 3 — verifica per perturbazione.** Con termine noto 7 si ottiene $(\tilde{x}_1, \tilde{x}_2) = (14/3,\ 7/3)$ con

$$\tilde{f} = 32{,}67 \approx 24 + 8$$

  (in più c'è il termine di curvatura: qui il prezzo ombra è esatto solo al margine, non per incrementi finiti).

2.2 Le condizioni di ottimalità: le condizioni KKT

Le condizioni di ottimalità dei problemi non lineari vincolati prendono il nome di **condizioni KKT** (da Karush, Kuhn e Tucker, che le formularono). Per il problema $\min f(x_1,\dots,x_n)$ soggetto a $g_i(x_1,\dots,x_n) \le 0$ per ogni $i \in \{1,2,\dots,m\}$ e $h_j(x_1,\dots,x_n) = 0$ per ogni $j \in \{1,2,\dots,q\}$, la Lagrangiana è

$$L(x_1,\dots,x_n) = f(x_1,\dots,x_n) + \sum_{i=1}^{m} \lambda_i\, g_i(x_1,\dots,x_n) + \sum_{j=1}^{q} \nu_j\, h_j(x_1,\dots,x_n), \qquad \lambda_i \ge 0.$$

Condizioni KKT

In un punto di ottimo regolare $(\tilde{x}_1,\dots,\tilde{x}_n)$ esistono moltiplicatori $\lambda_i \ge 0$ e ν_j tali che

$$\frac{\partial L}{\partial x_k}(\tilde{x}_1,\dots,\tilde{x}_n) = 0 \quad \forall k \in \{1,2,\dots,n\} \quad \text{(stazionarietà)},$$

$$\lambda_i\, g_i(\tilde{x}_1,\dots,\tilde{x}_n) = 0 \quad \forall i \in \{1,2,\dots,m\} \quad \text{(complementarietà)}.$$

Se il problema è convesso, le KKT sono anche *sufficienti*: un punto che le soddisfa è un ottimo globale.

Le KKT sono gli scarti complementari, generalizzati

Per un LP le KKT si riducono *esattamente* alle condizioni di ottimalità della programmazione lineare (Modellazione PL, «Teoria: programmazione lineare»): la stazionarietà della Lagrangiana equivale all'ammissibilità del duale (i moltiplicatori λ_i sono le variabili duali π_i, e i moltiplicatori dei vincoli di segno sono i costi ridotti), e la complementarietà $\lambda_i\, g_i(\tilde{x}_1,\dots,\tilde{x}_n) = 0$ è lo scarto complementare $\bar{\pi}_i \cdot \bar{s}_i = 0$. Gli scarti complementari sono le KKT nel caso lineare.

Le KKT sul QP dell'esempio, svolte per intero

Riprendiamo il QP appena risolto: $\min\ x_1^2 + 2x_2^2$ soggetto a $x_1 + x_2 \ge 6$, $x_1, x_2 \ge 0$.

- **Passo 1.** Riscriviamo il vincolo in forma standard: $g(x_1,x_2) = 6 - x_1 - x_2 \le 0$.
- **Passo 2.** Stazionarietà della Lagrangiana $L = x_1^2 + 2x_2^2 + \lambda\,(6 - x_1 - x_2)$:

  $$\frac{\partial L}{\partial x_1} = 2x_1 - \lambda = 0, \qquad \frac{\partial L}{\partial x_2} = 4x_2 - \lambda = 0 \ \Rightarrow\ x_1 = \frac{\lambda}{2},\ x_2 = \frac{\lambda}{4}$$

  (i vincoli $x_1, x_2 \ge 0$ risulteranno non attivi: i loro moltiplicatori sono nulli per complementarietà).
- **Passo 3.** Complementarietà: se il vincolo di soglia fosse inattivo ($\lambda = 0$) avremmo $x_1 = x_2 = 0$, che viola $x_1 + x_2 \ge 6$: impossibile. Quindi il vincolo è attivo: $\lambda/2 + \lambda/4 = 6 \Rightarrow \lambda = 8$, $\tilde{x}_1 = 4$, $\tilde{x}_2 = 2$, $\tilde{f} = 24$ — gli stessi numeri trovati per via elementare nell'esempio del QP, ora ottenuti con la procedura generale.
- **Passo 4.** Lettura del moltiplicatore: con termine noto $6 + \varepsilon$ il valore ottimo cresce di circa $\lambda\varepsilon = 8\varepsilon$. Verifica: con termine noto d la soluzione è $\tilde{x}_1 = 2d/3$, $\tilde{x}_2 = d/3$, quindi

  $$\tilde{f}(d) = \tfrac{2}{3}d^2 \ \Rightarrow\ \tilde{f}(6+\varepsilon) = 24 + 8\varepsilon + \tfrac{2}{3}\varepsilon^2. \quad \checkmark$$

Spiegazione riga per riga

I moltiplicatori KKT generalizzano i prezzi ombra dell'LP: misurano la variazione marginale del valore ottimo rispetto al termine noto del vincolo. Nel laboratorio, quando i duali non sono disponibili (modelli non lineari generali), stimiamo il moltiplicatore *per perturbazione*: si aumenta il termine noto di ε, si riottimizza e si calcola il rapporto incrementale. Questa verifica numerica è essa stessa un esercizio ricorrente.

2.3 In sintesi

In sintesi

- In un problema convesso ogni minimo locale è globale; un QP è convesso se e solo se $\boldsymbol{Q} \succeq 0$.
- Le condizioni KKT sono stazionarietà della lagrangiana, ammissibilità e complementarità; nei problemi convessi sono sufficienti.
- Nel caso lineare le condizioni KKT si riducono agli scarti complementari.
- I moltiplicatori sono prezzi ombra, e si possono stimare anche per perturbazione.

Esercizio 2.1 — KKT

Risolvere con le KKT: $\min (x-3)^2 + (y-1)^2$ soggetto a $x + 2y \le 2$, $x, y \ge 0$. Dire quali vincoli sono attivi all'ottimo e interpretare i moltiplicatori.

Esercizio 2.2 — convessità

Per quali valori del parametro γ la funzione $f(x_1, x_2) = x_1^2 + \gamma x_1 x_2 + x_2^2$ è convessa? (Scrivere Q e imporre $Q \succeq 0$.)

Capitolo 3

Il solver per i modelli non lineari

Prerequisiti

- Dal modulo *Modellazione PL*, Capitolo 3 «Il solver: costruire, risolvere, interpretare»: i comandi generali di `gurobipy`: costruire, risolvere, leggere la soluzione.
- Da questo modulo, Capitolo 2 «Richiami: convessità, programmazione quadratica e condizioni KKT»: i QP convessi e i moltiplicatori KKT.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- scrivere funzioni non lineari con vincoli funzionali $y = f(x)$ e rapporti con vincoli bilineari;
- scegliere tolleranze e riformulazioni che diano un buon scaling;
- leggere i moltiplicatori di un QP convesso con `Pi`;
- stimare un prezzo ombra per perturbazione quando il solver non dà i duali.

Questo capitolo raccoglie i comandi di `gurobipy` propri dei modelli non lineari: vincoli funzionali e bilineari, tolleranze, moltiplicatori dei QP. I comandi generali (installazione e licenza, i cinque passi per costruire un modello, il log, gli attributi della soluzione) sono in Modellazione PL, «Implementazione: modelli lineari» e nella guida al solver di quel corso.

Un solver solo per tutto il corso: Gurobi copre nativamente LP e QP (anche non convessi, con `m.Params.NonConvex = 2`) e, dalla versione 12, risolve *globalmente* anche i modelli con funzioni non lineari generali.

3.1 Vincoli funzionali e termini bilineari

Le funzioni non lineari — logaritmi, esponenziali, potenze — entrano nel modello tramite i *vincoli funzionali*: si introduce una variabile ausiliaria y e si impone $y = f(x)$. I rapporti come $1/(\mu-\lambda)$ si riformulano invece con una variabile ausiliaria e un vincolo bilineare ($w\,(\mu-\lambda) = 1$).

```
m.Params.FuncNonlinear = 1      # funzioni trattate in modo esatto (globale)
z = m.addVar(lb=-GRB.INFINITY)
m.addGenConstrLog(g, z)         # z = log(g);  anche Exp, Pow, Sin, ...
```

```
m.addQConstr(w * v == 1)        # termini bilineari: servono NonConvex = 2
```

3.2 Tolleranze e scalatura

Due accorgimenti pratici, entrambi usati negli script della dispensa:

- **tolleranze**: per analisi marginali accurate (differenze tra due ottimi vicini) stringere `MIPGap`, `FeasibilityTol` e `OptimalityTol` a 10^{-9};
- **scalatura**: riformulare per evitare quantità minuscole — ad esempio $q \le A\,p^{-\varepsilon}$ si scrive $q \cdot r \le A$ con $r = p^{\varepsilon}$, che tiene i numeri in un range sano.

3.3 I moltiplicatori nei QP convessi

Nei QP convessi con vincoli lineari Gurobi restituisce i moltiplicatori come negli LP: l'attributo `Pi` di ogni vincolo è il moltiplicatore KKT del capitolo 2, con la convenzione di segno di Gurobi (variazione del valore ottimo per unità di termine noto). Per i vincoli quadratici serve `m.Params.QCPDual = 1` e l'attributo è `QCPi`. A differenza degli LP, il moltiplicatore è esatto solo al margine: per incrementi finiti il valore ottimo cambia anche per la curvatura, e la verifica per perturbazione si fa con un ε piccolo.

3.4 Quando i duali non ci sono

Per i modelli con vincoli funzionali o bilineari il solver non fornisce i duali: prezzi ombra e moltiplicatori si stimano *per perturbazione* (aumentare il termine noto di ε, riottimizzare, rapporto incrementale), come per i moltiplicatori KKT del capitolo 2.

> **Perché un solver solo**
>
> Stessa sintassi, stessa checklist di interpretazione e — soprattutto — ottimo *globale certificato* anche nei problemi non convessi: con un solver locale ogni risultato andrebbe accompagnato dalla domanda "è solo un ottimo locale?". I problemi convessi con sola struttura quadratica o conica (vincoli come $d_x^2 + d_y^2 \le d^2$) non richiedono nulla di speciale.

3.5 In sintesi

In sintesi

- Le funzioni non lineari entrano con vincoli funzionali $y = f(x)$; i rapporti con vincoli bilineari e `NonConvex = 2`.
- Le analisi marginali vogliono tolleranze strette; una buona riformulazione migliora lo scaling.
- In un QP convesso i moltiplicatori si leggono con `Pi`; senza duali, i prezzi ombra si stimano per perturbazione.
- Un solo solver per tutti i modelli: la stessa sintassi, e un ottimo globale certificato.

Esercizio 3.1

Implementare il QP del capitolo 2, min $x_1^2 + 2x_2^2$ con $x_1 + x_2 \geq 6$, stampare `Pi` e `Slack` del vincolo e verificare il moltiplicatore ri-ottimizzando con termine noto 6,01.

Esercizio 3.2

Risolvere con i vincoli funzionali di Gurobi il problema in una variabile max $5\log(1+x) - x$ con $x \in [0, 10]$ e confrontare con la soluzione analitica $(5/(1+x) - 1 = 0 \Rightarrow x = 4)$.

Parte II

Modelli deterministici non lineari

Capitolo 4

Produzione con smoothing quadratico

Classe: QP convesso **Script:** `python/lab04_produzione_quadratica.py`
Pagina online

Prerequisiti

- Dal modulo *Modellazione PL*, Capitolo 4 «Produzione e scorte multiperiodali»: il modello lineare di produzione e scorte, con i suoi dati.
- Da questo modulo, Capitolo 2 «Richiami: convessità, programmazione quadratica e condizioni KKT»: i programmi quadratici convessi, le condizioni KKT e i moltiplicatori come prezzi ombra.
- Da questo modulo, Capitolo 3 «Il solver per i modelli non lineari»: gli obiettivi quadratici e i moltiplicatori di un QP in `gurobipy`.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- aggiungere un termine quadratico convesso a un modello lineare;
- leggere i moltiplicatori di un QP e verificarli per perturbazione;
- tracciare la frontiera costo-regolarità al variare del peso.

4.1 Che cosa cambia rispetto al modello lineare

Il caso è quello di Modellazione PL, «Produzione e scorte»: tre prodotti, sei mesi, una risorsa condivisa, con gli stessi dati e la stessa descrizione. Il piano a costo minimo di quel modello cambia ritmo bruscamente da un mese all'altro: la produzione totale passa da 219,5 a 389,0 unità, con salti fino a 81,4 unità in un mese. Ogni salto ha un costo che il modello lineare non vede: straordinari, turni da riorganizzare, fornitori da riallineare.

Qui aggiungiamo all'obiettivo una penalità *quadratica* sui salti della produzione totale. Un salto doppio costa quattro volte tanto: il modello preferisce molti piccoli aggiustamenti a pochi grandi cambi di ritmo.

4.2 Il modello

Che cosa cambia. Dati, variabili e vincoli sono quelli del modello lineare: n prodotti indicizzati da i, m mesi indicizzati da t, domanda d_{it}, costi c_{it} e h_{it}, ore unitarie a_i, capacità b_t, scorta iniziale s_{i0}; variabili x_{it} (produzione) e s_{it} (scorta a fine mese). Si aggiunge un solo dato:

Simbolo	Tipo	Significato
γ	$\in \mathbb{Q}_{\geq 0}$	peso dei salti della produzione totale (€ per unità^2)

Usando le stesse variabili, un modello QP per il problema è il seguente:

Modello QP di produzione con smoothing

$$\min \sum_{i=1}^{n}\sum_{t=1}^{m}\left(c_{it}\,x_{it} + h_{it}\,s_{it}\right) + \gamma \sum_{t=2}^{m}\Big(\sum_{i=1}^{n} x_{it} - \sum_{i=1}^{n} x_{i,t-1}\Big)^2 \tag{4.1a}$$

$$\text{soggetto a} \quad s_{i,t-1} + x_{it} = d_{it} + s_{it}, \quad \forall i \in \{1,2,\dots,n\},\ \forall t \in \{1,2,\dots,m\}, \tag{4.1b}$$

$$\sum_{i=1}^{n} a_i\,x_{it} \leq b_t, \qquad \forall t \in \{1,2,\dots,m\}, \tag{4.1c}$$

$$x_{it} \geq 0, \qquad \forall i \in \{1,2,\dots,n\},\ \forall t \in \{1,2,\dots,m\}, \tag{4.1d}$$

$$s_{it} \geq 0, \qquad \forall i \in \{1,2,\dots,n\},\ \forall t \in \{1,2,\dots,m\}. \tag{4.1e}$$

Descrizione della funzione obiettivo e dei vincoli del modello QP:

- la funzione obiettivo quadratica (4.1a) somma al costo di produzione e di giacenza del modello lineare il peso γ per il quadrato di ogni salto della produzione totale fra due mesi consecutivi;
- i vincoli lineari (4.1b) sono i bilanci delle scorte ($n\,m$ vincoli lineari);
- i vincoli lineari (4.1c) limitano le ore macchina di ogni mese (m vincoli lineari);
- i vincoli (4.1d) e (4.1e) definiscono le variabili del modello.

Perché il QP è convesso

Ogni salto $\sum_i x_{it} - \sum_i x_{i,t-1}$ è una funzione lineare delle variabili, e il quadrato di una funzione lineare è convesso. Una somma di funzioni convesse con pesi $\gamma \geq 0$ resta convessa, e i vincoli sono lineari: il QP è convesso, ogni minimo locale è globale e le condizioni KKT del capitolo 2 sono necessarie e sufficienti. Con $\gamma = 0$ si ritrova il modello lineare.

4.3 Esempio svolto a mano

Un prodotto, due mesi

Domanda $d_1 = 60$ e $d_2 = 100$, costo di produzione $c = 10$, giacenza $h = 1$, scorta iniziale nulla, capacità abbondante. Il modello lineare produce su domanda: $x_1 = 60$, $x_2 = 100$, costo 1600, salto 40.

- **Passo 1 — risolvendo si ottiene**, con $\gamma = 0{,}5$:

  $$\tilde{x}_1 = 79{,}75, \quad \tilde{x}_2 = 80{,}25, \quad \tilde{s}_1 = 19{,}75, \quad \tilde{z} = 1619{,}875, \quad \tilde{\pi}_1 = 9{,}5, \quad \tilde{\pi}_2 = 10{,}5,$$

  dove $\tilde{\pi}_1$ e $\tilde{\pi}_2$ sono i moltiplicatori dei due bilanci.

- **Passo 2 — una sola variabile.** Dai bilanci $x_1 = 60 + s_1$ e $x_2 = 100 - s_1$, quindi il salto è $40 - 2s_1$ e l'obiettivo è

  $$f(s_1) = 1600 + s_1 + \gamma\,(40 - 2s_1)^2.$$

- **Passo 3 — stazionarietà.** $f'(s_1) = 1 - 4\gamma\,(40 - 2s_1) = 0$ dà $s_1 = 20 - 1/(8\gamma)$: con $\gamma = 0{,}5$ si trova $s_1 = 19{,}75$, come il solver. Un'unità a scorta costa $h = 1$ e riduce il salto di 2: si accumula finché il risparmio marginale sulla penalità uguaglia la giacenza.

- **Passo 4 — la soglia.** $f'(0) = 1 - 160\gamma$: per $\gamma \le 1/160 = 0{,}00625$ la derivata in $s_1 = 0$ non è negativa e l'ottimo resta senza scorta. Con $\gamma = 0{,}004$ il solver dà infatti $\tilde{s}_1 = 0$, $\tilde{z} = 1606{,}4$, $\tilde{\pi} = (9{,}68;\ 10{,}32)$.

- **Passo 5 — i moltiplicatori.** Con scorta positiva le KKT danno $\tilde{\pi}_2 - \tilde{\pi}_1 = h = 1$: un'unità di domanda in più nel mese 2 costa una giacenza in più di una nel mese 1. Con $\gamma = 0{,}004$ la differenza è $0{,}64 < h$, e infatti tenere scorta non conviene.

4.4 Implementazione

Rispetto allo script lineare cambia solo l'obiettivo; la versione integrale è nello script indicato in testa al capitolo:

```
tot = {t: gp.quicksum(x[i, t] for i in prodotti) for t in mesi}
costo = gp.quicksum(c[i]*x[i, t] + h[i]*s[i, t]
                    for i in prodotti for t in mesi)
salti = gp.quicksum((tot[t] - tot[t-1]) * (tot[t] - tot[t-1])
                    for t in mesi[1:])
m.setObjective(costo + gamma * salti, GRB.MINIMIZE)
m.optimize()
pi = [v_cap[t].Pi for t in mesi]      # moltiplicatori della capacità
```

4.5 Risultati

```
gamma = 0,0: obiettivo 32.889,33  costo lineare 32.889,33  salto max 81,4
   produzione totale:  219,5  301,0  341,4  374,3  389,0  324,8
gamma = 0,5: obiettivo 33.003,10  costo lineare 33.000,96  salto max  1,2
```

```
   produzione totale:  323,3  324,3  325,5  326,1  325,9  324,8
gamma = 2,0: obiettivo 33.004,74  costo lineare 33.004,14  salto max  0,4
```

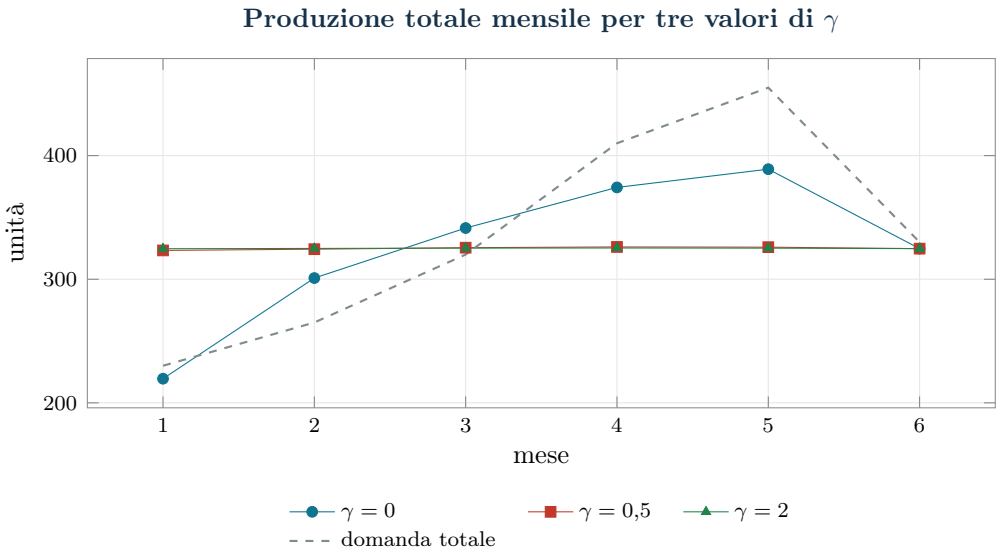


Figura 4.1: Produzione totale mensile: con $\gamma = 0$ (modello lineare) il profilo insegue la domanda; già con $\gamma = 0{,}5$ diventa quasi piatto, intorno a 325 unità al mese.

Il piano regolare costa 111,63 € di produzione e giacenza in più del piano lineare (+0,34%). Con $\gamma = 0{,}5$ il piano per prodotto è:

```
mese              1      2      3      4      5      6
prod. 1     233,3  189,3  132,1  105,7   99,6  160,0
prod. 2      50,0   80,0  110,0  140,0  150,0  100,0
prod. 3      40,0   55,0   83,5   80,4   76,4   64,8
scorte prod. 1: 153,3  212,6  194,7  110,4  0,0  0,0
scorte prod. 3:   0,0    0,0   23,5   23,9  5,2  0,0
```

Insight manageriale

Nel modello lineare la scorta anticipata era quella del prodotto 3, il più costoso in ore. Con lo smoothing a scorta va soprattutto il prodotto 1, fino a 212,6 unità: la penalità conta le *unità*, non le ore, e il prodotto 1 dà più unità per ora macchina ($a_1 = 0{,}9$). Un termine nuovo nell'obiettivo può cambiare la struttura della soluzione, non solo i suoi numeri.

4.6 Analisi di sensitività

Moltiplicatori della capacità

```
mese 1: Pi(gamma=0) =  0,000   Pi(gamma=0,5) =  0,000 EUR/ora
mese 2: Pi(gamma=0) = -0,762   Pi(gamma=0,5) =  0,000 EUR/ora
mese 3: Pi(gamma=0) = -1,524   Pi(gamma=0,5) =  0,000 EUR/ora
mese 4: Pi(gamma=0) = -2,286   Pi(gamma=0,5) = -0,667 EUR/ora
mese 5: Pi(gamma=0) = -3,048   Pi(gamma=0,5) = -1,333 EUR/ora
mese 6: Pi(gamma=0) = -3,810   Pi(gamma=0,5) = -3,134 EUR/ora
Verifica: +1    ora nel mese 6 -> rapporto incrementale -3,060
Verifica: +0,01 ore nel mese 6 -> rapporto incrementale -3,133
```

Il moltiplicatore di un QP è esatto solo al margine

Nel modello lineare un'ora in più nel mese 6 fa scendere il costo esattamente del prezzo ombra, finché si resta nell'intervallo di validità. Nel QP il valore ottimo ha curvatura: con un'ora in più la variazione è −3,060, con 0,01 ore il rapporto incrementale è −3,133, cioè il moltiplicatore −3,134. Con lo smoothing la capacità dei primi tre mesi non è più attiva: il piano livellato non satura le ore all'inizio dell'orizzonte.

La frontiera costo-regolarità

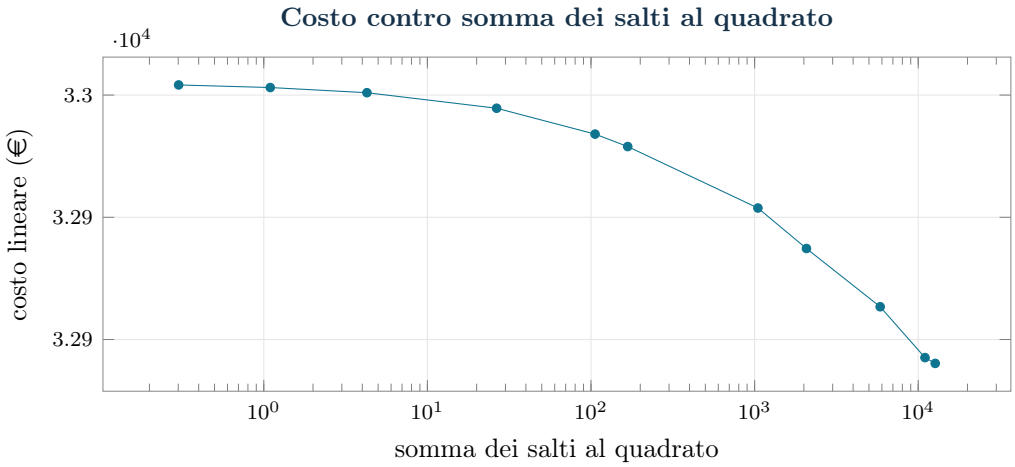


Figura 4.2: Ogni punto è il piano ottimo per un valore di γ fra 0,001 e 2: a sinistra i piani più regolari, in alto i più costosi.

```
gamma = 0,001: costo 32.890,23  somma salti^2 12.703,7  salto max 72,5
gamma = 0,010: costo 32.937,22  somma salti^2  2.074,7  salto max 26,0
gamma = 0,050: costo 32.978,93  somma salti^2    167,7  salto max  7,4
gamma = 0,500: costo 33.000,96  somma salti^2      4,3  salto max  1,2
```

Insight manageriale

I primi passi verso la regolarità costano pochissimo: con $\gamma = 0{,}01$ la somma dei salti al quadrato scende di oltre sei volte (da 13.699 a 2.075) per 48 €. Un piano quasi piatto costa in tutto circa 112 € su un orizzonte di 33.000: se un cambio di ritmo costa all'azienda più di qualche euro, conviene livellare.

4.7 In sintesi

In sintesi

- Un termine quadratico convesso aggiunto a un LP dà un QP convesso.
- Il peso γ traccia la frontiera costo-regolarità: i primi passi costano poco.
- I moltiplicatori del QP si leggono con `Pi`, ma valgono al margine.
- Un termine nuovo può cambiare la struttura della soluzione.

4.8 Esercizi

Esercizio 4.1 — a mano

Nell'esempio svolto a mano porre $\gamma = 0{,}05$: calcolare s_1, x_1, x_2, il valore ottimo e i moltiplicatori dei bilanci, poi controllare con il solver.

Esercizio 4.2 — verifica

Con $\gamma = 0{,}5$ verificare per perturbazione il moltiplicatore della capacità del mese 5, con incrementi di 1 e di 0,01 ore, e confrontare con la verifica del mese 6 nel testo.

Esercizio 4.3 — smoothing per prodotto

Penalizzare i salti di ogni prodotto, $\gamma \sum_{i=1}^{n} \sum_{t=2}^{m} (x_{it} - x_{i,t-1})^2$, con $\gamma = 0{,}5$. Confrontare costo e profili con lo smoothing della produzione totale.

Esercizio 4.4 — smoothing lineare

Sostituire il quadrato con il valore assoluto, $\gamma \sum_{t=2}^{m} |v_t - v_{t-1}|$ con $v_t = \sum_i x_{it}$, scritto come LP con due variabili non negative per salto. Con $\gamma = 5$, che forma ha il profilo? Perché?

Esercizio 4.5 — livello obiettivo

Penalizzare lo scostamento da un livello fisso, $\gamma \sum_{t=1}^{m} (v_t - 325)^2$, con $\gamma = 0{,}5$. Confrontare costo e profilo con lo smoothing dei salti.

Capitolo 5

Supply chain con congestione e sostenibilità

Classe: LP / NLP convesso **Script:** `python/lab05_supplychain.py`

Pagina online

Prerequisiti

- Dal modulo *Modellazione PL*, Capitolo 2 «Richiami: programmazione lineare e analisi di sensitività»: prezzi ombra, costi ridotti e i loro intervalli.
- Da questo modulo, Capitolo 2 «Richiami: convessità, programmazione quadratica e condizioni KKT»: i programmi quadratici convessi, le condizioni KKT e i moltiplicatori come prezzi ombra.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- formulare un problema di flusso a costo minimo con vincoli di conservazione;
- modellare la congestione con termini quadratici convessi;
- tracciare una frontiera costi-emissioni con un prezzo interno della CO_2;
- capire perché le soluzioni degli LP saltano da un vertice all'altro.

5.1 Motivazione gestionale

I prodotti attraversano una rete di stabilimenti, hub e mercati. Ogni tratta ha un costo, una capacità e un'impronta di CO_2; il costo effettivo cresce rapidamente quando una tratta si avvicina alla saturazione. Tre domande manageriali: come instradare i flussi al minimo costo? cosa cambia se penalizziamo la congestione? quanto deve valere la CO_2 perché le rotte "pulite" diventino convenienti?

È il problema quotidiano di chi gestisce distribuzione e logistica: scegliere le rotte, decidere quanto caricare ciascuna tratta, valutare investimenti in capacità e, sempre più spesso, rendere conto dell'impronta carbonica delle spedizioni. La struttura matematica — il flusso su rete — è tra le più riutilizzabili di tutta la ricerca operativa.

5.2 Costruzione guidata del modello

Il problema a parole. *Decidiamo* quante unità far viaggiare su ciascuna tratta della rete. *L'obiettivo* è minimizzare il costo totale di trasporto (nelle varianti: più congestione o CO_2). *I vincoli* impongono che in ogni nodo i flussi tornino — gli stabilimenti non spediscono più di quanto producono, gli hub non trattengono nulla, i mercati ricevono esattamente la loro domanda — e che nessuna tratta superi la propria capacità.

Dati (input del modello)

Simbolo	Tipo	Significato
$G = (N, A)$	grafo diretto	nodi N e archi $A \subseteq N \times N$; scriviamo $\lvert N\rvert$ e $\lvert A\rvert$ per il numero di nodi e di archi
b_i	$\in \mathbb{Q}$	saldo del nodo $i \in N$: offerta se > 0, domanda se < 0, transito se $= 0$
u_{ij}	$\in \mathbb{Q}_{>0}$	capacità dell'arco $(i, j) \in A$ (unità)
c_{ij}	$\in \mathbb{Q}_{\geq 0}$	costo di trasporto di un'unità sull'arco (i, j) (€)
e_{ij}	$\in \mathbb{Q}_{\geq 0}$	emissioni per unità trasportata su (i, j) (kg CO_2)

Variabili decisionali

Introduciamo una variabile non negativa per ciascuno degli $|A|$ archi:

$$x_{ij} = \text{unità di prodotto inviate sull'arco } (i, j), \qquad \forall (i, j) \in A.$$

Usando queste variabili, un modello LP per il problema è il seguente:

Modello LP di flusso a costo minimo

$$\min \sum_{(i,j)\in A} c_{ij}\, x_{ij} \tag{5.1a}$$

$$\text{soggetto a} \sum_{j:\,(i,j)\in A} x_{ij} - \sum_{j:\,(j,i)\in A} x_{ji} = b_i, \qquad \forall i \in N, \tag{5.1b}$$

$$x_{ij} \leq u_{ij}, \qquad \forall (i, j) \in A, \tag{5.1c}$$

$$x_{ij} \geq 0, \qquad \forall (i, j) \in A. \tag{5.1d}$$

Descrizione della funzione obiettivo e dei vincoli del modello LP:

- la funzione obiettivo lineare (5.1a) minimizza il costo totale di trasporto: ogni unità che attraversa l'arco (i, j) paga il costo unitario c_{ij};
- i vincoli lineari (5.1b) sono le conservazioni del flusso: in ogni nodo, ciò che esce meno ciò che entra è pari al saldo b_i — positivo negli stabilimenti, negativo nei mercati, nullo negli hub ($|N|$ vincoli lineari); la somma dei saldi su tutti i nodi deve essere ≥ 0, altrimenti il modello è inammissibile;
- i vincoli lineari (5.1c) assicurano che nessuna tratta trasporti più della propria capacità ($|A|$ vincoli lineari);
- i vincoli (5.1d) definiscono le variabili del modello.

Varianti sull'obiettivo (i vincoli non cambiano):

$$\underbrace{\textstyle\sum_{(i,j)\in A}\left(c_{ij}\,x_{ij}+\alpha\,c_{ij}\,x_{ij}^2/u_{ij}\right)}_{\text{congestione (convessa)}} \qquad \underbrace{\textstyle\sum_{(i,j)\in A}(c_{ij}+\tau\,e_{ij})\,x_{ij}}_{\text{prezzo interno } CO_2}$$

$$\underbrace{\min z \;:\; x_{ij}/u_{ij}\le z,\;\; \forall (i,j)\in A}_{\text{minimax utilizzo}}$$

dove $\alpha \in \mathbb{Q}_{\ge 0}$ pesa la congestione e $\tau \in \mathbb{Q}_{\ge 0}$ (€/kg) è il prezzo interno della CO_2.

Spiegazione riga per riga

Congestione. Il termine $\alpha\,c_{ij}\,x_{ij}^2/u_{ij}$ è convesso e cresce con l'utilizzo dell'arco: rappresenta straordinari, code ai terminal, premi assicurativi. Il suo effetto è *ripartire* i flussi su più rotte invece di saturare la più economica.
Prezzo della CO_2. Non serve un vincolo sulle emissioni: basta aggiungerle al costo con il prezzo interno τ. Variando τ si traccia l'intera frontiera costo-emissioni (approccio "a peso"; l'alternativa "a vincolo" impone $\sum_{(i,j)\in A} e_{ij}\,x_{ij} \le \bar{e}$ e legge il duale).

5.3 Esempio svolto a mano

Due rotte in concorrenza

Un mercato chiede 100 unità servibili da due rotte: rotta 1 con $c_1 = 2$ € e $u_1 = 80$; rotta 2 con $c_2 = 5$ € e capacità abbondante.
LP puro. Si riempie la rotta economica e si trabocca sull'altra: $x_1 = 80$, $x_2 = 20$, costo $2\cdot 80 + 5\cdot 20 = 260$ €. Prezzo ombra della domanda: servire la 101-esima unità costerebbe 5 € (la rotta 1 è satura); prezzo ombra della capacità u_1: $+1$ unità di capacità fa risparmiare $5 - 2 = 3$ €.
Con congestione ($\alpha = 1$): costo rotta $1 = 2x_1 + 2x_1^2/80$. Il costo marginale della rotta 1, $2 + 4x_1/80$, va eguagliato a quello della rotta 2 (altrimenti conviene spostare flusso): $2 + x_1/20 = 5 \Rightarrow x_1 = 60$, $x_2 = 40$. La rotta economica *non viene più saturata*: il modello lascia margine di sicurezza, proprio come farebbe un pianificatore prudente.

5.4 Caso di studio

Rete con 2 stabilimenti (S1, S2), 2 hub (H1, H2) e 4 mercati (M1–M4); 12 archi con capacità, costi ed emissioni *non* proporzionali ai costi: le tratte economiche sono "su strada" (inquinanti), quelle costose "su ferro" (pulite). Dati in `dati/supplychain_archi.csv`; offerta $(260, 240)$, domanda totale 450.

```
m = gp.Model("supply_chain")
x = m.addVars(A, name="x", ub=U)    # ub direttamente sull'arco
m.addConstrs((x.sum(s, "*") <= offerta[s] for s in offerta))
m.addConstrs((x.sum("*", h) == x.sum(h, "*") for h in hub))
v_dom = m.addConstrs((x.sum("*", k) == domanda[k]
                      for k in domanda), name="domanda")
obj = gp.quicksum((c[a] + tau * e[a]) * x[a] for a in A)
if congestione > 0:    # termine quadratico convesso
    obj += gp.quicksum(congestione * c[a] * x[a] * x[a] / U[a]
```

```
                          for a in A)
m.setObjective(obj, GRB.MINIMIZE)
```

5.5 Risultati

```
LP costo minimo : costo 3.385,00 EUR  emissioni 2.730 kg  utilizzo max 100%
  archi saturi: S1->H1, S2->H2, H2->M3
  prezzi ombra della domanda: M1 8,00  M2 9,50  M3 10,00  M4 10,50 EUR/unita'
  costi ridotti archi a zero (soglia = SAObjLow):
    S2->H1 +2,00 (soglia 5,00)   H1->M4 +0,50 (soglia 5,50)
    H2->M1 +4,50 (soglia 1,50)   H2->M2 +1,00 (soglia 3,00)  EUR/unita'
Congestione a=1: costo 3.702,81 EUR  emissioni 2.530 kg  utilizzo max  90%
Minimax        : utilizzo massimo minimo possibile 58,4%
```

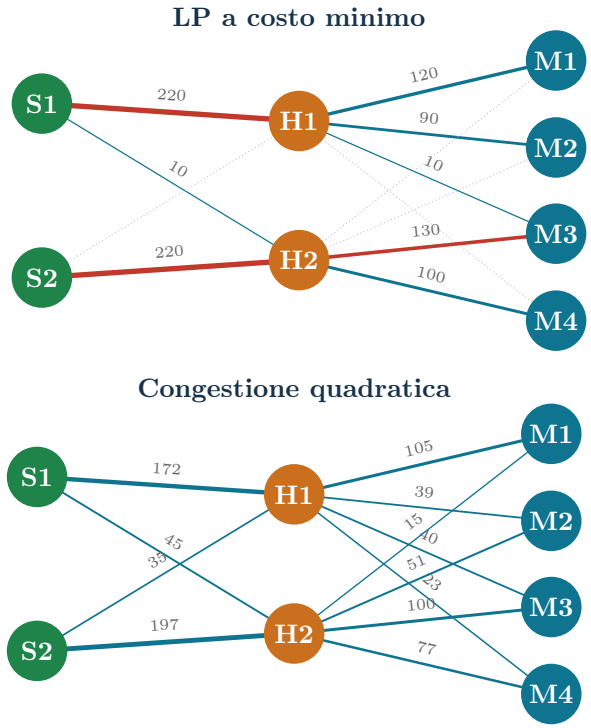


Figura 5.1: I flussi ottimi sulla rete (stabilimenti S verdi, hub H arancioni, mercati M teal; lo spessore degli archi è proporzionale al flusso, il rosso indica un arco saturo, il numero è il flusso in unità). Sopra: LP a costo minimo, che satura tre archi. Sotto: modello con congestione quadratica, che ripartisce i flussi su più rotte e non satura alcun arco.

> **Insight manageriale**
>
> I prezzi ombra della domanda (8–10,50 €/unità) sono i *costi marginali di servizio* dei quattro mercati: è il numero da confrontare con il margine commerciale prima di accettare nuovi ordini, e la base corretta per un prezzo di trasferimento interno. M4 è il mercato più caro da servire: +31% rispetto a M1.

I costi ridotti degli archi non usati, con il loro range

Quattro archi restano a zero e il loro costo ridotto (attributo `x[a].RC`) è la loro *soglia di convenienza*: la rotta S2 → H1 (costo 7 €/unità, `RC` = +2) entrerebbe nella soluzione ottima solo se il suo costo scendesse sotto $7 - 2 = 5$ €/unità — il numero da portare al tavolo in una trattativa con il vettore. Come sempre, accanto al costo ridotto citiamo il suo range di validità `SAObjLow`–`SAObjUp`: per una variabile a zero è $[\text{soglia}, +\infty)$, cioè il costo dell'arco può crescere quanto vuole (resta inutilizzato) e la soglia stessa è `SAObjLow`. La rotta H1 → M4 è a un passo dall'essere usata (`RC` = +0,50): basta uno sconto di 50 centesimi. Le variabili positive hanno tutte costo ridotto nullo.

5.6 Analisi di sensitività: il prezzo della CO_2

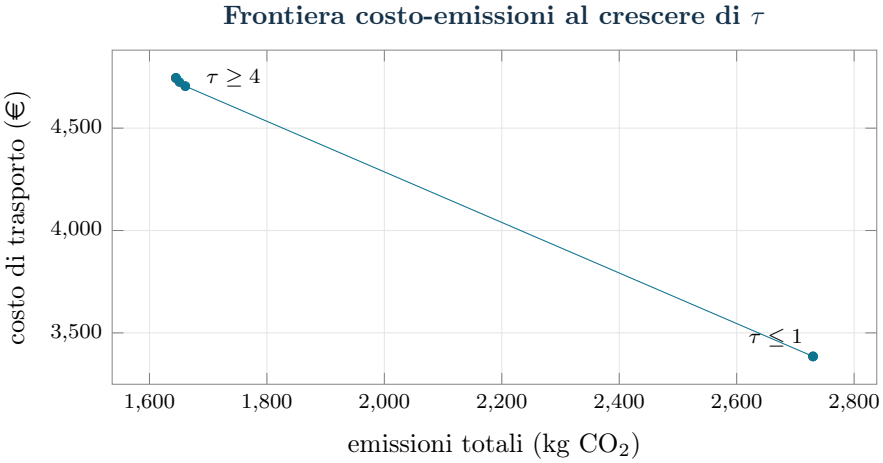


Figura 5.2: Frontiera costo-emissioni: ogni punto è la soluzione ottima per un prezzo interno della CO_2 (τ); in basso a destra l'assetto "su strada" (economico, 2730 kg), in alto a sinistra l'assetto "su ferro" (caro, 1645 kg). La frontiera procede per salti tra vertici dell'LP.

```
tau = 0.0 ... 1.0 : costo 3385  emissioni 2730   (rotte su strada)
tau = 1.24        : costo 4025  emissioni 2173   (vertice intermedio)
tau = 1.26        : costo 4265  emissioni 1981
tau = 1.5         : costo 4705  emissioni 1661   (assetto "ferro")
tau = 4.0 e oltre : costo 4745  emissioni 1645   (stabile)
```

Perché la frontiera procede a salti

Le soluzioni di un LP stanno nei *vertici* del poliedro ammissibile: al crescere di τ l'ottimo resta fermo su un vertice, poi *salta* al successivo. Una bisezione sul modello mostra che fino a $\tau = 1$ non cambia nulla, che tra $\tau = 1{,}0$ e $\tau = 1{,}5$ la rete attraversa vertici intermedi (a $\tau = 1{,}24$: 2173 kg) e che il salto principale avviene esattamente a $\tau = 1{,}25$ €/kg. Managerialmente: un prezzo interno della CO_2 sotto la soglia non cambia le rotte; appena sopra, ristruttura la rete in blocco. Il modello con congestione, strettamente convesso, si sposterebbe invece con continuità.

Il minimax "puro" ($\min z$) è degenere: qualunque instradamento con utilizzo $\leq z^*$ è ottimo, anche costosissimo. In pratica si usa sempre una combinazione (costo $+\rho \cdot z$) o si vincola z e si minimizza il costo.

5.7 In sintesi

In sintesi

- I prezzi ombra della domanda sono i costi marginali di servizio dei mercati; i costi ridotti degli archi non usati sono soglie di convenienza.
- La congestione modellata con termini quadratici abbassa l'utilizzo massimo a un costo maggiore.
- Al variare del prezzo della CO_2 τ si traccia la frontiera costi-emissioni: in un LP l'ottimo salta fra vertici, un modello strettamente convesso si muove con continuità.
- Il minimax puro è degenere: va combinato con il costo.

5.8 Esercizi

Esercizio 5.1 — verifica

Verificare per perturbazione il prezzo ombra della domanda di M4 (portarla a 101) e quello della capacità dell'arco H2→M3.

Esercizio 5.2 — guasto di un hub

Ridurre del 50% le capacità degli archi entranti e uscenti da H2: quanto costa il guasto? quali mercati soffrono di più? (Confrontare i prezzi ombra prima/dopo.)

Esercizio 5.3 — soglia del ferro

Riprodurre con una bisezione su τ la soglia $\tau = 1{,}25$ €/kg trovata nel testo, ed elencare quali archi cambiano flusso attraversandola. Confrontare la soglia con il prezzo ETS corrente della CO_2.

Esercizio 5.4 — vincolo sulle emissioni

Riformulare con vincolo $\sum_{(i,j)\in A} e_{ij}\, x_{ij} \leq \bar{e}$ per $\bar{e} = 2000$ kg e verificare che il duale del vincolo coincide con il τ "a peso" che produce le stesse emissioni.

Esercizio 5.5 — barriera

Sostituire il termine di congestione con la barriera $\alpha\, x_{ij}/(u_{ij} - x_{ij})$ e confrontare: quale formulazione tiene i flussi più lontani dalla saturazione?

Capitolo 6

Portafoglio di Markowitz

Classe: QP convesso **Script:** `python/lab06_markowitz.py`

Pagina online

Prerequisiti

- Da questo modulo, Capitolo 2 «Richiami: convessità, programmazione quadratica e condizioni KKT»: i programmi quadratici convessi, le condizioni KKT e i moltiplicatori come prezzi ombra.
- Dal modulo *Ottimizzazione convessa*, Capitolo 4 «Funzioni quadratiche»: le forme quadratiche e le matrici di covarianza come matrici semidefinite positive.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- formulare e risolvere un QP convesso;
- costruire e leggere la frontiera efficiente;
- capire l'effetto dei vincoli di mandato;
- diffidare delle stime dei rendimenti attesi.

6.1 Motivazione gestionale

Quanto investire in ciascun titolo per bilanciare rendimento atteso e rischio? Il modello di Markowitz è il QP più famoso della storia: minimizza la varianza del portafoglio imponendo un rendimento minimo. Il risultato non è un numero ma una *frontiera efficiente*: il menu completo dei compromessi rischio-rendimento tra cui il decisore sceglie in base alla propria avversione al rischio.

Il problema nasce in finanza (Markowitz, 1952, premio Nobel) ma la sua struttura — allocare un budget tra alternative rischiose e correlate — riappare ovunque: mix di progetti di R&S, portafogli di clienti, perfino l'allocazione di capacità produttiva tra mercati volatili. È anche il ponte naturale tra l'ottimizzazione e la statistica: i dati del modello sono *stime*, con tutte le conseguenze del caso.

6.2 Costruzione guidata del modello

Il problema a parole. *Decidiamo* la quota di capitale da investire in ciascun titolo. *L'obiettivo* è minimizzare il rischio del portafoglio, misurato dalla varianza del suo rendimento. *I vincoli*: investire tutto il capitale (le quote sommano a 1), garantire un rendimento atteso minimo e rispettare le quote minime e massime per titolo.

Dati (input del modello)

Simbolo	Tipo	Significato
n	$\in \mathbb{Z}_{\geq 1}$	numero di titoli; i titoli sono indicizzati da $i \in \{1, 2, \dots, n\}$
μ_i	$\in \mathbb{Q}$	rendimento atteso del titolo i (per anno)
q_{ij}	$\in \mathbb{Q}$	covarianza tra i rendimenti dei titoli i e j; la matrice $\boldsymbol{Q} = (q_{ij})$ è semidefinita positiva ($\boldsymbol{Q} \succeq 0$)
$\bar{r}$	$\in \mathbb{Q}$	rendimento atteso minimo richiesto
ℓ_i, u_i	$\in \mathbb{Q}$, $0 \leq \ell_i \leq u_i \leq 1$	quota minima e massima investibile nel titolo i

Variabili decisionali

Introduciamo le seguenti n variabili non negative:

$$x_i = \text{quota di capitale investita nel titolo } i, \qquad \forall i \in \{1, 2, \dots, n\}.$$

Usando queste variabili, un modello QP per il problema è il seguente:

Modello QP di minima varianza con rendimento minimo

$$\min \quad \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij}\, x_i\, x_j \tag{6.1a}$$

$$\text{soggetto a} \quad \sum_{i=1}^{n} \mu_i\, x_i \geq \bar{r}, \tag{6.1b}$$

$$\sum_{i=1}^{n} x_i = 1, \tag{6.1c}$$

$$x_i \leq u_i, \qquad \forall i \in \{1, 2, \dots, n\}, \tag{6.1d}$$

$$x_i \geq \ell_i, \qquad \forall i \in \{1, 2, \dots, n\}. \tag{6.1e}$$

Descrizione della funzione obiettivo e dei vincoli del modello QP:

- la funzione obiettivo quadratica (6.1a) minimizza la varianza del rendimento del portafoglio; contiene le covarianze q_{ij}, ed è qui che nasce la diversificazione: combinare titoli poco correlati abbassa il rischio complessivo sotto quello dei singoli componenti;
- il vincolo lineare (6.1b) impone che il rendimento atteso del portafoglio raggiunga la soglia $\bar{r}$: è la "manopola" con cui si percorre la frontiera efficiente, e il suo moltiplicatore dice quanta varianza costa ogni punto di rendimento in più (un vincolo lineare);

- il vincolo lineare (6.1c) impone di investire tutto il capitale (un vincolo lineare);
- i vincoli lineari (6.1d) sono i tetti per titolo, tipici vincoli regolamentari o di mandato (n vincoli lineari);
- i vincoli (6.1e) definiscono le variabili del modello (con $\ell_i = 0$: divieto di vendita allo scoperto).

Formulazioni equivalenti. La prima, *media-varianza*, mette il rischio nell'obiettivo con un peso λ:

$$\max \sum_{i=1}^{n} \mu_i x_i - \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j.$$

La seconda, *massimo rendimento a rischio limitato*, lo mette nei vincoli: max $\sum_{i=1}^{n} \mu_i x_i$ con varianza $\leq \bar{\sigma}^2$. Variando $\bar{r}$ (o λ, o $\bar{\sigma}$) si percorre la stessa frontiera.

Spiegazione riga per riga

Convessità. Ogni matrice di covarianza è semidefinita positiva, quindi il QP è convesso e l'ottimo trovato è *globale certificato*.

6.3 Esempio svolto a mano

Due titoli non correlati

Volatilità $\sigma_1 = 20\%$, $\sigma_2 = 30\%$, correlazione nulla. Minimizzare la varianza con $x_1 + x_2 = 1$ (senza vincolo di rendimento).

- **Passo 1.** Sostituendo $x_2 = 1 - x_1$: $f(x_1) = 0{,}04\, x_1^2 + 0{,}09\,(1 - x_1)^2$.
- **Passo 2.** $f'(x_1) = 0{,}08\, x_1 - 0{,}18\,(1 - x_1) = 0 \;\Rightarrow\; x_1 = \dfrac{0{,}09}{0{,}04 + 0{,}09} = \dfrac{9}{13} \approx 69{,}2\%$.
- **Passo 3.** Varianza del portafoglio: $f(9/13) = 0{,}04 \cdot 0{,}479 + 0{,}09 \cdot 0{,}095 = 0{,}0277$, cioè volatilità 16,6%: **meno rischioso di entrambi i titoli**. È l'effetto diversificazione nella sua forma più pura: i pesi ottimi sono inversamente proporzionali alle varianze.

6.4 Caso di studio

Otto ETF settoriali (*Exchange Traded Fund*: fondi quotati che replicano un paniere di titoli di un settore), 60 rendimenti mensili simulati con un modello a un fattore di mercato (file `dati/markowitz_rendimenti.csv`); $\boldsymbol{\mu}$ e $\boldsymbol{Q}$ sono *stimati* da questi dati, come nella pratica. Statistiche annualizzate:

```
ENE: mu = 19,12%  vol = 18,65%      SAN: mu =  2,80%  vol = 11,93%
FIN: mu =  8,27%  vol = 19,59%      CON: mu = 11,25%  vol = 11,67%
TEC: mu =  1,58%  vol = 29,27%      UTL: mu = 11,99%  vol =  7,53%
IND: mu =  9,97%  vol = 16,60%      MAT: mu = 20,93%  vol = 23,93%
```

```
m = gp.Model("markowitz")
x = m.addVars(n, ub=u, name="x")          # quote, 0 <= x_i <= u
m.addConstr(x.sum() == 1, name="budget")
```

```
m.addConstr(gp.quicksum(mu[i]*x[i] for i in range(n)) >= r_min)
m.setObjective(gp.quicksum(Q[i, j]*x[i]*x[j]
                           for i in range(n) for j in range(n)),
               GRB.MINIMIZE)
m.optimize()
```

6.5 Risultati

```
Minima varianza globale : rendimento 11,06%, volatilita'  6,03%
Equipesato (1/n)        : rendimento 10,74%, volatilita' 10,13%
Composizione min varianza: ENE 5,2%  IND 1,9%  SAN 11,5%  CON 22,3%  UTL 58,3%
```

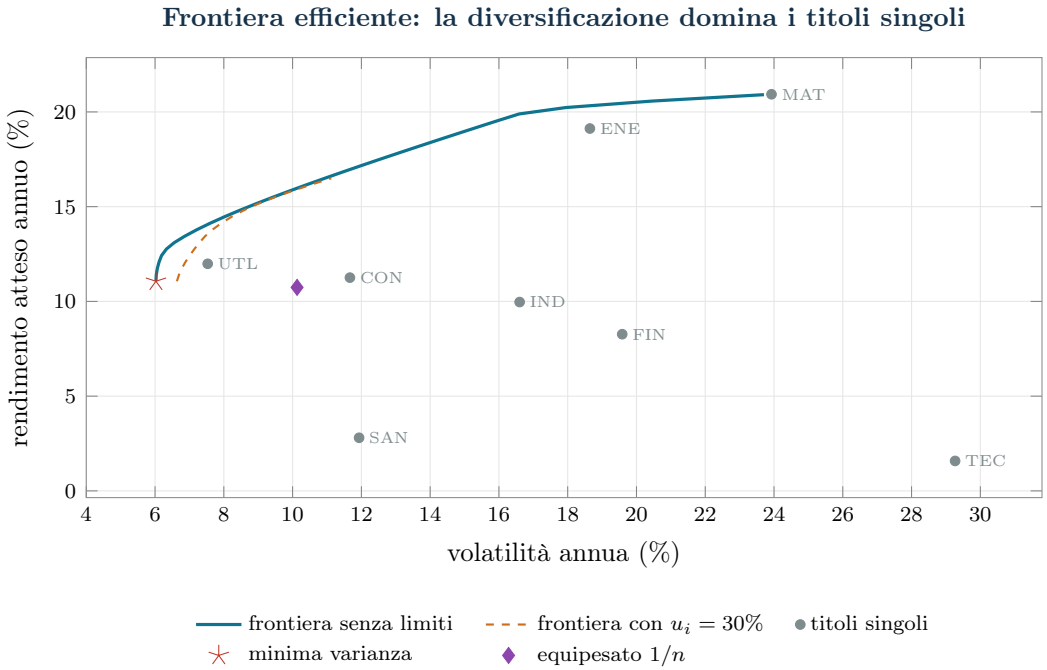


Figura 6.1: Piano rischio-rendimento: gli otto titoli (punti grigi), la frontiera efficiente senza limiti (linea teal), la frontiera con tetto del 30% per titolo (tratteggiata arancione), il portafoglio di minima varianza (stella rossa) e l'equipesato $1/n$ (rombo viola), che risulta dominato.

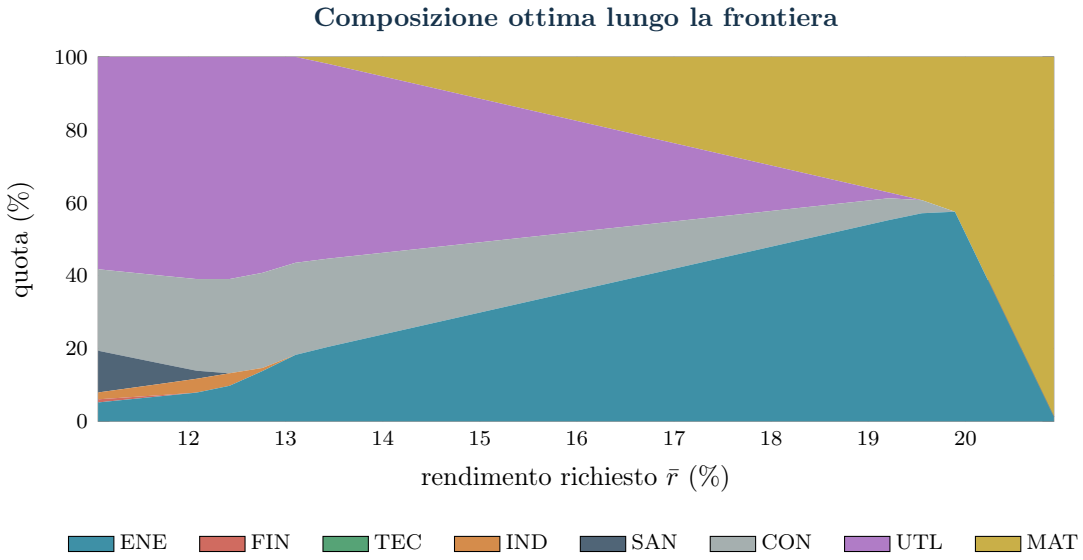


Figura 6.2: Composizione del portafoglio ottimo (aree impilate, in %) al crescere del rendimento richiesto $\bar{r}$: a sinistra domina il mix difensivo (UTL, CON, SAN), verso destra il portafoglio si concentra progressivamente sui titoli ad alto rendimento (MAT, ENE) fino al portafoglio monotitolo all'estremo.

Insight manageriale

Tre messaggi dalla figura. (1) Il portafoglio di minima varianza (volatilità 6%) è molto meno rischioso del miglior titolo singolo (UTL, 7,5%) pur rendendo l'11%. (2) L'equipesato $1/n$ — la strategia "non so nulla" — è nettamente dominato: stessa area di rendimento ma quasi il doppio della volatilità. (3) Il tetto $u_i = 30\%$ taglia la parte alta della frontiera: il costo dei vincoli di mandato si *vede* come distanza tra le due curve.

6.6 Analisi di sensitività

```
r_min =  6%: vol 6,03%   (vincolo NON attivo: coincide col min varianza)
r_min =  8%: vol 6,03%   (idem)
r_min = 10%: vol 6,03%   (idem)
r_min = 12%: vol 6,10%   d(varianza)/d(r_min) ~ 0,0221
```

Un vincolo che non costa nulla

Fino a $\bar{r} = 11\%$ il vincolo di rendimento è *inattivo*: il portafoglio di minima varianza rende già l'11,06%, quindi chiedere "almeno l'8%" non costa nulla e il moltiplicatore è zero. Solo oltre l'11,06% il vincolo morde e ogni punto di rendimento in più si paga in varianza (moltiplicatore $\approx 0{,}022$ a $\bar{r} = 12\%$). Riconoscere i vincoli inattivi evita di "pagare" vincoli gratis.

Nei dati simulati il titolo TEC ha α vero dell'11% annuo, ma su 60 mesi il rendimento *stimato* è 1,6%: il rumore domina. È la lezione pratica più importante di Markowitz: **le stime dei rendimenti attesi sono molto più instabili di quelle delle covarianze**, e i portafogli ottimizzati inseguono gli errori di stima. Rimedi usati in pratica: vincoli u_i,

shrinkage delle stime, oppure ottimizzare solo il rischio (minima varianza).

6.7 In sintesi

In sintesi

- Markowitz minimizza la varianza del portafoglio con un rendimento atteso minimo: un QP convesso.
- Lungo la frontiera efficiente si passa dal mix difensivo ai titoli ad alto rendimento; i vincoli di mandato hanno un costo visibile.
- Sotto il rendimento di minima varianza il vincolo di rendimento non è attivo e il suo moltiplicatore è nullo; oltre, ogni punto di rendimento costa varianza.
- Le stime dei rendimenti attesi sono fragili: tetti, shrinkage e minima varianza sono i rimedi.

6.8 Esercizi

Esercizio 6.1 — verifica

Verificare a mano l'esempio dei due titoli con $\rho = 0{,}5$: la formula diventa $x_1 = (\sigma_2^2 - \rho\sigma_1\sigma_2)/(\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2)$. La diversificazione conviene ancora?

Esercizio 6.2 — frontiera

Ricalcolare la frontiera con $u_i = 20\%$: quanto rendimento massimo si perde? Rappresentare le tre frontiere ($u = 1,\ 0{,}3,\ 0{,}2$) nello stesso grafico.

Esercizio 6.3 — stabilità delle stime

Rieseguire lo script con un seme casuale diverso (`default_rng(1)`) e confrontare le composizioni ottime a parità di $\bar{r}$: quanto cambiano? Quale implicazione operativa ne traete?

Esercizio 6.4 — tracking error

Il benchmark è l'equipesato $x_i^B = 1/8$. Minimizzare il tracking error $\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}(x_i - x_i^B)(x_j - x_j^B)$ con rendimento atteso $\geq 12\%$: quanto ci si deve scostare dal benchmark?

Esercizio 6.5 — costi di transazione

Partendo dal portafoglio equipesato, aggiungere il costo quadratico $\kappa \sum_{i=1}^{n}(x_i - 1/8)^2$ con $\kappa = 0{,}5$: come cambia la composizione rispetto all'ottimo "pulito"?

Capitolo 7

Pricing e revenue management

Classe: NLP concavo / non convesso **Script:** `python/lab07_pricing.py`
Pagina online

Prerequisiti

- Da questo modulo, Capitolo 2 «Richiami: convessità, programmazione quadratica e condizioni KKT»: i programmi quadratici convessi, le condizioni KKT e i moltiplicatori come prezzi ombra.
- Da questo modulo, Capitolo 3 «Il solver per i modelli non lineari»: i vincoli bilineari e `NonConvex = 2`; i prezzi ombra stimati per perturbazione.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- modellare la domanda in funzione del prezzo e capire quando il profitto è concavo;
- gestire un obiettivo bilineare non convesso;
- valutare correttamente la capacità: un posto in più non vale il margine pieno;
- riconoscere gli ottimi d'angolo, dove nessuna derivata si annulla.

7.1 Motivazione gestionale

Quale prezzo massimizza il profitto di un concerto, di un hotel, di un abbonamento, quando la domanda diminuisce al crescere del prezzo? Qui il prezzo non è un dato ma una *variabile*: la domanda diventa endogena e il profitto $p \cdot q$ introduce un termine bilineare. È il capitolo in cui il laboratorio incontra per la prima volta la non convessità — e impara a maneggiarla.

Fissare il prezzo è forse la decisione aziendale a più alto impatto sul margine: un punto percentuale di prezzo vale spesso più di molti punti di volume. Il revenue management — nato nelle compagnie aeree e oggi standard in hotel, eventi, noleggi — aggiunge il vincolo di capacità: i posti invenduti stasera non si recuperano domani.

7.2 Costruzione guidata del modello

Il problema a parole. *Decidiamo* il prezzo di vendita e la quantità da vendere. *L'obiettivo* è massimizzare il profitto, cioè ricavi meno costi variabili. *I vincoli*: non si può vendere più di quanto il mercato chiede a quel prezzo, né più della capacità disponibile.

Dati (input del modello)

Simbolo	Tipo	Significato
$d(\cdot)$	funzione $\mathbb{Q}_{\geq 0} \to \mathbb{Q}_{\geq 0}$	domanda attesa al prezzo p, decrescente in p
c	$\in \mathbb{Q}_{\geq 0}$	costo marginale unitario (€)
k	$\in \mathbb{Q}_{>0}$	capacità disponibile (posti, camere, unità)

Variabili decisionali

Introduciamo le seguenti 2 variabili non negative:

$$\begin{cases} p = \text{prezzo di vendita deciso dall'impresa (€)} \\ q = \text{quantità venduta (unità)} \end{cases}$$

Usando queste variabili, un modello di pricing per il problema è il seguente:

Modello di pricing con vincolo di capacità

$$\max \quad p\,q - c\,q \tag{7.1a}$$

$$\text{soggetto a} \quad q \leq d(p), \tag{7.1b}$$

$$q \leq k, \tag{7.1c}$$

$$p \geq 0, \tag{7.1d}$$

$$q \geq 0. \tag{7.1e}$$

Descrizione della funzione obiettivo e dei vincoli del modello:

- la funzione obiettivo (7.1a) massimizza il profitto $p\,q - c\,q = (p - c)\,q$, ricavo meno costo variabile; contiene il prodotto tra le due variabili p e q (termine *bilineare*, non convesso), ed è qui che nasce tutta la difficoltà del capitolo;
- il vincolo (7.1b) impone di non vendere più di quanto il mercato chiede al prezzo p (un vincolo, lineare se d è lineare);
- il vincolo lineare (7.1c) impone di non vendere più della capacità disponibile; quando è attivo, è lui a determinare il prezzo ottimo (un vincolo lineare);
- i vincoli (7.1d) e (7.1e) definiscono le variabili del modello.

Funzioni di domanda usate nel caso di studio, con i rispettivi dati (tutti razionali positivi): lineare $d(p) = a - b\,p$; a elasticità costante $d(p) = \theta\,p^{-\varepsilon}$ con $\varepsilon > 1$; logistica $d(p) = m/\big(1 + e^{-\alpha+\beta p}\big)$.

Spiegazione riga per riga

Perché $q \le d(p)$ e non $q = d(p)$? All'ottimo la disuguaglianza (7.1b) è attiva da sola (vendere meno del possibile a quel prezzo non conviene mai se $p > c$), ma scriverla come $\le$ mantiene la regione ammissibile più semplice e il modello ammissibile anche quando $d(p) > k$.
Dove sta la non convessità? Nel prodotto $p \cdot q$ dell'obiettivo. Con domanda lineare, sostituendo $q = a - b\,p$, il profitto $(p-c)(a-b\,p)$ è una parabola concava: il problema *ridotto* è facile. Gurobi risolve comunque la versione bilineare all'ottimo globale (`NonConvex=2`); per le domande non lineari generali (elasticità costante, logistica) bastano i vincoli funzionali di Gurobi (`addGenConstrPow`, `addGenConstrExp`): stesso solver, ottimo ancora globale.

7.3 Esempio svolto a mano

Domanda lineare, con e senza capacità

Concerto: $d(p) = 1200 - 5p$ (quindi $a = 1200$, $b = 5$), costo marginale $c = 20$ €, teatro da $k = 400$ posti.

- **Passo 1 — senza capacità.** Profitto $\Pi(p) = (p-20)(1200-5p)$; $\Pi'(p) = 1200 - 10p + 100 = 0$ dà $p^\circ = \dfrac{a/b + c}{2} = \dfrac{240 + 20}{2} = 130$ €, con $q^\circ = 1200 - 650 = 550$ biglietti.
- **Passo 2 — la capacità morde.** $q^\circ = 550 > 400$: il vincolo è attivo, quindi si vende esattamente k e il prezzo che svuota il teatro è $p^* = (a-k)/b = (1200-400)/5 = 160$ €. Profitto: $(160-20)\cdot 400 = 56.000$ €.
- **Passo 3 — quanto vale un posto in più?** Aumentando k, il prezzo ottimo scende di $1/b$: $\dfrac{d\Pi}{dk} = \underbrace{(p^* - c)}_{140} + k\,\underbrace{\dfrac{dp^*}{dk}}_{-1/5} = 140 - 80 = 60$ €. Un posto in più *non* vale il margine pieno (140 €): per riempirlo bisogna abbassare il prezzo a tutti. Errore classico di valutazione degli investimenti di capacità.

7.4 Caso di studio e implementazione

Stessi dati dell'esempio (è il caso raro in cui l'istanza realistica è già piccola), più le varianti a elasticità costante ($\theta = 6\cdot 10^6$, $\varepsilon = 2{,}2$), logistica ($m = 900$, $\alpha = 6$, $\beta = 0{,}045$) e il multiprodotto.

```
m = gp.Model("pricing_lineare")
m.Params.NonConvex = 2              # obiettivo bilineare p*q
p = m.addVar(ub=a/b, name="p")
q = m.addVar(name="q")
m.addConstr(q <= a - b*p, name="domanda")
v_cap = m.addConstr(q <= K, name="capienza")
m.setObjective(p*q - c*q, GRB.MAXIMIZE)
m.optimize()
# niente duali nei QP non convessi: sensitivita' per perturbazione
v_cap.RHS = K + 1; m.optimize()
```

```
Gurobi:  p* = 160,00 EUR, q* = 400, profitto = 56.000,00 EUR
Valore marginale di un posto: 59,80 EUR (teoria: 60,00)
Elasticita' costante: p* = 79,11 EUR  (ottimo nel punto in cui d(p) = k)
Logistica           : p* = 138,29 EUR, profitto = 47.316,83 EUR
```

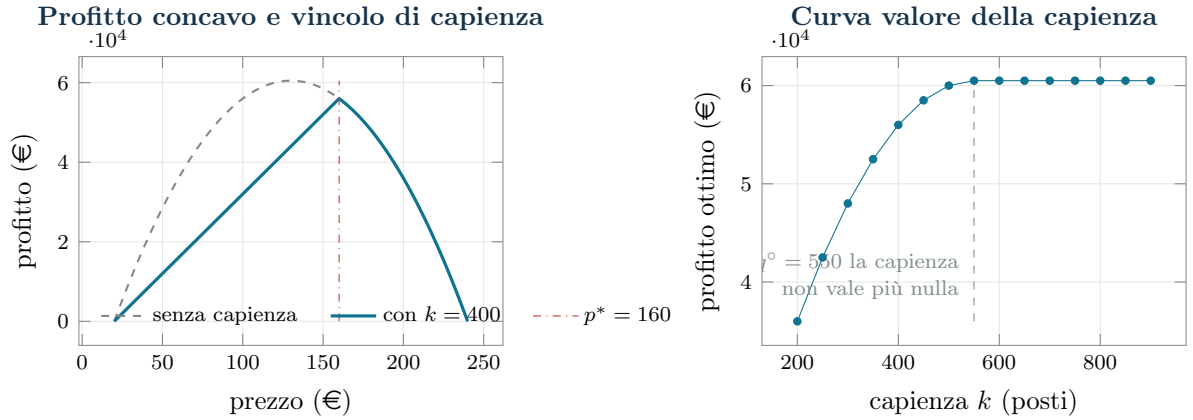


Figura 7.1: A sinistra: profitto in funzione del prezzo per la domanda lineare, senza vincolo (parabola grigia tratteggiata) e con capienza $k = 400$ (linea teal): il vincolo taglia la parabola e sposta l'ottimo a $p^* = 160$ €. A destra: profitto ottimo al crescere della capienza; oltre $q^\circ = 550$ posti la curva diventa piatta e la capacità aggiuntiva non vale più nulla.

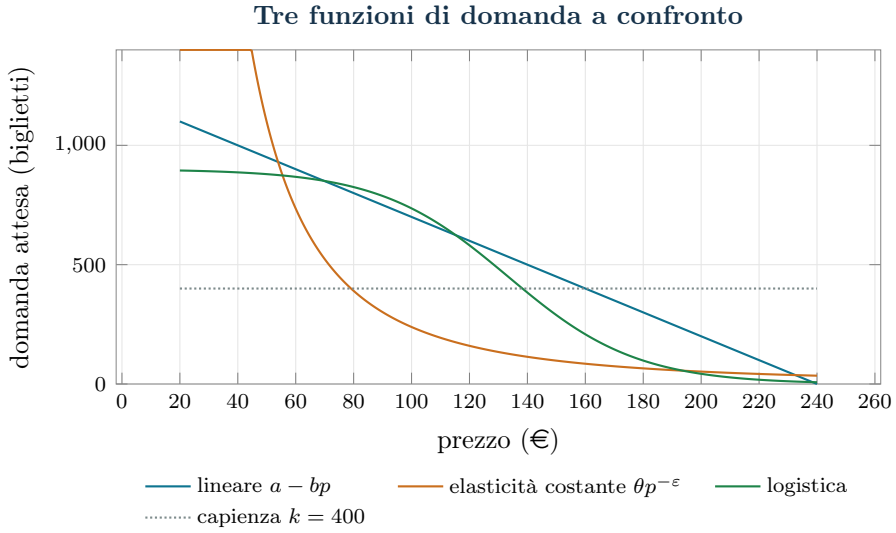


Figura 7.2: Le tre funzioni di domanda del caso di studio a confronto (biglietti attesi in funzione del prezzo) con la capienza $k = 400$ (linea punteggiata): la lineare taglia l'asse, l'elasticità costante ha coda lunga, la logistica descrive la transizione graduale tra domanda alta e bassa.

L'ottimo nel punto di spigolo

Con elasticità costante l'ottimo *non vincolato* sarebbe $p = c\varepsilon/(\varepsilon - 1) = 36{,}67$ €, ma a quel prezzo la domanda supererebbe di gran lunga la capienza. Per p sotto 79,11 € si vende comunque k (conviene quindi alzare il prezzo); sopra, il profitto $(p - c)\,\theta p^{-\varepsilon}$ decresce. L'ottimo $p^* = 79{,}11 = (\theta/k)^{1/\varepsilon}$ sta nel *punto di spigolo* dove $d(p) = k$: non annulla nessuna derivata. Morale: mai cercare l'ottimo solo tra i punti stazionari quando ci sono vincoli.

7.5 Versione multiprodotto

Due categorie di biglietti con sostituzione (alzare il prezzo della platea spinge parte della domanda in galleria): $d_1(p_1, p_2) = 500 - 2p_1 + 0{,}6\,p_2$, $d_2(p_1, p_2) = 900 + 0{,}8\,p_1 - 4p_2$, costi $(30, 15)$ €, capacità $(150, 300)$ posti.

```
platea   : p1* = 234,04 EUR   q1* = 150/150 (piena)
galleria : p2* = 196,81 EUR   q2* = 300/300 (piena)
profitto totale: 85.148,94 EUR
```

Insight manageriale

Entrambe le categorie si riempiono, ma i prezzi non sono "quelli che svuotano" ciascuna sala presa da sola: il modello sfrutta la sostituzione, tenendo la platea cara per spingere domanda verso la galleria. Con prodotti sostituti i prezzi vanno decisi *congiuntamente*: ottimizzarli uno alla volta lascia soldi sul tavolo.

Con funzioni di domanda generali il profitto può non essere concavo: un solver locale può fermarsi su un ottimo locale. Prassi del laboratorio: studiare dominio e derivate della funzione *prima* di fidarsi del solver, provare più punti iniziali, e tracciare sempre il grafico del profitto sull'intervallo rilevante.

7.6 In sintesi

In sintesi

- Con domanda lineare il profitto è concavo; quando la capacità è attiva il prezzo sale fino a riempirla esattamente.
- Un posto in più vale $p^* - c - k/b$, non il margine pieno: la capacità ha valore solo finché è attiva.
- Gurobi risolve il modello bilineare all'ottimo globale con `NonConvex = 2`; senza duali, il valore di un posto si stima per perturbazione.
- Con prodotti sostituti i prezzi vanno decisi insieme; con profitto non concavo: studio della funzione, più punti di partenza, un grafico.

7.7 Esercizi

Esercizio 7.1 — verifica

Verificare con Gurobi il valore marginale del posto per $k = 300$ (dalla tabella: 100 €) e ricavarlo anche con la formula $p^* - c - k/b$.

Esercizio 7.2 — costo marginale

Come cambiano p^*, q^* e il valore del posto se il costo marginale sale a $c = 60$ €? Rifare i tre passi dell'esempio a mano.

Esercizio 7.3 — prezzo di riferimento

Aggiungere la penalità reputazionale $-\gamma\,(p-120)^2$ con $\gamma = 2$: quale prezzo bilancia profitto e reputazione? Il problema resta concavo?

Esercizio 7.4 — elasticità

Per la domanda a elasticità costante, tracciare p^* in funzione di $\varepsilon \in [1{,}3;\ 3]$ con $k = 400$: per quali ε il vincolo di capienza resta attivo?

Esercizio 7.5 — multiprodotto

Nel modello a due categorie, di quanto aumenta il profitto con 50 posti di platea in più? E con 50 di galleria? Quale ampliamento consiglierebbe, a parità di costo?

Capitolo 8

Allocazione del budget pubblicitario

Classe: NLP convesso **Script:** `python/lab08_budget.py`
Pagina online

Prerequisiti

- Dal modulo *Ottimizzazione convessa*, Capitolo 3 «Funzioni convesse e concave»: le funzioni concave e i rendimenti decrescenti.
- Da questo modulo, Capitolo 2 «Richiami: convessità, programmazione quadratica e condizioni KKT»: i programmi quadratici convessi, le condizioni KKT e i moltiplicatori come prezzi ombra.
- Da questo modulo, Capitolo 3 «Il solver per i modelli non lineari»: i vincoli funzionali per le funzioni non lineari in `gurobipy`.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- modellare i rendimenti decrescenti con funzioni concave;
- risolvere NLP convessi con i vincoli funzionali di Gurobi;
- verificare numericamente le condizioni KKT;
- costruire la curva del valore del budget e usare λ per negoziare il budget.

8.1 Motivazione gestionale

Come ripartire una campagna da 100.000 € tra social, search, TV e influencer, quando ogni canale mostra rendimenti marginali decrescenti? I primi euro su un canale rendono molto, i successivi sempre meno: la risposta è una funzione concava della spesa, e l'ottimo segue una regola economica elegante che questo capitolo dimostra e verifica numericamente.

L'allocazione di un budget tra impieghi con rendimenti decrescenti è uno schema universale: marketing su più canali, fondi tra progetti, tempo tra attività, fertilizzante tra campi. La teoria dice qualcosa di forte e verificabile: all'ottimo i rendimenti marginali si eguagliano — ed è esattamente ciò che il solver restituirà, cifra per cifra.

8.2 Costruzione guidata del modello

Il problema a parole. *Decidiamo* quanto spendere in ciascun canale pubblicitario. *L'obiettivo* è massimizzare la risposta totale attesa della campagna. *I vincoli*: la spesa complessiva non può superare il budget, e ciascun canale ha un investimento massimo utile.

Dati (input del modello)

Simbolo	Tipo	Significato
n	$\in \mathbb{Z}_{\geq 1}$	numero di canali pubblicitari; i canali sono indicizzati da $i \in \{1, 2, \ldots, n\}$
b	$\in \mathbb{Q}_{>0}$	budget totale (migliaia di €)
$r_i(\cdot)$	funzione concava crescente	risposta attesa del canale i in funzione della spesa; nel caso di studio $r_i(x) = a_i \log(1+k_i\, x)$ con $a_i, k_i \in \mathbb{Q}_{>0}$ dati
u_i	$\in \mathbb{Q}_{>0}$	massimo investimento utile o consentito nel canale i

Variabili decisionali

Introduciamo le seguenti n variabili non negative:

$$x_i = \text{spesa nel canale } i \text{ (migliaia di €)}, \qquad \forall i \in \{1, 2, \ldots, n\}.$$

Usando queste variabili, un modello NLP convesso per il problema è il seguente:

Modello di allocazione con risposta concava

$$\max \quad \sum_{i=1}^{n} r_i(x_i) \tag{8.1a}$$

$$\text{soggetto a} \quad \sum_{i=1}^{n} x_i \leq b, \tag{8.1b}$$

$$x_i \leq u_i, \qquad \forall i \in \{1, 2, \ldots, n\}, \tag{8.1c}$$

$$x_i \geq 0, \qquad \forall i \in \{1, 2, \ldots, n\}. \tag{8.1d}$$

Descrizione della funzione obiettivo e dei vincoli del modello:

- la funzione obiettivo concava (8.1a) massimizza la risposta totale della campagna; è separabile per canale e somma di funzioni concave, quindi il problema ha ottimo globale certificabile;
- il vincolo lineare (8.1b) impone che la spesa complessiva non superi il budget: è l'unico vincolo che lega i canali tra loro — senza di esso ogni canale andrebbe al proprio tetto (un vincolo lineare);
- i vincoli lineari (8.1c) sono i tetti per canale: il limite oltre cui il canale non assorbe altra spesa utile (n vincoli lineari);
- i vincoli (8.1d) definiscono le variabili del modello.

La condizione economica all'ottimo

Per le KKT, all'ottimo esiste $\lambda \geq 0$ (prezzo ombra del budget) tale che, per ogni canale non bloccato ai limiti 0 o u_i:

$$r_i'(x_i^*) = \lambda.$$

In parole: **l'ultimo euro investito rende lo stesso in tutti i canali attivi**. Se così non fosse, spostare un euro dal canale con marginale basso a quello con marginale alto migliorerebbe la risposta: non saremmo all'ottimo. I canali al tetto u_i possono avere marginale $> \lambda$ (vorremmo investirci di più ma non si può); quelli a zero hanno marginale iniziale $< \lambda$ (non valgono nemmeno il primo euro).

8.3 Esempio svolto a mano

Due canali logaritmici

$r_1(x) = 100\log(1+0{,}2x)$, $r_2(x) = 60\log(1+0{,}1x)$, budget $b = 50$ (migliaia di €), senza tetti.

- **Passo 1 — condizione di uguaglianza dei marginali.**

$$\frac{100 \cdot 0{,}2}{1+0{,}2x_1} = \frac{60 \cdot 0{,}1}{1+0{,}1x_2} \quad\Longrightarrow\quad \frac{20}{1+0{,}2x_1} = \frac{6}{1+0{,}1x_2}.$$

- **Passo 2 — budget.** $x_2 = 50 - x_1$. Sostituendo: $20\,(1+0{,}1(50-x_1)) = 6\,(1+0{,}2x_1)$, cioè $20\,(6-0{,}1x_1) = 6+1{,}2x_1 \Rightarrow 120-2x_1 = 6+1{,}2x_1 \Rightarrow x_1 = \dfrac{114}{3{,}2} = 35{,}6$, $x_2 = 14{,}4$.

- **Passo 3 — prezzo ombra.** $\lambda = 20/(1+0{,}2\cdot 35{,}6) = 2{,}46$: un euro di budget in più renderebbe $\approx 2{,}46$ unità di risposta — lo stesso numero da qualunque canale lo si faccia entrare.

8.4 Caso di studio e implementazione

Quattro canali, $r_i(x) = a_i\log(1+k_ix)$, budget $b = 100$ (migliaia di €), dati in `dati/budget_canali.csv`. Il problema è concavo e lo risolviamo con i vincoli funzionali di Gurobi: una variabile ausiliaria $z_i = \log(g_i)$ con $g_i = 1 + k_ix_i$, e l'obiettivo diventa lineare nelle z_i (ottimo globale certificato):

```
m.Params.FuncNonlinear = 1          # log trattato in modo esatto
x = m.addVars(4, ub=u)
g = m.addVars(4, lb=1.0)            # g_i = 1 + k_i x_i
z = m.addVars(4, lb=-GRB.INFINITY)
for i in range(4):
    m.addConstr(g[i] == 1 + k[i] * x[i])
    m.addGenConstrLog(g[i], z[i])
m.addConstr(x.sum() <= B)
m.setObjective(gp.quicksum(a[i] * z[i] for i in range(4)),
               GRB.MAXIMIZE)
```

8.5 Risultati e verifica delle KKT

```
Risposta totale: 1.449,8 (migliaia di contatti utili)
     canale |  spesa | tetto | risposta | marginale
     social |   24,3 |    60 |    385,3 |   8,2693
     search |   34,8 |    80 |    539,5 |   8,2693
         TV |   22,9 |   120 |    235,2 |   8,2693
 influencer |   18,0 |    35 |    289,8 |   8,2693

Verifica: +1000 EUR di budget -> risposta +8,244 ~ lambda
```

Insight manageriale

La condizione teorica si vede nei numeri: i quattro marginali coincidono alla quarta cifra ($\lambda = 8{,}2693$), e l'aumento reale di risposta con mille euro in più (8,244) conferma la lettura del moltiplicatore. Nota manageriale: la TV riceve *meno* dei social nonostante il tetto più alto — non conta la dimensione del canale ma la velocità con cui satura (k_i).

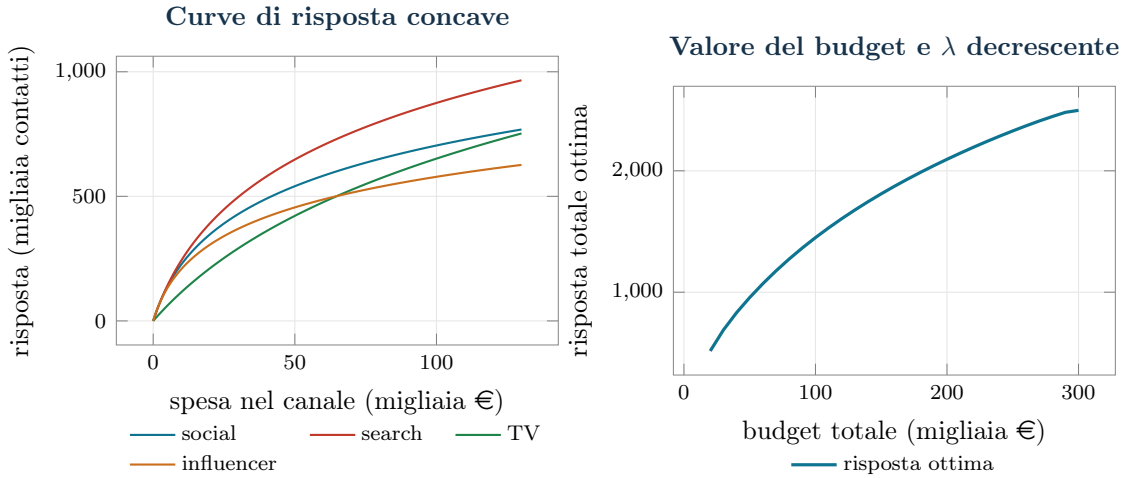


Figura 8.1: A sinistra: le curve di risposta $r_i(x) = a_i \log(1 + k_i x)$ dei quattro canali — concave, con velocità di saturazione diverse. A destra: la risposta totale ottima al crescere del budget (curva valore), concava: ogni tranche di budget rende meno della precedente.

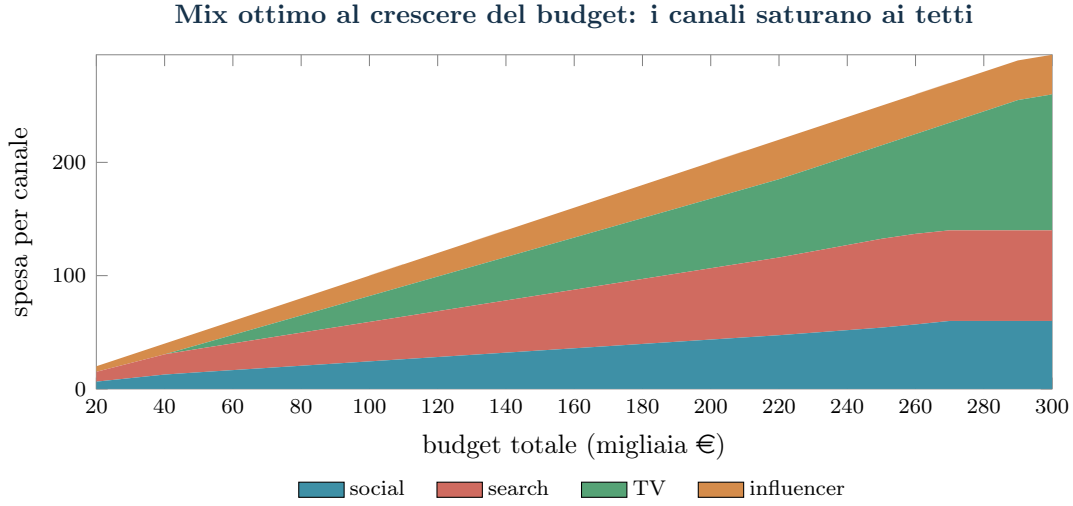


Figura 8.2: Mix ottimo di spesa (aree impilate, migliaia di €) al crescere del budget totale: i canali entrano e crescono insieme mantenendo uguali i ritorni marginali; l'influencer satura per primo al suo tetto (35), seguito dagli altri.

8.6 Analisi di sensitività

```
b =  20: risposta   515,9   lambda = 18,971
b =  60: risposta 1.070,4   lambda = 10,909
b = 100: risposta 1.449,8   lambda =  8,244
b = 180: risposta 1.988,2   lambda =  5,538
b = 260: risposta 2.370,2   lambda =  4,049
b = 300: risposta 2.498,1   lambda =  0,000  (tutti i canali ai tetti)
```

Spiegazione riga per riga

La curva valore-budget è concava: λ scende da 19 a 4 man mano che il budget cresce. A $b = 300$ tutti i canali sono ai tetti ($60 + 80 + 120 + 35 = 295$): il budget smette di essere la risorsa scarsa e λ crolla a zero. Per un CFO: la *curva* di λ è l'argomento quantitativo per decidere quanto budget concedere al marketing — si finanzia finché λ supera il valore di un euro investito altrove.

8.7 In sintesi

In sintesi

- Le risposte logaritmiche sono concave: massimizzarne la somma con un budget è un problema convesso con ottimo globale certificato.
- Gurobi tratta il logaritmo con un vincolo funzionale dopo la riformulazione $z_i = \log(1 + k_i x_i)$.
- All'ottimo i rendimenti marginali dei canali sotto il tetto coincidono con λ, come prevedono le condizioni KKT.
- La risposta cresce con il budget in modo concavo e λ cala: la sua curva è lo strumento per negoziare il budget.

8.8 Esercizi

Esercizio 8.1 — verifica

Risolvere l'esempio a mano con $b = 80$: dove finisce il budget aggiuntivo? Come cambia λ?

Esercizio 8.2 — risposta esponenziale

Sostituire la risposta della TV con $r(x) = 520\,(1 - e^{-0{,}025x})$ e ripetere l'ottimizzazione: cambia il mix? La funzione ha un tetto naturale: il bound u_i serve ancora?

Esercizio 8.3 — canale escluso

Aggiungere un quinto canale con marginale iniziale $a_5 k_5 = 5 < \lambda$: verificare che l'ottimo lo lascia a zero e interpretare il suo "costo ridotto" $a_5 k_5 - \lambda$.

Esercizio 8.4 — copertura equa

Il budget serve tre segmenti di clientela con coperture diverse per canale. Massimizzare la copertura del segmento peggiore ($\max z$ con $z \le \text{copertura}_g$): quanto costa l'equità in risposta totale?

Capitolo 9

Localizzazione continua di un servizio

Classe: NLP convesso **Script:** `python/lab09_localizzazione.py`

Pagina online

Prerequisiti

- Dal modulo *Ottimizzazione convessa*, Capitolo 1 «Insiemi convessi»: gli insiemi convessi, con cui si costruiscono i vincoli geografici.
- Da questo modulo, Capitolo 2 «Richiami: convessità, programmazione quadratica e condizioni KKT»: i programmi quadratici convessi, le condizioni KKT e i moltiplicatori come prezzi ombra.
- Da questo modulo, Capitolo 3 «Il solver per i modelli non lineari»: i vincoli funzionali e le tolleranze in `gurobipy`.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- formulare tre obiettivi di localizzazione e capire quando coincidono;
- scrivere le distanze euclidee con vincoli conici e risolverle con Gurobi da buoni punti di partenza;
- gestire vincoli geografici convessi;
- costruire una frontiera efficienza-equità con il metodo ε-vincolo.

9.1 Motivazione gestionale

Dove collocare una stazione di ricarica rapida, un micro-hub logistico, un presidio sanitario, per essere "vicini" alla domanda? La risposta dipende da cosa significa vicini: distanza *media* (efficienza), distanza *massima* (equità verso l'utente più sfortunato) o distanza *quadratica* (che porta al baricentro). Tre obiettivi, tre punti diversi sulla mappa: la geometria rende il trade-off visibile come in nessun altro modello.

Le decisioni di localizzazione sono tra le più costose da correggere: un hub nel posto sbagliato resta nel posto sbagliato per anni. La versione continua del problema (coordinate libere sul piano) è il laboratorio ideale per un tema che riappare in tutte le decisioni pubbliche e private: il conflitto tra *efficienza* (servire bene in media) ed *equità* (non abbandonare nessuno).

9.2 Costruzione guidata del modello

Il problema a parole. *Decidiamo* le coordinate in cui collocare la nuova struttura. *L'obiettivo* — ed è la vera scelta manageriale — può essere minimizzare la distanza media pesata dagli utenti, oppure la distanza dell'utente più lontano, oppure la distanza quadratica. *I vincoli*, quando presenti, delimitano l'area ammissibile (una zona rettangolare, un raggio massimo da un'infrastruttura).

Dati (input del modello)

Simbolo	Tipo	Significato
n	$\in \mathbb{Z}_{\geq 1}$	numero di punti di domanda (quartieri), indicizzati da $i \in \{1, 2, \ldots, n\}$
(a_i, b_i)	$\in \mathbb{Q}^2$	coordinate del quartiere i (km)
w_i	$\in \mathbb{Q}_{\geq 0}$	peso del quartiere i: popolazione, domanda o priorità
(a_0, b_0), r	$\in \mathbb{Q}^2$, $\in \mathbb{Q}_{>0}$	centro e raggio dell'eventuale vincolo geografico (distanza massima da un'infrastruttura)

Variabili decisionali

Introduciamo le seguenti 2 variabili libere (possono assumere qualunque valore reale) e, per la versione minimax, una variabile ausiliaria non negativa:

$$\begin{cases} x = \text{coordinata est della nuova struttura (km)} \\ y = \text{coordinata nord della nuova struttura (km)} \\ z = \text{distanza massima dal quartiere più lontano (solo minimax)} \end{cases}$$

Usando queste variabili, i modelli per i tre obiettivi classici sono i seguenti:

Modello di Weber (distanza media pesata)

$$\min \quad \sum_{i=1}^{n} w_i \sqrt{(x - a_i)^2 + (y - b_i)^2} \tag{9.1a}$$

$$\text{soggetto a} \quad x \gtreqless 0, \tag{9.1b}$$

$$y \gtreqless 0. \tag{9.1c}$$

Modello minimax (distanza massima)

$$\min \quad z \tag{9.2a}$$

$$\text{soggetto a} \quad \sqrt{(x - a_i)^2 + (y - b_i)^2} \leq z, \qquad \forall i \in \{1, 2, \ldots, n\}, \tag{9.2b}$$

$$x \gtreqless 0, \tag{9.2c}$$

$$y \gtreqless 0, \tag{9.2d}$$

$$z \geq 0. \tag{9.2e}$$

Descrizione delle funzioni obiettivo e dei vincoli dei due modelli:

- la funzione obiettivo convessa (9.1a) minimizza la distanza totale pesata dai quartieri (efficienza); i vincoli (9.1b) e (9.1c) definiscono le variabili, libere;
- la funzione obiettivo (9.2a) minimizza la distanza del quartiere più lontano (equità); i vincoli convessi (9.2b) impongono che ogni quartiere disti al più z dalla struttura (n vincoli); i vincoli (9.2c)–(9.2e) definiscono le variabili;
- terza variante, **quadratica**: minimizzare $\sum_{i=1}^{n} w_i\big[(x-a_i)^2+(y-b_i)^2\big]$, che ha soluzione in forma chiusa (il baricentro pesato).

Vincoli geografici convessi tipici, da aggiungere a qualunque variante: zona rettangolare $x^L \le x \le x^U$, $y^L \le y \le y^U$; distanza massima da un'infrastruttura $(x-a_0)^2+(y-b_0)^2 \le r^2$.

Spiegazione riga per riga

Quadratico ⇒ baricentro. Annullando il gradiente: $x^* = \sum_{i=1}^{n} w_i\, a_i / \sum_{i=1}^{n} w_i$ (e analogo per y): l'unico caso con formula chiusa, utile come punto di partenza per gli altri.
Weber. Non ha forma chiusa; la funzione non è differenziabile nei punti (a_i, b_i). All'ottimo i "tiri" unitari pesati dei quartieri si annullano: $\sum_{i=1}^{n} w_i\, \boldsymbol{u}_i = \mathbf{0}$, dove $\boldsymbol{u}_i$ è il versore dalla struttura verso il quartiere i.
Minimax. L'ottimo è il centro del *cerchio minimo* che racchiude tutti i punti: dipende solo dai quartieri estremi e ignora i pesi — per questo protegge le periferie.

9.3 Esempio svolto a mano

Tre clienti su una strada

Clienti nei km 0, 4 e 10 con pesi 1, 1 e 3.
Weber 1D = mediana pesata. Il costo è $f(x) = |x| + |x-4| + 3|x-10|$. Spostandosi di δ a sinistra da $x = 10$, il costo dei primi due clienti scende di 2δ ma quello del terzo sale di 3δ: peggiora. Da destra, simmetricamente, peggiora ancora. Ottimo: $x^* = 10$, il punto che lascia *almeno metà del peso totale* ($5/2 = 2{,}5$) da entrambi i lati — la mediana pesata.
Baricentro. $x = (1\cdot 0 + 1\cdot 4 + 3\cdot 10)/5 = 6{,}8$: sensibile a *quanto* sono lontani i punti, non solo da che parte stanno.
Minimax. Punto medio degli estremi: $(0+10)/2 = 5$, pesi ignorati.
Tre obiettivi ragionevoli, tre risposte diverse: 10, 6,8, 5. La scelta dell'obiettivo È la decisione manageriale.

9.4 Caso di studio

Dodici quartieri con pesi tra 5 e 25 mila abitanti (`dati/localizzazione_quartieri.csv`); vincolo tecnico: la stazione deve stare entro $r = 2$ km dalla cabina elettrica in $(7, 6)$.

Il trucco di modellazione: una variabile $d_k \ge$ distanza euclidea dal quartiere k, imposta con il vincolo conico (convesso) $d_x^2 + d_y^2 \le d_k^2$; Weber minimizza $\sum_k w_k d_k$, il minimax una z con $d_k \le z$.

```
d = m.addVars(n)                    # d_k >= distanza dal quartiere k
```

```
for k in range(n):
    m.addConstr(dx[k] == px - coord[k, 0])
    m.addConstr(dy[k] == py - coord[k, 1])
    m.addQConstr(dx[k]**2 + dy[k]**2 <= d[k]*d[k])    # cono
# Weber:    min sum w_k d_k
# minimax: min z  con  d_k <= z
# vincolo geografico: (px-7)^2 + (py-6)^2 <= R^2
```

9.5 Risultati

```
Baricentro pesato : (4,897, 5,397)
Weber             : (4,886, 5,310)  dist. media 3,344 km  max 6,314 km
Minimax           : (5,496, 5,029)  dist. media 3,391 km  max 5,680 km
Weber vincolato   : (5,083, 5,429)  costo +0,2% (vincolo attivo, ottimo sul
    bordo)
```

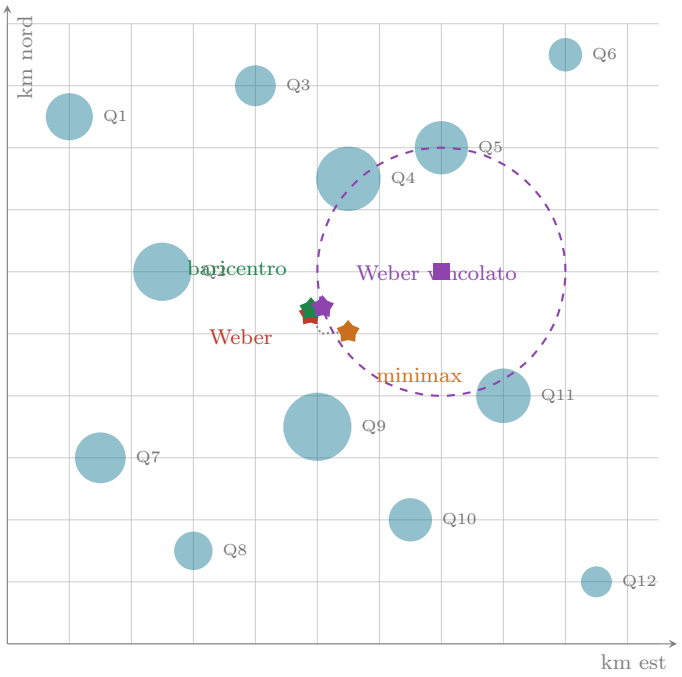


Figura 9.1: Mappa della città: i cerchi teal sono i 12 quartieri (area proporzionale alla popolazione); le stelle indicano le quattro localizzazioni ottime (Weber in rosso, minimax in arancione, baricentro in verde, Weber vincolato in viola); il cerchio viola tratteggiato è il vincolo dei 2 km dalla cabina elettrica; la linea punteggiata è la traiettoria delle soluzioni di compromesso tra minimax e Weber.

> **Insight manageriale**
>
> Weber e baricentro quasi coincidono qui (i pesi sono distribuiti in modo bilanciato), ma il minimax si sposta di oltre mezzo km verso sud-est per proteggere i quartieri periferici. Il vincolo della cabina costa pochissimo (+0,2% di distanza totale): scoprire che un vincolo temuto è quasi gratis è un risultato manageriale importante quanto l'ottimo stesso.

9.6 Analisi di sensitività: la frontiera efficienza-equità

Minimizziamo la distanza media imponendo un tetto D alla distanza massima (metodo ε-constraint), e variamo D tra i due estremi:

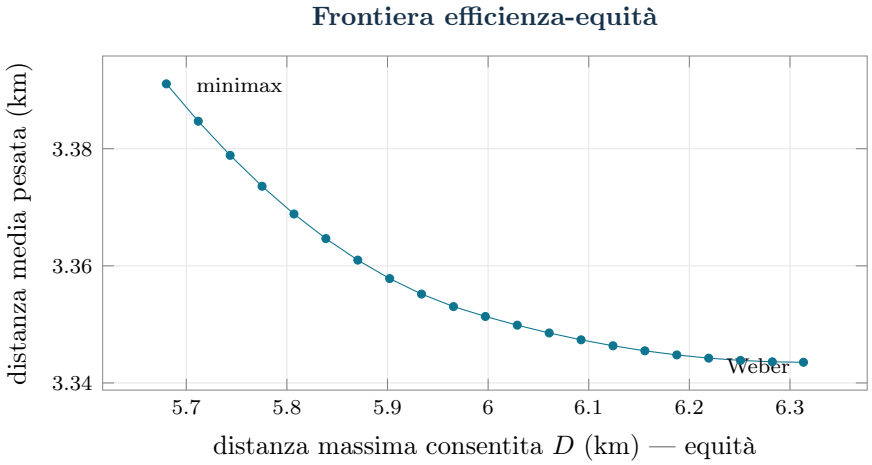


Figura 9.2: Frontiera efficienza-equità (ε-constraint): distanza media pesata minima ottenibile per ogni tetto D imposto alla distanza massima. Gli estremi sono la soluzione minimax (a sinistra) e il punto di Weber (a destra); la curva quasi piatta indica che l'equità costa pochissimo in efficienza.

```
tetto D = 5,680 km: media 3,391   (soluzione minimax)
tetto D = 5,997 km: media 3,351
tetto D = 6,314 km: media 3,344   (soluzione Weber)
```

Spiegazione riga per riga

La frontiera è quasi piatta: garantire al quartiere più lontano 0,63 km in meno costa solo 47 metri di distanza media. Quando la frontiera è così piatta, la soluzione equa è "quasi gratis" — un argomento fortissimo in una discussione pubblica. Il caso opposto (frontiera ripida) segnalerebbe un vero conflitto efficienza-equità.

9.7 In sintesi

In sintesi

- Weber minimizza la distanza media pesata, il minimax la distanza massima, l'obiettivo quadratico dà il baricentro.
- Le distanze euclidee si scrivono con vincoli conici convessi.
- La scelta dell'obiettivo è la decisione gestionale; un vincolo temuto può costare quasi nulla.
- La frontiera efficienza-equità misura quanto costa l'equità.

9.8 Esercizi

Esercizio 9.1 — verifica

Calcolare a mano il baricentro pesato dei 12 quartieri dai dati CSV e confrontarlo con l'output dello script.

Esercizio 9.2 — peso che cambia tutto

Portare il peso di Q12 (il quartiere in basso a destra) da 5 a 40: di quanto si spostano Weber e minimax? Quale dei due è più sensibile ai pesi, e perché?

Esercizio 9.3 — due strutture

Con *due* stazioni e l'assegnazione di ogni quartiere alla più vicina, il problema resta convesso? Provare un approccio alternato (assegna–ottimizza) e discutere perché trova solo ottimi locali.

Esercizio 9.4 — distanza Manhattan

Sostituire la distanza euclidea con $|x - a_i| + |y - b_i|$ e mostrare che il problema di Weber diventa un LP (introdurre variabili ausiliarie per i valori assoluti). Risolverlo con Gurobi e confrontare l'ottimo.

Esercizio 9.5 — moltiplicatore del raggio

Stimare per perturbazione il moltiplicatore del vincolo $r = 2$ km (risolvere con $r = 2{,}1$): quanto vale, in km di distanza totale, ogni 100 m di raggio in più?

Capitolo 10

Ricarica di veicoli elettrici con profilo regolare

Classe: QP convesso **Script:** `python/lab10_ricarica_ev_quadratica.py`

Pagina online

Prerequisiti

- Dal modulo *Modellazione PL*, Capitolo 5 «Ricarica intelligente di veicoli elettrici»: il modello lineare di ricarica, con i suoi dati.
- Da questo modulo, Capitolo 2 «Richiami: convessità, programmazione quadratica e condizioni KKT»: i programmi quadratici convessi, le condizioni KKT e i moltiplicatori come prezzi ombra.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- penalizzare le variazioni di una grandezza aggregata con un termine quadratico;
- leggere il moltiplicatore del fabbisogno di energia in un QP;
- tracciare la frontiera fra spesa e irregolarità del prelievo.

10.1 Che cosa cambia rispetto al modello lineare

Il caso è quello di Modellazione PL, «Ricarica intelligente di veicoli elettrici»: sei furgoni da ricaricare di notte, prezzi orari, carico di base dell'edificio e contatore da 120 kW, con gli stessi dati. La ricarica a costo minimo insegue le ore economiche e crea un prelievo "nervoso": il prelievo totale sale fino a 103,4 kW e cambia fino a 54,9 kW da un'ora all'altra. Rampe così ripide affaticano trasformatore e contatore e, in molte tariffe, si pagano.

Qui aggiungiamo all'obiettivo una penalità *quadratica* sui salti orari del prelievo totale. Il modello lineare limitava solo il picco; il termine quadratico addolcisce l'intero profilo.

10.2 Il modello

Che cosa cambia. Dati, variabili e vincoli sono quelli del modello lineare: n veicoli indicizzati da v, m ore indicizzate da t (con $\Delta t = 1$ h), prezzi π_t, disponibilità a_{vt}, energia e_v, potenze $\bar{p}_v$, carico di base b_t, contatore k, rendimento η; variabili x_{vt} (potenza al veicolo v nell'ora t). Si aggiunge un solo dato:

Simbolo	Tipo	Significato
γ	$\in \mathbb{Q}_{\geq 0}$	peso dei salti del prelievo totale (€ per kW^2)

Indichiamo con $z_t = b_t + \sum_{v=1}^{n} x_{vt}$ il prelievo totale nell'ora t: è un'espressione lineare delle variabili, non una variabile nuova. Usando le stesse variabili, un modello QP per il problema è il seguente:

Modello QP di ricarica con profilo regolare

$$\min \quad \sum_{v=1}^{n}\sum_{t=1}^{m} \pi_t \,\Delta t\, x_{vt} + \gamma \sum_{t=2}^{m} \big(z_t - z_{t-1}\big)^2 \tag{10.1a}$$

$$\text{soggetto a} \quad \eta \sum_{t=1}^{m} \Delta t\, x_{vt} \geq e_v, \qquad \forall v \in \{1,2,\dots,n\}, \tag{10.1b}$$

$$\sum_{v=1}^{n} x_{vt} + b_t \leq k, \qquad \forall t \in \{1,2,\dots,m\}, \tag{10.1c}$$

$$x_{vt} \leq a_{vt}\,\bar{p}_v, \quad \forall v \in \{1,2,\dots,n\},\ \forall t \in \{1,2,\dots,m\}, \tag{10.1d}$$

$$x_{vt} \geq 0, \qquad \forall v \in \{1,2,\dots,n\},\ \forall t \in \{1,2,\dots,m\}. \tag{10.1e}$$

Descrizione della funzione obiettivo e dei vincoli del modello QP:

- la funzione obiettivo quadratica (10.1a) somma alla spesa energetica il peso γ per il quadrato di ogni salto orario del prelievo totale;
- i vincoli lineari (10.1b) garantiscono l'energia richiesta da ogni veicolo (n vincoli lineari);
- i vincoli lineari (10.1c) limitano il prelievo al contatore (m vincoli lineari);
- i vincoli lineari (10.1d) limitano la potenza di ogni veicolo e la azzerano quando è assente ($n\,m$ vincoli lineari);
- i vincoli (10.1e) definiscono le variabili del modello.

Il carico di base entra nella penalità

I salti sono quelli del prelievo *totale*, carico di base compreso: anche con ricarica nulla il profilo dell'edificio ha le sue rampe, e il modello può usare la ricarica per addolcirle. Il QP è convesso per la stessa ragione del capitolo 4: somma di quadrati di funzioni lineari, vincoli lineari.

10.3 Esempio svolto a mano

Un veicolo, due ore

Servono $e = 10$ kWh ($\eta = 1$), prezzi $\pi_1 = 0{,}10$ e $\pi_2 = 0{,}20$ €/kWh, potenza massima $\bar{p} = 12$ kW, carico di base nullo. Il modello lineare carica tutto nell'ora economica: $x_1 = 10$, spesa 1,00 €.

- **Passo 1 — risolvendo si ottiene**, con $\gamma = 0{,}01$:

$$\tilde{x}_1 = 6{,}25, \quad \tilde{x}_2 = 3{,}75, \quad \tilde{z} = 1{,}4375, \quad \tilde{\lambda} = 0{,}15,$$

  dove $\tilde{\lambda}$ è il moltiplicatore del fabbisogno; la spesa è 1,375 €.

- **Passo 2 — una sola variabile.** Il vincolo di energia è attivo: con $x_1 = 5 + d$ e $x_2 = 5 - d$ l'obiettivo è

$$f(d) = 1{,}5 - 0{,}1\,d + 4\gamma\,d^2,$$

  minimo in $d = 1/(80\gamma)$: con $\gamma = 0{,}01$ si trova $d = 1{,}25$, come il solver. Vale finché $x_2 \geq 0$, cioè per $\gamma \geq 1/400$; sotto quella soglia si carica tutto nell'ora 1.

- **Passo 3 — il moltiplicatore.** Stazionarietà rispetto a x_1 e x_2:

$$0{,}10 - 2\gamma\,(x_2 - x_1) = \lambda = 0{,}20 + 2\gamma\,(x_2 - x_1) \;\Rightarrow\; \lambda = \tfrac{0{,}10+0{,}20}{2} = 0{,}15.$$

  Un kWh in più costa la *media* dei due prezzi, per ogni $\gamma \geq 1/400$: il solver dà $\tilde{\lambda} = 0{,}15$ anche con $\gamma = 0{,}1$.

Nel modello lineare il kWh marginale costava il prezzo dell'ora marginale (0,10); con la penalità si divide fra le due ore per tenere piccolo il salto.

10.4 Implementazione

Rispetto allo script lineare cambia solo l'obiettivo; la versione integrale è nello script indicato in testa al capitolo:

```
z = {t: gp.quicksum(x[v, t] for v in veicoli) + base[t] for t in ore}
spesa = gp.quicksum(prezzo[t] * x[v, t] for v in veicoli for t in ore)
salti = gp.quicksum((z[t] - z[t-1]) * (z[t] - z[t-1]) for t in ore[1:])
m.setObjective(spesa + gamma * salti, GRB.MINIMIZE)
```

10.5 Risultati

```
gamma = 0,000: spesa 20,36  picco 103,4 kW  salto max 54,9  somma salti^2
    7268,4
gamma = 0,001: spesa 21,76  picco  77,3 kW  salto max 15,4  somma salti^2
    1188,4
gamma = 0,010: spesa 24,67  picco  68,0 kW  salto max 10,0  somma salti^2
    298,2
```

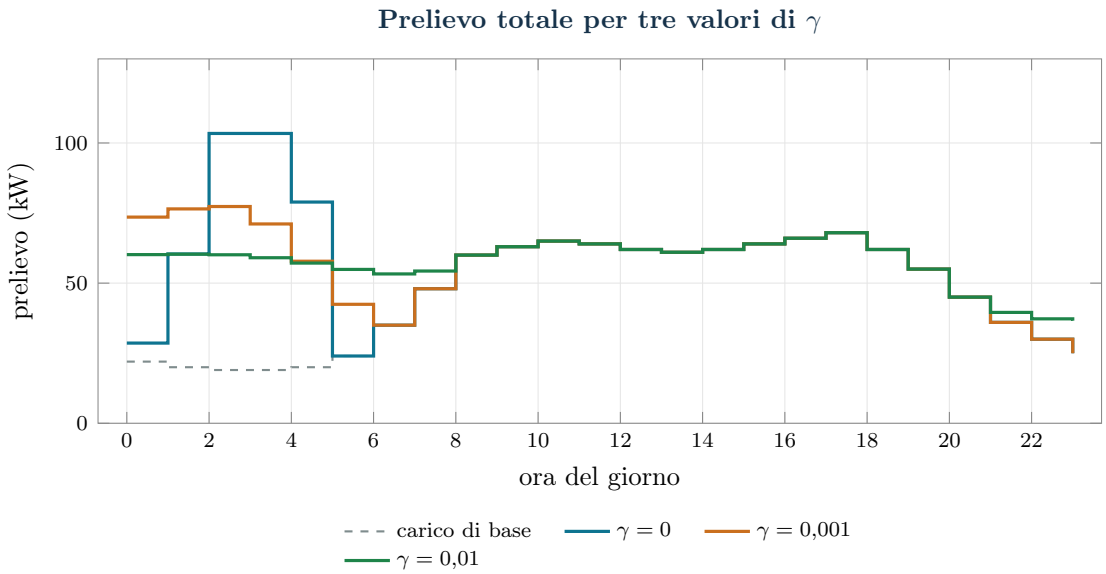


Figura 10.1: Prelievo totale ora per ora: il modello lineare ($\gamma = 0$) concentra la ricarica nelle ore più economiche; al crescere di γ la ricarica si distribuisce sulla notte e le rampe del mattino si addolciscono.

Insight manageriale

Con $\gamma = 0{,}01$ il profilo regolare costa 4,31 € a notte in più e porta il picco a 68 kW, il minimo possibile. Il compromesso lineare costo $+\,\rho$ picco di Modellazione PL arrivava allo stesso picco spendendo meno, perché guarda solo il massimo; il QP paga anche per addolcire ogni rampa. Scegliere fra i due modelli significa decidere che cosa costa davvero all'azienda: il picco o le variazioni.

10.6 Analisi di sensitività

Moltiplicatori del fabbisogno

```
            V1       V2       V3       V4       V5       V6    (EUR/kWh)
g=0       0,0947   0,0842   0,0842   0,0947   0,0947   0,0842
g=0,001   0,0886   0,0886   0,0886   0,0886   0,0886   0,0886
g=0,01    0,0916   0,0916   0,0916   0,0916   0,0916   0,0916
Verifica (g = 0,01): e_1 + 1 kWh -> 0,0917;   e_1 + 0,01 kWh -> 0,0916
```

Un solo prezzo per tutti i veicoli

Nel modello lineare ogni veicolo ha il suo kWh marginale, il prezzo della sua ora marginale diviso per η. Nel QP tutti i moltiplicatori coincidono: la penalità lega i veicoli attraverso il prelievo totale, e un kWh in più di qualunque veicolo si ottiene ridistribuendo la ricarica di tutti. La verifica per perturbazione con 0,01 kWh conferma il valore 0,0916.

La frontiera spesa-nervosismo

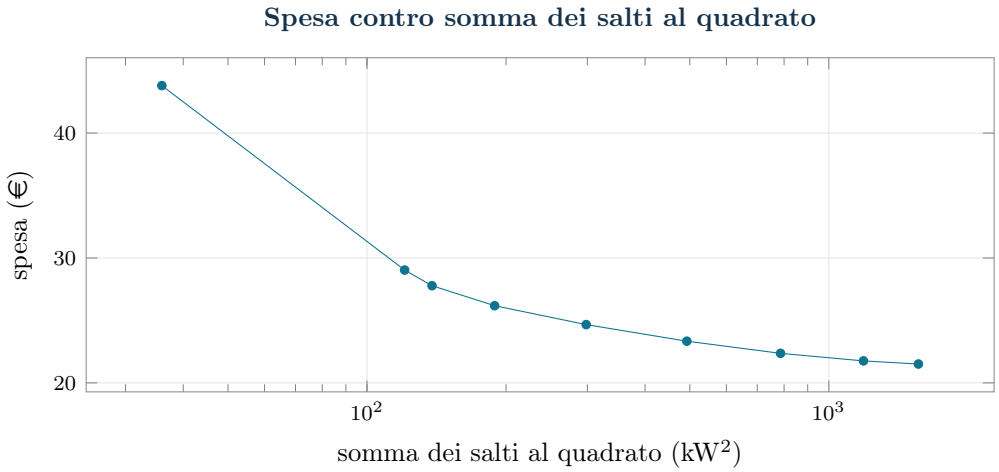


Figura 10.2: Ogni punto è la ricarica ottima per un valore di γ fra 0,0005 e 0,5.

```
gamma = 0,0005: spesa 21,51  somma salti^2 1563,4  picco 82,3 kW
gamma = 0,0050: spesa 23,34  somma salti^2  492,6  picco 68,0 kW
gamma = 0,1000: spesa 29,03  somma salti^2  120,7  picco 68,0 kW
gamma = 0,5000: spesa 43,80  somma salti^2   36,0  picco 68,0 kW
```

Insight manageriale

La frontiera è ripida all'inizio: con 1,15 € in più a notte ($\gamma = 0{,}0005$) la somma dei salti al quadrato scende da 7.268 a 1.563. Oltre $\gamma = 0{,}1$ ogni ulteriore addolcimento costa molto: il carico di base ha rampe proprie, e per seguirle la ricarica deve spostarsi in ore care.

10.7 In sintesi

In sintesi

- Il termine quadratico penalizza ogni variazione del prelievo, non solo il picco.
- Il kWh marginale si divide fra le ore: nell'esempio costa il prezzo medio.
- Nel QP i moltiplicatori dei veicoli coincidono.
- La frontiera è ripida all'inizio: i primi passi di regolarizzazione costano poco.

10.8 Esercizi

Esercizio 10.1 — a mano

Nell'esempio svolto a mano porre $\gamma = 0{,}05$: calcolare x_1, x_2, la spesa, il valore ottimo e il moltiplicatore del fabbisogno, poi controllare con il solver.

Esercizio 10.2 — verifica

Con $\gamma = 0{,}001$ verificare per perturbazione il moltiplicatore del fabbisogno di V3, con incrementi di 1 e di 0,01 kWh.

Esercizio 10.3 — solo la ricarica

Penalizzare i salti della sola ricarica, $\gamma \sum_{t=2}^{m} \big(\sum_v x_{vt} - \sum_v x_{v,t-1}\big)^2$, con $\gamma = 0{,}01$. Come cambiano spesa, picco e salto massimo del prelievo totale?

Esercizio 10.4 — penalità sulla potenza

Sostituire la penalità dei salti con $\gamma \sum_{t=1}^{m} z_t^2$ (perdite che crescono col quadrato della potenza), con $\gamma = 0{,}001$. Che forma prende il prelievo totale nelle ore notturne?

Esercizio 10.5 — contatore

Con $\gamma = 0{,}001$ ridurre il contatore a $k = 70$ kW. Quanto aumenta l'obiettivo? Quale moltiplicatore della rete diventa non nullo?

Capitolo 11

Capacità di servizio e tempi di attesa

Classe: NLP convesso (coda M/M/1) **Script:** `python/lab11_code.py`

Pagina online

Prerequisiti

- Dal modulo *Ottimizzazione convessa*, Capitolo 3 «Funzioni convesse e concave»: le funzioni convesse di una variabile e la loro minimizzazione.
- Da questo modulo, Capitolo 2 «Richiami: convessità, programmazione quadratica e condizioni KKT»: i programmi quadratici convessi, le condizioni KKT e i moltiplicatori come prezzi ombra.
- Da questo modulo, Capitolo 3 «Il solver per i modelli non lineari»: i vincoli funzionali e la stima dei prezzi ombra per perturbazione.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- usare le formule della coda M/M/1 dentro un problema di ottimizzazione;
- ricavare l'ottimo analiticamente e confermarlo numericamente;
- dare un prezzo a una promessa di servizio;
- proteggere il dimensionamento dall'incertezza sulla domanda.

11.1 Motivazione gestionale

Quanta capacità assegnare a un servizio clienti, a uno sportello, a un servizio cloud? Più capacità costa; poca capacità fa esplodere le attese. La teoria delle code fornisce la relazione quantitativa tra capacità e attesa, e l'ottimizzazione la trasforma in una decisione. Il messaggio centrale è contro-intuitivo per chi ragiona "a efficienza": **l'utilizzazione ottima non è il 100%**.

Ogni servizio con arrivi aleatori — call center, pronto soccorso, help desk, API di un servizio cloud — vive lo stesso dilemma: la capacità costa, l'attesa pure. La teoria delle code dà la relazione esatta tra le due; l'ottimizzazione sceglie il punto giusto. Il capitolo è breve ma il suo messaggio (l'utilizzazione ottima non è il 100%) vale da solo una discussione in consiglio di amministrazione.

11.2 Costruzione guidata del modello

Il problema a parole. *Decidiamo* quanta capacità di servizio acquistare. *L'obiettivo* è minimizzare il costo totale: capacità comprata più il valore del tempo che i clienti passano nel sistema. *Il vincolo* strutturale è uno solo, la stabilità della coda ($\mu > \lambda$); nelle varianti si aggiunge la promessa di servizio $w \le w_{\max}$.

Usiamo il modello di coda più semplice, indicato con la notazione standard **M/M/1**: arrivi *markoviani* (processo di Poisson), tempi di servizio *markoviani* (esponenziali) e *un* servente. Per questo sistema valgono le formule chiuse riportate in tabella; ciò che qui è nuovo è trattare la capacità μ come una variabile di ottimizzazione.

Dati (input del modello)

Simbolo	Tipo	Significato
λ	$\in \mathbb{Q}_{>0}$	tasso medio di arrivo dei clienti (richieste/ora)
c	$\in \mathbb{Q}_{>0}$	costo di un'unità di capacità (€ per unità di μ all'ora)
h	$\in \mathbb{Q}_{>0}$	valore di un'ora di permanenza di un cliente nel sistema (€)

Variabile decisionale e grandezze derivate

Introduciamo una variabile continua:

$$\mu = \text{capacità del servizio (tasso di servizio, richieste/ora)},$$

con il requisito di stabilità $\mu > \lambda$. Dalla teoria delle code M/M/1 derivano il tempo medio nel sistema $w(\mu) = 1/(\mu - \lambda)$ (in letteratura: W) e il numero medio di clienti nel sistema $l(\mu) = \lambda/(\mu - \lambda)$ (legge di Little; in letteratura: L).

Usando questa variabile, il modello per il problema è il seguente:

Modello di costo totale capacità + attesa

$$\min \quad c\,\mu + h\,\frac{\lambda}{\mu - \lambda} \tag{11.1a}$$

$$\text{soggetto a} \quad \mu > \lambda. \tag{11.1b}$$

Descrizione della funzione obiettivo e del vincolo del modello:

- la funzione obiettivo (11.1a) è la somma di due termini: il primo, $c\,\mu$, compra la capacità; il secondo, $h \cdot l(\mu)$, monetizza il tempo complessivo che i clienti passano nel sistema; è convessa su $(\lambda, +\infty)$;
- il vincolo (11.1b) definisce la variabile e garantisce la stabilità della coda: con $\mu \le \lambda$ la fila cresce senza limite.

Annullando la derivata dell'obiettivo si ottiene la soluzione in forma chiusa:

$$C'(\mu) = c - \frac{h\,\lambda}{(\mu - \lambda)^2} = 0 \qquad \Longrightarrow \qquad \boxed{\mu^* = \lambda + \sqrt{h\,\lambda / c}}$$

Spiegazione riga per riga

La formula ha la struttura classica "capacità = domanda + scorta di sicurezza": si copre la domanda media λ e si aggiunge un margine $\sqrt{h\lambda/c}$ che cresce con il valore del tempo dei clienti e decresce con il costo della capacità. Il margine cresce come $\sqrt{\lambda}$: i sistemi grandi godono di economie di scala nel "cuscinetto" (pooling).

11.3 Esempio svolto a mano

Sportello con 8 clienti l'ora

$\lambda = 8$/h, $c = 2$ € per unità di capacità, $h = 4$ €/h.

- **Passo 1.** $\mu^* = 8 + \sqrt{4 \cdot 8/2} = 8 + 4 = 12$ clienti/ora.
- **Passo 2.** Utilizzazione $\rho = 8/12 = 66{,}7\%$; tempo medio $w = 1/(12-8) = 15$ minuti; costo $C = 2 \cdot 12 + 4 \cdot 8/4 = 32$ €/h.
- **Passo 3 — perché non tagliare la capacità?** Con $\mu = 9$ ($\rho = 89\%$, "più efficiente"): $C = 18 + 32/1 = 50$ €/h, molto peggio. L'apparente spreco (33% di capacità inutilizzata) è ciò che tiene corte le code.

11.4 Caso di studio e risultati

Servizio clienti: $\lambda = 42$ richieste/ora, $c = 3$ €, $h = 1{,}5$ €.

```
Analitico : mu* = 42 + sqrt(1,5*42/3) = 46,583   costo 153,495 EUR/h
Gurobi    : mu* = 46,582 (bilineare w(mu-lam) = 1, NonConvex=2: coincide)
All'ottimo: rho = 90,2%   w = 13,1 minuti
```

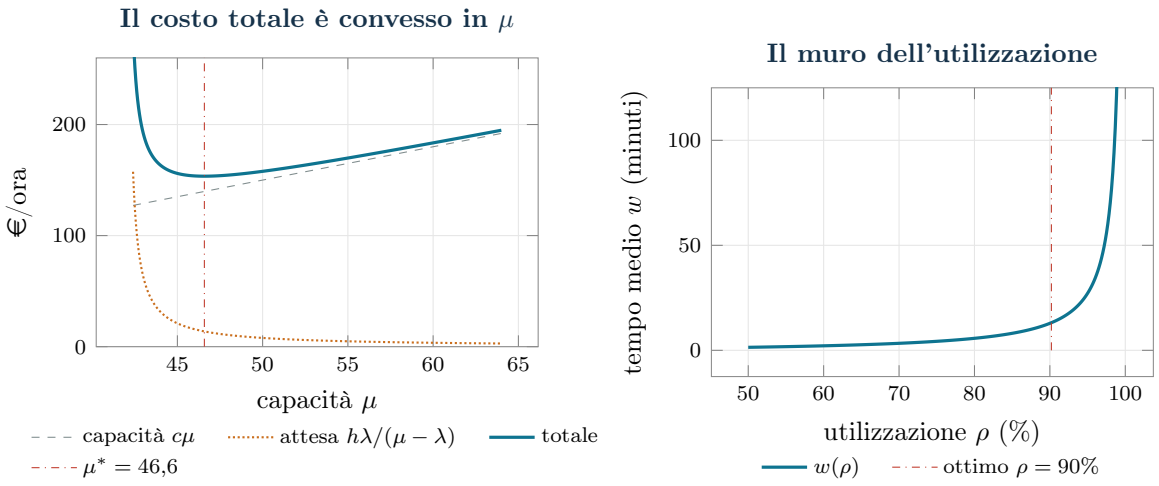


Figura 11.1: A sinistra: le due componenti del costo (capacità, crescente; attesa, decrescente) e il costo totale, convesso, con il minimo in $\mu^* = 46{,}6$. A destra: il "muro dell'utilizzazione" — il tempo medio nel sistema w in funzione dell'utilizzazione ρ: oltre il 90% l'attesa esplode in modo iperbolico.

Insight manageriale

Il grafico di destra è forse il più importante della dispensa da mostrare a un manager: tra $\rho = 90\%$ e $\rho = 99\%$ il tempo di attesa si moltiplica per dieci. "Saturare le risorse" e "dare un buon servizio" sono obiettivi in conflitto matematico, non organizzativo: nessuna riorganizzazione elimina l'iperbole $w = 1/(\mu - \lambda)$.

11.5 Analisi di sensitività: il prezzo di una promessa

Se il marketing promette "tempo medio sotto $w_{\max}$", serve $\mu \geq \lambda + 1/w_{\max}$: il vincolo diventa attivo quando è più stringente dell'ottimo economico (13,1 minuti).

```
w_max = 12 min: mu = 47,00  costo 153,60   (quasi gratis)
w_max =  9 min: mu = 48,67  costo 155,45
w_max =  6 min: mu = 52,00  costo 162,30
w_max =  4 min: mu = 57,00  costo 175,20
w_max =  3 min: mu = 62,00  costo 189,15
w_max =  2 min: mu = 72,00  costo 218,10   (+42% sul costo base)
```

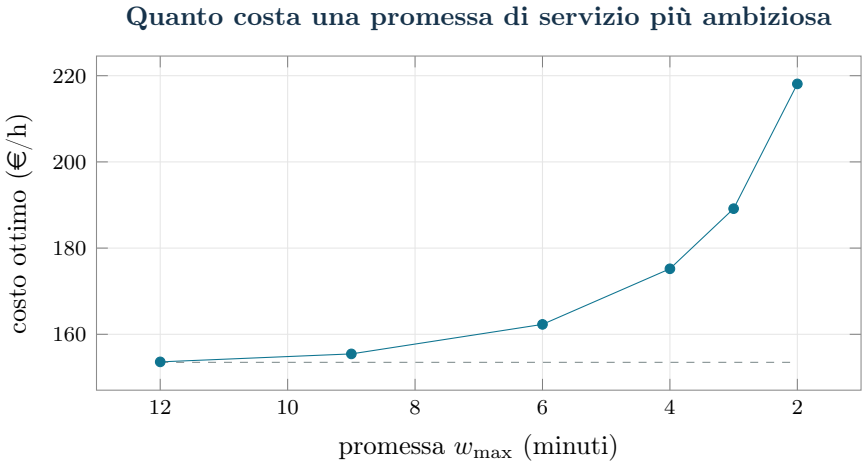


Figura 11.2: Costo ottimo in funzione della promessa di servizio $w_{\max}$ (asse invertito: verso destra la promessa diventa più ambiziosa). La linea tratteggiata è il costo senza promessa: fino a 12 minuti la promessa è quasi gratuita, sotto i 4 minuti ogni minuto in meno costa in modo esplosivo.

Il costo marginale della promessa

Il prezzo ombra del vincolo cresce in modo esplosivo: passare da 12 a 9 minuti costa 1,85 €/h, da 3 a 2 minuti costa 28,95 €/h. Quando il vincolo è attivo, $\mu = \lambda + 1/w_{\max}$ e $C = c\lambda + c/w_{\max} + h\lambda\, w_{\max}$, da cui $dC/dw_{\max} = -c/w_{\max}^2 + h\lambda$ (con $w_{\max}$ in ore): a $w_{\max} = 6$ minuti vale $-300{+}63 = -237$ €/h per ora di promessa — esattamente la stima per differenze finite dello script. È il numero da dare al marketing *prima* che firmi lo SLA (*Service Level Agreement*, l'impegno contrattuale sul livello di servizio).

Robustezza

Se λ è incerto in $[36, 48]$, dimensionare sull'ottimo nominale $\mu^* = 46{,}58$ è **instabile**: con $\lambda = 48 > \mu^*$ la coda diverge (costo infinito). Minimizzare il costo del caso peggiore dà $\mu_{\text{rob}} = 52{,}90$

con costo worst-case 173,39 €/h: si paga un premio assicurativo di $\approx 13\%$ per essere protetti da un disastro. Le decisioni di capacità vanno sempre verificate contro il *massimo* plausibile della domanda, non solo contro la media.

11.6 In sintesi

In sintesi

- Il costo totale $c\mu + h\lambda/(\mu - \lambda)$ è convesso, con ottimo $\mu^* = \lambda + \sqrt{h\lambda/c}$: la domanda più un margine di sicurezza.
- L'utilizzo ottimo non è il 100%: vicino alla saturazione il tempo di attesa esplode.
- Una promessa $w \le w_{\max}$ diventa $\mu \ge \lambda + 1/w_{\max}$, e il suo prezzo ombra esplode per promesse ambiziose.
- Dimensionare sulla domanda media può rendere la coda instabile: la soluzione robusta costa di più e protegge dal disastro.

11.7 Esercizi

Esercizio 11.1 — verifica

Ricavare per conto proprio la formula $dC/dw_{\max} = -c/w_{\max}^2 + h\lambda$ della spiegazione precedente e verificarla numericamente a $w_{\max} = 4$ e 9 minuti con differenze finite.

Esercizio 11.2 — penalità quadratica

Sostituire il costo d'attesa lineare con $h\,[1/(\mu-\lambda)]^2$: ricavare μ^* analiticamente e confrontare l'utilizzazione ottima con il caso lineare.

Esercizio 11.3 — due code vs una

Due sportelli separati ricevono $\lambda = 21$ ciascuno; in alternativa un unico sistema riceve $\lambda = 42$ con capacità aggregata. A parità di costo di capacità, quale configurazione dà attese minori? (È l'argomento quantitativo per il pooling.)

Esercizio 11.4 — frontiera capacità-servizio

Tracciare la frontiera (costo, w) variando $w_{\max}$ con continuità e individuare il punto oltre il quale ogni minuto promesso in meno costa più di 10 €/h.

Parte III

Ottimizzazione e machine learning

Capitolo 12

Support Vector Machine: ottimizzazione per il machine learning

Classe: QP convesso **Script:** `python/lab12_svm.py`

Pagina online

Prerequisiti

- Da questo modulo, Capitolo 2 «Richiami: convessità, programmazione quadratica e condizioni KKT»: i programmi quadratici convessi, le condizioni KKT e i moltiplicatori come prezzi ombra.
- Dal modulo *Ottimizzazione convessa*, Capitolo 6 «Problemi di ottimizzazione convessi e concavi»: i problemi di ottimizzazione convessi e le loro condizioni di ottimalità.
- Da questo modulo, Capitolo 3 «Il solver per i modelli non lineari»: gli obiettivi quadratici in `gurobipy`.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- formulare la classificazione a margine rigido e morbido come QP;
- ricavare e risolvere il duale, riconoscendo i vettori di supporto;
- ottenere frontiere non lineari con i kernel restando nel mondo convesso;
- gestire classi sbilanciate con costi asimmetrici e valutare con le metriche giuste.

12.1 Motivazione gestionale

La Support Vector Machine collega direttamente l'ottimizzazione convessa al machine learning: il classificatore si ottiene risolvendo un QP, con una lettura geometrica chiara. Le applicazioni gestionali sono immediate: rischio di credito, churn, manutenzione predittiva, controllo qualità, frodi. In questo capitolo *non* usiamo librerie di ML: ogni modello è un QP scritto e risolto con Gurobi, perché l'obiettivo è capire *che cosa* ottimizza un classificatore.

Nota modellistica: le etichette $y_i \in \{-1, +1\}$ sono *dati osservati*, non variabili — le variabili (coefficienti, intercetta, scarti, moltiplicatori) sono tutte continue.

Questo capitolo chiude il cerchio: gli stessi strumenti usati per produzione, logistica e finanza — QP convessi, dualità, moltiplicatori — sono il motore di uno dei classificatori più eleganti del machine learning. Capire la SVM come problema di ottimizzazione, e non come scatola nera di libreria, dà il controllo completo su ciò che il modello sta davvero facendo.

12.2 Hard margin: la geometria

Il problema a parole. *Decidiamo* i coefficienti di un iperpiano che separi due classi di osservazioni. *L'obiettivo* è massimizzare il margine, cioè la distanza tra l'iperpiano e i punti più vicini delle due classi (un classificatore con margine ampio generalizza meglio). *I vincoli* chiedono che ogni osservazione stia dal lato giusto dell'iperpiano — nel soft margin, a meno di uno scarto pagato in obiettivo.

Dati (input del modello)

Simbolo	Tipo	Significato
n	$\in \mathbb{Z}_{\geq 1}$	numero di osservazioni del training set, indicizzate da $i \in \{1, 2, \dots, n\}$
p	$\in \mathbb{Z}_{\geq 1}$	numero di caratteristiche, indicizzate da $j \in \{1, 2, \dots, p\}$
x_{ij}	$\in \mathbb{Q}$	valore della caratteristica j per l'osservazione i; $\boldsymbol{x}_i \in \mathbb{Q}^p$ è il vettore dell'osservazione i
y_i	$\in \{-1, +1\}$	etichetta *osservata* della classe di i (un dato, non una variabile)
C	$\in \mathbb{Q}_{>0}$	peso delle violazioni del margine (solo soft margin); maiuscolo per aderenza alla letteratura di machine learning, eccezione alla convenzione che riserva le maiuscole a insiemi e matrici

Variabili decisionali

Introduciamo le seguenti $p + 1$ variabili libere:

$$\begin{cases} w_j = \text{coefficiente } j \text{ dell'iperpiano separatore} \\ b = \text{intercetta dell'iperpiano} \end{cases} \qquad \forall j \in \{1, 2, \dots, p\},$$

che definiscono l'iperpiano $\sum_{j=1}^{p} w_j\, x_j + b = 0$.

Usando queste variabili, il modello QP hard margin è il seguente:

Modello QP hard margin

$$\min \quad \frac{1}{2} \sum_{j=1}^{p} w_j^2 \tag{12.1a}$$

$$\text{soggetto a} \quad y_i \Big(\sum_{j=1}^{p} w_j\, x_{ij} + b \Big) \geq 1, \qquad \forall i \in \{1, 2, \dots, n\}, \tag{12.1b}$$

$$w_j \gtreqless 0, \qquad \forall j \in \{1, 2, \dots, p\}, \tag{12.1c}$$

$$b \gtreqless 0. \tag{12.1d}$$

Descrizione della funzione obiettivo e dei vincoli del modello QP:

- la funzione obiettivo quadratica convessa (12.1a) minimizza $\frac{1}{2}\|\boldsymbol{w}\|^2$; poiché il margine geometrico tra gli iperpiani di supporto è $2/\|\boldsymbol{w}\|$, minimizzarla equivale a *massimizzare il margine*;
- i vincoli lineari (12.1b) impongono che ogni osservazione stia dal lato giusto dell'iperpiano, a distanza almeno unitaria in scala $\boldsymbol{w}$ (n vincoli lineari);
- i vincoli (12.1c) e (12.1d) definiscono le variabili del modello, tutte libere.

Due punti, tutto a mano

$\boldsymbol{x}_1 = (0,0)$ con $y_1 = -1$ e $\boldsymbol{x}_2 = (2,2)$ con $y_2 = +1$.

- **Passo 1 — vincoli.** $-(b) \geq 1$ e $2w_1 + 2w_2 + b \geq 1$, cioè $b \leq -1$ e $2(w_1 + w_2) \geq 1 - b \geq 2$.
- **Passo 2 — simmetria e ottimo.** Per minimizzare $w_1^2 + w_2^2$ con $w_1 + w_2 \geq 1$ conviene $w_1 = w_2 = t$ e prendere le disuguaglianze attive: $4t + b = 1$, $b = -1 \Rightarrow t = 1/2$. Quindi $\boldsymbol{w} = (0{,}5;\ 0{,}5)$, $b = -1$: l'iperpiano $0{,}5x_1 + 0{,}5x_2 = 1$ è l'asse del segmento tra i due punti.
- **Passo 3 — margine.** $2/\|\boldsymbol{w}\| = 2/\sqrt{0{,}5} = 2\sqrt{2} = 2{,}83$: esattamente la distanza tra i due punti. Entrambi i vincoli sono attivi: entrambi i punti sono *support vector*. Con dati reali sovrapposti, però, l'hard margin è inammissibile: serve il soft margin.

12.3 Soft margin

Con dati reali sovrapposti l'hard margin è inammissibile: si aggiungono n variabili non negative di scarto,

$$\xi_i = \text{violazione del margine dell'osservazione } i, \qquad \forall i \in \{1, 2, \ldots, n\},$$

con $\xi_i > 1$ che corrisponde a una classificazione errata.

Modello QP soft margin (primale)

$$\min \quad \frac{1}{2}\sum_{j=1}^{p} w_j^2 + C\sum_{i=1}^{n} \xi_i \tag{12.2a}$$

$$\text{soggetto a} \quad y_i\Big(\sum_{j=1}^{p} w_j\, x_{ij} + b\Big) \geq 1 - \xi_i, \qquad \forall i \in \{1, 2, \ldots, n\}, \tag{12.2b}$$

$$w_j \gtreqless 0, \qquad \forall j \in \{1, 2, \ldots, p\}, \tag{12.2c}$$

$$b \gtreqless 0, \tag{12.2d}$$

$$\xi_i \geq 0, \qquad \forall i \in \{1, 2, \ldots, n\}. \tag{12.2e}$$

Descrizione della funzione obiettivo e dei vincoli del modello QP:

- la funzione obiettivo quadratica convessa (12.2a) bilancia due termini: la semplicità del modello ($\frac{1}{2}\|\boldsymbol{w}\|^2$, margine ampio) e il costo delle violazioni sul training set ($C\sum_{i=1}^{n}\xi_i$); C è il prezzo assegnato alle violazioni;
- i vincoli lineari (12.2b) richiedono la classificazione corretta *a meno dello scarto* ξ_i, pagato in obiettivo (n vincoli lineari);
- i vincoli (12.2c), (12.2d) e (12.2e) definiscono le variabili del modello: w_j e b libere, ξ_i non negative.

Caso di studio: rischio di credito

90 clienti, due indicatori standardizzati (solidità patrimoniale, puntualità nei pagamenti), 55 affidabili e 35 insolventi, parzialmente sovrapposti (`dati/svm_clienti.csv`).

```
C =  0,05: w = (0,678, 0,474)  b = -0,148 | margine 2,418 | errori 2 | nel
    margine 22
C =  1,00: w = (1,433, 0,611)  b = -0,236 | margine 1,284 | errori 2 | nel
    margine  7
C = 20,00: w = (5,058, 2,338)  b = -2,936 | margine 0,359 | errori 0 | nel
    margine  3
```

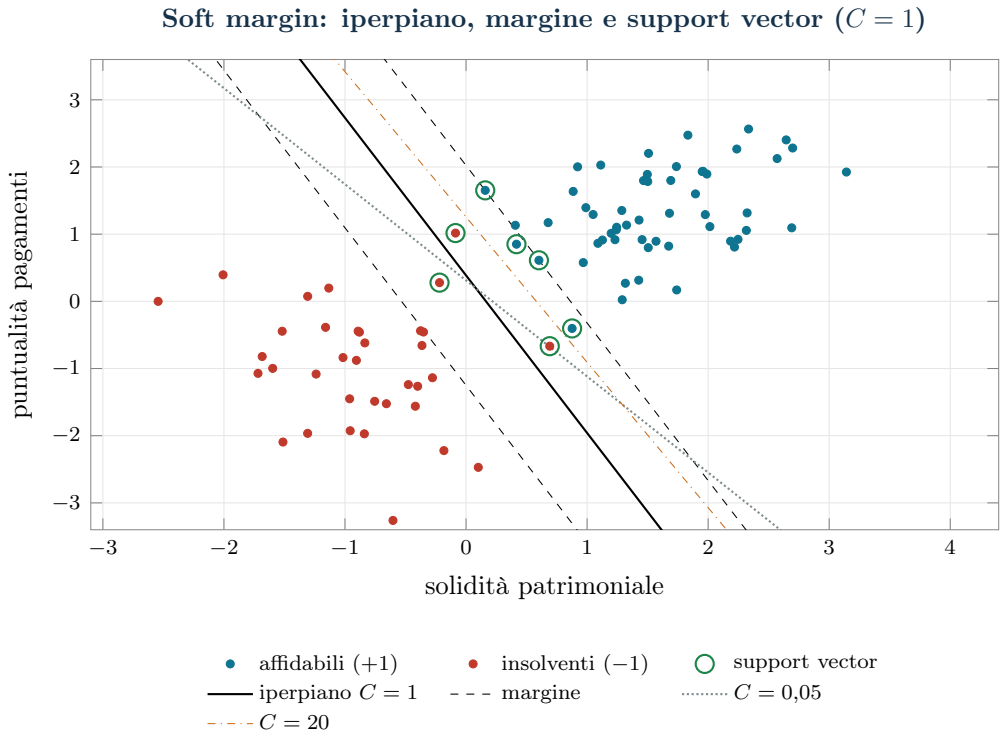


Figura 12.1: I 90 clienti nel piano dei due indicatori: affidabili (teal), insolventi (rosso), support vector cerchiati in verde. La retta nera è l'iperpiano ottimo per $C = 1$ con le sue rette di margine (tratteggiate); le rette grigia e arancione sono gli iperpiani per $C = 0{,}05$ (margine ampio) e $C = 20$ (margine stretto).

Insight manageriale

C piccolo compra un margine ampio accettando 22 punti dentro il margine; C grande insegue ogni punto del training set (zero errori, margine strettissimo): è il compromesso **regolarizzazione–aderenza**, e va scelto con dati di validazione, mai guardando il test set. Si noti che le tre rette sono simili ma non uguali: la pendenza cambia perché i punti "pagati" cambiano.

12.4 La formulazione duale e i support vector

Associando un moltiplicatore α_i a ciascun vincolo di classificazione (12.2b), si ottengono n nuove variabili:

α_i = moltiplicatore del vincolo di classificazione dell'osservazione i, $\quad \forall i \in \{1, 2, \ldots, n\}$.

Modello QP duale del soft margin

$$\max \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \, \alpha_j \, y_i \, y_j \, \boldsymbol{x}_i' \boldsymbol{x}_j \tag{12.3a}$$

$$\text{soggetto a} \quad \sum_{i=1}^{n} \alpha_i \, y_i = 0, \tag{12.3b}$$

$$\alpha_i \leq C, \qquad \forall i \in \{1, 2, \ldots, n\}, \tag{12.3c}$$

$$\alpha_i \geq 0, \qquad \forall i \in \{1, 2, \ldots, n\}. \tag{12.3d}$$

Descrizione della funzione obiettivo e dei vincoli del modello duale:

- la funzione obiettivo quadratica concava (12.3a) dipende dai dati *solo* tramite i prodotti scalari $\boldsymbol{x}_i' \boldsymbol{x}_j$: è l'osservazione che apre la porta al kernel;
- il vincolo lineare (12.3b) deriva dalla stazionarietà rispetto all'intercetta b (un vincolo lineare);
- i vincoli (12.3c) limitano ogni moltiplicatore al prezzo C delle violazioni (n vincoli lineari);
- i vincoli (12.3d) definiscono le variabili del modello.

Relazione primale-duale: $\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \, y_i \, \boldsymbol{x}_i$ e $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}'\boldsymbol{x} + b)$.

```
w dal duale = (1,433, 0,611)   b = -0,236     (identici al primale)
valore duale 4,7868 = valore primale (dualita' forte)
alpha = 0        : 83 punti  -> NON influenzano la frontiera
0 < alpha < C    :  2 punti  -> support vector esattamente sul margine
alpha = C        :  5 punti  -> dentro il margine o mal classificati
```

Perché si chiamano support vector

La soluzione dipende *solo* dai 7 punti con $\alpha_i > 0$: sono loro a "sostenere" l'iperpiano; gli altri 83 potrebbero sparire senza cambiare nulla. La complementarietà KKT organizza tutto: $\alpha_i = 0 \Rightarrow$ punto oltre il margine; $0 < \alpha_i < C \Rightarrow$ esattamente sul margine (da questi

si ricava b); $\alpha_i = C \Rightarrow$ il vincolo "paga" scarto $\xi_i > 0$. Il duale, inoltre, dipende dai dati solo tramite i prodotti scalari $\boldsymbol{x}_i'\boldsymbol{x}_j$: questa osservazione apre la porta al kernel.

12.5 Kernel: frontiere non lineari da un QP convesso

Sostituendo $\boldsymbol{x}_i'\boldsymbol{x}_j$ con un kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ — ad esempio il kernel RBF (*Radial Basis Function*) gaussiano $K(\boldsymbol{x}, \boldsymbol{z}) = e^{-\gamma\|\boldsymbol{x}-\boldsymbol{z}\|^2}$ — si classifica in uno spazio trasformato senza mai costruirlo: $f(\boldsymbol{x}) = \text{sign}\big(\sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\big)$. La frontiera nello spazio originale può essere arbitrariamente curva, ma il problema di training *resta un QP convesso* se il kernel è valido (matrice semidefinita positiva).

Caso di studio: anomalie "a corona"

90 osservazioni di processo: il funzionamento normale al centro, le anomalie disposte ad anello — nessuna retta può separarle.

```
gamma = 0,1: errori di training  1   support vector 20
gamma = 0,7: errori di training  0   support vector 15
gamma = 5,0: errori di training  0   support vector 61  <- "memorizza" i punti
```

Kernel RBF: la frontiera di livello 0 per tre valori di γ

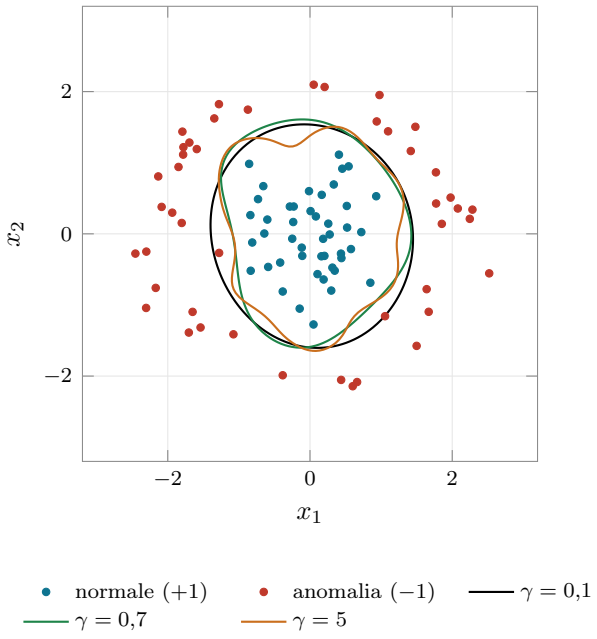


Figura 12.2: Il dataset "a corona" (funzionamento normale al centro, anomalie ad anello) e le frontiere di decisione del kernel RBF per tre valori di γ: con $\gamma = 0{,}1$ la frontiera è un cerchio regolare, con $\gamma = 5$ si frammenta intorno ai singoli punti (overfitting).

Con $\gamma = 5$ la frontiera si frammenta in isole intorno ai singoli punti e i support vector salgono a 61 su 90: il modello sta memorizzando il training set (overfitting), pur restando un QP convesso — convessità del *training* e capacità di *generalizzare* sono proprietà diverse.

C e γ vanno scelti insieme per cross-validation.

12.6 Classi sbilanciate e costi asimmetrici

Nelle applicazioni gestionali la classe importante è rara e l'errore sui rari costa di più. Basta pesare gli scarti: $C_{-}\sum_{i:\,y_i=-1}\xi_i + C_{+}\sum_{i:\,y_i=+1}\xi_i$.

```
costi uguali (C=1)          : precision 1,00   recall insolventi 0,94
costo 10x sugli insolventi  : precision 0,97   recall insolventi 1,00
```

Insight manageriale

Con il costo 10× la frontiera arretra verso la classe "sana": nessun insolvente sfugge (recall 100%) al prezzo di qualche falso allarme in più (precision dal 100% al 97%). I pesi devono riflettere i *costi decisionali* (quanto costa un fido concesso a chi non ripagherà vs. un cliente buono respinto), non le frequenze delle classi.

12.7 Support Vector Regression

La SVR stima una funzione continua ignorando gli errori entro un tubo di ampiezza ε:

$$\min \tfrac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \quad \text{con} \quad |y_i - (\boldsymbol{w}'\boldsymbol{x}_i + b)| \le \varepsilon + \xi_i^{(*)} \quad \forall i.$$

```
Retta SVR: domanda = -10,86 * prezzo + 220,32    (vera: -11p + 220)
Tubo eps = 8: 14 punti su 40 fuori dal tubo -> solo loro determinano la retta
```

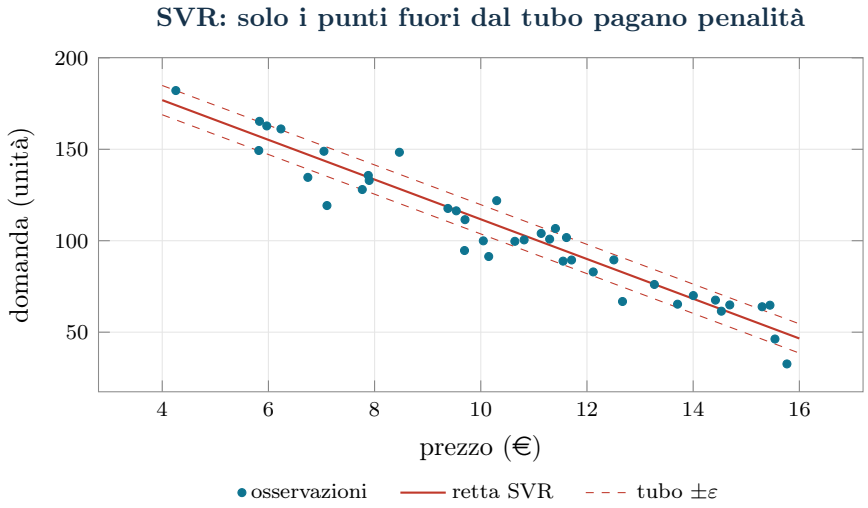


Figura 12.3: Support Vector Regression sulla relazione prezzo–domanda: le 40 osservazioni, la retta stimata e il tubo di tolleranza $\pm\varepsilon = 8$: solo i 14 punti fuori dal tubo pagano penalità e determinano la retta.

12.8 In sintesi

In sintesi

- Massimizzare il margine è un QP convesso; il margine morbido paga le violazioni con un prezzo C.
- Nel duale solo i vettori di supporto hanno moltiplicatori non nulli, e solo loro determinano la frontiera.
- Un kernel sostituisce il prodotto scalare nel duale: frontiere non lineari, sempre un QP convesso; C e γ si scelgono per convalida incrociata.
- Con classi sbilanciate due prezzi C_- e C_+ riflettono i costi delle decisioni; si valuta con precision e recall.

12.9 Esercizi

Esercizio 12.1 — verifica

Risolvere a mano l'hard margin con $\boldsymbol{x}_1 = (0, 0)$, $y_1 = -1$ e $\boldsymbol{x}_2 = (4, 0)$, $y_2 = +1$, e verificare con Gurobi ($\boldsymbol{w} = (0{,}5;\ 0)$, $b = -1$, margine 4).

Esercizio 12.2 — complementarietà

Per $C = 1$, elencare dallo script i 7 punti con $\alpha_i > 0$ e verificare numericamente le tre condizioni di complementarietà ($\alpha_i = 0 \Rightarrow \xi_i = 0$ e margine ≥ 1; ecc.).

Esercizio 12.3 — curva di validazione

Dividere i 90 clienti in training (60) e validazione (30); tracciare l'errore di validazione per $C \in \{0{,}01;\ 0{,}1;\ 1;\ 10;\ 100\}$ e scegliere C. L'errore di training è monotono in C? E quello di validazione?

Esercizio 12.4 — soglia decisionale

A parità di modello ($C = 1$), spostare la soglia: classificare insolvente se $\boldsymbol{w}'\boldsymbol{x} + b < \theta$ con $\theta \in [-1, 1]$. Tracciare precision e recall in funzione di θ: dove sta il punto che massimizza il profitto se un insolvente non individuato costa 10 e un falso allarme 1?

Esercizio 12.5 — one-class

Implementare la One-Class SVM sul dataset a corona usando solo i punti normali: $\min \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \xi_i - \rho$ soggetto a $\boldsymbol{w}'\phi(\boldsymbol{x}_i) \geq \rho - \xi_i$ (kernel RBF, $\gamma = 0{,}7$). Quale frazione di anomalie individua per $\nu \in \{0{,}05;\ 0{,}1;\ 0{,}2\}$?

Capitolo 13

Regressione ai minimi quadrati: stimare i parametri con un QP

Classe: QP convesso **Script:** `python/lab13_minimi_quadrati.py`
Pagina online

Prerequisiti

- Dal modulo *Modellazione PL*, Capitolo 10 «Regressione robusta e quantile: stimare i parametri con un LP»: la regressione con scarti assoluti e quantili come LP, sugli stessi dati.
- Da questo modulo, Capitolo 2 «Richiami: convessità, programmazione quadratica e condizioni KKT»: i programmi quadratici convessi, le condizioni KKT e i moltiplicatori come prezzi ombra.
- Dal modulo *Algebra lineare*, Capitolo 6 «Sistemi di equazioni lineari»: i sistemi lineari, da cui vengono le equazioni normali.

Obiettivi

Alla fine di questo capitolo sarai in grado di:

- scrivere i minimi quadrati come QP;
- leggere le condizioni KKT come equazioni normali;
- confrontare minimi quadrati, scarti assoluti e quantili sugli stessi dati;
- restringere i coefficienti con una penalità quadratica (ridge).

13.1 Che cosa cambia rispetto al modello lineare

Il caso è quello di Modellazione PL, «Regressione robusta e quantile»: sessanta giorni di vendite di una panetteria, con temperatura, prezzo e weekend, cinque dei quali durante due scioperi dei trasporti. Dati e descrizione sono gli stessi. Là la retta si stima minimizzando la somma degli scarti *assoluti*, con un LP.

Qui si minimizza la somma degli scarti *al quadrato*: è la regressione dei minimi quadrati, la più usata in pratica, ed è un QP convesso. Il confronto con le regressioni robusta e quantile mostra che cosa si guadagna e che cosa si perde cambiando l'obiettivo.

13.2 Il modello

Che cosa cambia. I dati sono quelli del modello lineare: n osservazioni indicizzate da i, p caratteristiche indicizzate da j, valori x_{ij} e valori osservati y_i. Il quantile τ non serve più. Introduciamo $p+1$ variabili libere per la funzione da stimare, w_j (coefficiente della caratteristica j) e b (intercetta), e una variabile libera per ogni osservazione:

$$r_i = \text{residuo dell'osservazione } i, \qquad \forall i \in \{1, 2, \dots, n\}.$$

Usando queste variabili, il modello QP dei minimi quadrati è il seguente:

Modello QP dei minimi quadrati

$$\min \quad \sum_{i=1}^{n} r_i^2 \tag{13.1a}$$

$$\text{soggetto a} \quad \sum_{j=1}^{p} w_j\, x_{ij} + b + r_i = y_i, \qquad \forall i \in \{1, 2, \dots, n\}, \tag{13.1b}$$

$$w_j \gtreqless 0, \qquad \forall j \in \{1, 2, \dots, p\}, \tag{13.1c}$$

$$b \gtreqless 0, \tag{13.1d}$$

$$r_i \gtreqless 0, \qquad \forall i \in \{1, 2, \dots, n\}. \tag{13.1e}$$

Descrizione della funzione obiettivo e dei vincoli del modello QP:

- la funzione obiettivo quadratica (13.1a) somma i quadrati dei residui;
- i vincoli lineari (13.1b) definiscono il residuo di ogni osservazione come differenza fra il dato e la previsione (n vincoli lineari);
- i vincoli (13.1c), (13.1d) e (13.1e) definiscono le variabili del modello, tutte libere.

Una variabile per residuo, senza parti positive

Nel modello lineare ogni scostamento si spezza in una parte positiva e una negativa, per pagarne il valore assoluto. Il quadrato non distingue il segno: basta un residuo libero r_i. L'obiettivo $\sum_i r_i^2$ è convesso, i vincoli sono lineari: le KKT sono necessarie e sufficienti.

Senza caratteristiche: torna la media

Con $p = 0$ resta la sola intercetta. Sette giorni di vendite: 88, 96, 104, 112, 120, 132, 148 pezzi (gli stessi dell'esempio del modello lineare).

- **Passo 1 — risolvendo si ottiene**

  $$\tilde{b} = 114{,}2857 = \tfrac{800}{7}, \qquad \tilde{z} = 2619{,}4286,$$

  con residui $\tilde{r} = (-26{,}29;\ -18{,}29;\ -10{,}29;\ -2{,}29;\ 5{,}71;\ 17{,}71;\ 33{,}71)$ e moltiplicatori $\tilde{\pi}_i = 2\,\tilde{r}_i$.

- **Passo 2 — stazionarietà rispetto a b.** La Lagrangiana $\sum_i r_i^2 - \sum_i \pi_i(b + r_i - y_i)$ dà $2r_i = \pi_i$ e $\sum_i \pi_i = 0$, quindi $\sum_i r_i = 0$: la stima è la **media** $800/7$, e i residui si compensano.

- **Passo 3 — il confronto.** Sugli stessi sette numeri la regressione robusta ($\tau = 1/2$) dà la mediana 112, quella quantile con $\tau = 9/13$ dà 120.
- **Passo 4 — un valore anomalo.** Se l'ultimo valore passa da 148 a 248, la media sale di 14,29 pezzi (a 128,57), la mediana resta 112.

13.3 Le KKT sono le equazioni normali

Con le caratteristiche la stazionarietà della Lagrangiana rispetto a r_i, b e w_j dà

$$\pi_i = 2\,r_i, \qquad \sum_{i=1}^{n} r_i = 0, \qquad \sum_{i=1}^{n} r_i\,x_{ij} = 0 \quad \forall j \in \{1, 2, \dots, p\}.$$

I residui ottimi hanno somma nulla e sono ortogonali a ogni caratteristica: sono le **equazioni normali** della statistica, ottenute qui come condizioni KKT. Il moltiplicatore $\tilde{\pi}_i$ dice quanto cresce la somma dei quadrati se il dato y_i aumenta di un'unità: il doppio del residuo. Nel modello lineare i duali valevano $\pm 1/2$ o un valore interno; qui sono proporzionali ai residui, e un dato lontano pesa tanto quanto è lontano.

13.4 Caso di studio: minimi quadrati contro scarti assoluti

Stimiamo la relazione con la sola temperatura ($p = 1$) sui 60 giorni.

```
QP (minimi quadrati)  : domanda = 202,34 - 3,61 * temperatura
LP (scarti assoluti)  : domanda = 209,77 - 3,63 * temperatura
QP senza gli scioperi : domanda = 213,58 - 3,72 * temperatura
Somma dei quadrati 68.286,6   errore quadratico medio (radice) 33,74 pezzi
Scarto assoluto medio sui 55 giorni normali: QP 14,81  LP 13,08  QP pulito
    13,05
Quota della somma dei quadrati dovuta ai 5 giorni di sciopero: 71,4%
Verifica KKT: somma dei residui 0,000; somma residui * temperatura 0,000
```

I quadrati danno voce agli anomali

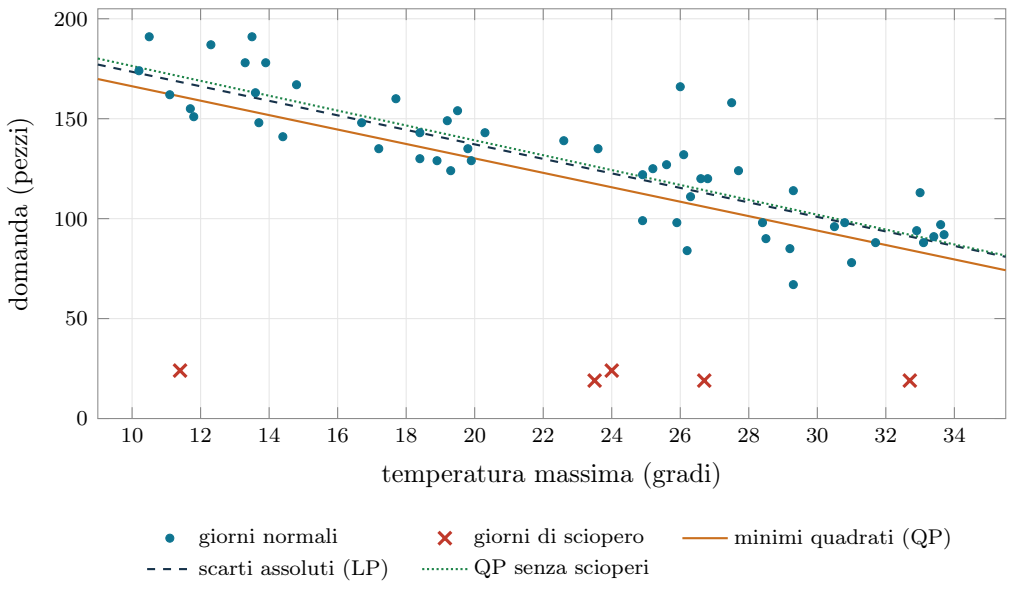


Figura 13.1: I 60 giorni della panetteria: i cinque giorni di sciopero tirano verso il basso la retta dei minimi quadrati (arancione) di oltre 11 pezzi sull'intercetta; la retta degli scarti assoluti (blu) resta vicina a quella dei minimi quadrati calcolata senza gli scioperi (verde).

Insight manageriale

Cinque giorni su sessanta producono il 71% della somma dei quadrati: residui di -137 pezzi pesano come migliaia di residui piccoli. I minimi quadrati vanno quindi accompagnati da un controllo degli anomali; la regressione robusta di Modellazione PL non ne ha bisogno. In cambio, sui dati puliti i minimi quadrati hanno una soluzione in forma chiusa (le equazioni normali) e stimano la media, che è la grandezza giusta quando gli errori sono simmetrici.

13.5 Previsione per il newsvendor: media più scorta o quantile

Sui 55 giorni normali, con tre caratteristiche (temperatura, prezzo, weekend):

```
Minimi quadrati: domanda = 272,21 - 3,84 temp - 22,52 prezzo + 31,57 weekend
                 (deviazione standard dei residui 9,40)
Quantile 9/13  : domanda = 281,19 - 3,96 temp - 23,21 prezzo + 31,47 weekend
Giorno tipo (28 gradi, prezzo 3,00, weekend):
  media prevista (QP)                  128,7 pezzi
  media + z*sigma, z = 0,5024          133,5 pezzi
  quantile 9/13 previsto (LP)          132,1 pezzi
```

Due strade per la stessa quantità

Al newsvendor serve il quantile 9/13 della domanda. I minimi quadrati danno la media; per arrivare al quantile si aggiunge una scorta $z\,\sigma$, con z il quantile della normale standard, supponendo residui normali. La regressione quantile stima il quantile direttamente, senza ipotesi sulla forma degli errori. Qui le due strade differiscono di 1,4 pezzi; sui 55 giorni la quota sotto la stima "media $+\,z\sigma$" è 0,709, vicina all'obiettivo 0,692.

13.6 Analisi di sensitività

Un solo giorno anomalo

Spostiamo verso il basso la domanda del giorno normale più caldo (33,7 gradi, domanda 92) e ristimiamo sui 55 giorni normali.

spostamento	QP: b	w	LP: b	w
0	213,58	-3,723	210,49	-3,578
-50	217,09	-3,918	211,63	-3,689
-100	220,59	-4,113	211,63	-3,689
-200	227,60	-4,503	211,63	-3,689

Insight manageriale

La stima dei minimi quadrati si sposta *in proporzione* all'errore: ogni pezzo in meno nel giorno anomalo cambia la pendenza di circa 0,004. La stima LP cambia una volta, quando il punto passa sotto la retta, poi non si muove più: oltre quel punto conta solo il segno del residuo. Un giorno caldo, lontano dal centro dei dati, ha una *leva* alta sui minimi quadrati.

La regressione ridge

Aggiungendo all'obiettivo $\lambda \sum_{j=1}^{p} w_j^2$ si ottiene la regressione *ridge*, ancora un QP convesso. Su sette caratteristiche standardizzate (le tre utili e quattro senza legame con la domanda):

lambda	temp	prezzo	weekend	follower	pioggia	borsa	giorno	rmse	nulli
0	-27,34	-9,45	11,67	1,44	-2,44	0,72	-1,86	8,40	0
30	-17,44	-5,18	6,61	1,64	-2,77	0,72	-1,15	13,96	0
300	-4,15	-1,13	1,34	0,61	-1,00	0,27	-0,26	27,11	0
3000	-0,48	-0,13	0,15	0,08	-0,13	0,04	-0,03	31,02	0

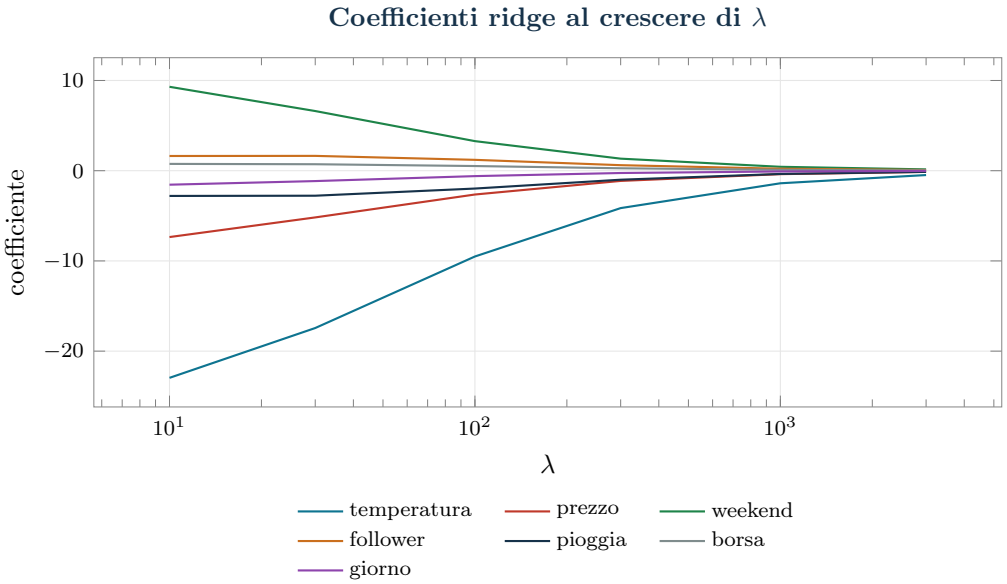


Figura 13.2: I coefficienti ridge si restringono tutti verso zero al crescere di λ, ma nessuno si annulla.

Insight manageriale

La penalità quadratica restringe i coefficienti ma non li azzera mai: la sua derivata vale $2\lambda w_j$, che si annulla proprio in $w_j = 0$, e quindi non c'è un "costo fisso" che spinga un coefficiente esattamente a zero. Il budget sui valori assoluti di Modellazione PL, invece, lascia fuori le caratteristiche inutili. Per *selezionare* si usa la penalità lineare; per *stabilizzare* stime con caratteristiche correlate, quella quadratica.

13.7 In sintesi

In sintesi

- I minimi quadrati sono un QP convesso, e le loro condizioni KKT sono le equazioni normali.
- Stimano la media: ottimi con errori simmetrici, fragili con gli outlier, dove gli scarti assoluti tengono meglio.
- Per il newsvendor si usa la media più una scorta, oppure direttamente il quantile.

- Il ridge stabilizza i coefficienti; un budget lineare su di essi seleziona le variabili esplicative.

13.8 Esercizi

Esercizio 13.1 — a mano

Con le sette vendite dell'esempio, calcolare a mano la somma dei quadrati dei residui per $b = 112$ (la mediana) e per $b = 800/7$ (la media). Di quanto è più grande la prima?

Esercizio 13.2 — equazioni normali

Scrivere e risolvere le equazioni normali $\boldsymbol{A}'\boldsymbol{A}\,\boldsymbol{c} = \boldsymbol{A}'\boldsymbol{y}$ per la retta con la sola temperatura sui 55 giorni normali, e controllare che coincidano con la soluzione del QP.

Esercizio 13.3 — pesi

Minimizzare la somma pesata $\sum_i \omega_i r_i^2$ con $\omega_i = 0{,}01$ nei giorni di sciopero e 1 negli altri, sui 60 giorni con la sola temperatura. Dove finisce la retta?

Esercizio 13.4 — Huber

Sostituire il quadrato con la perdita di Huber: r_i^2 per $|r_i| \le 20$ e $40|r_i| - 400$ oltre. Scriverla come QP con $r_i = q_i + u_i - v_i$, $|q_i| \le 20$, obiettivo $\sum_i (q_i^2 + 40u_i + 40v_i)$, e stimare la retta con la sola temperatura sui 60 giorni.

Esercizio 13.5 — validazione

Stimare minimi quadrati e scarti assoluti con le tre caratteristiche sui primi 40 giorni (scioperi compresi) e confrontare lo scarto assoluto medio sui 20 giorni restanti, esclusi gli scioperi.

Parte IV

Organizzazione del corso

Capitolo 14

Organizzazione del corso

14.1 Percorso essenziale in quattro laboratori

	Contenuto	Obiettivi di apprendimento
Lab 1	Markowitz (cap. 6)	costruire un QP convesso; tracciare una frontiera; discutere la fragilità delle stime
Lab 2	Produzione con smoothing (cap. 4)	aggiungere un termine quadratico a un modello lineare; leggere i moltiplicatori di un QP; verificarli per perturbazione
Lab 3	Pricing *oppure* budget (capp. 7–8)	modellare funzioni non lineari; studiare concavità; verificare le KKT numericamente
Lab 4	Progetto a scelta	supply chain, localizzazione, ricarica EV, code, SVM o minimi quadrati; presentazione manageriale dei risultati

I capitoli 12–13 possono sostituire il laboratorio 3 in un'edizione orientata al machine learning, oppure diventare approfondimenti per progetti individuali.

14.2 Errori ricorrenti da evitare

1. Leggere `.X` o `.Pi` senza controllare `m.Status`.
2. Dimenticare `lb=-GRB.INFINITY` sulle variabili libere (b della SVM, i residui dei minimi quadrati).
3. Usare il moltiplicatore di un QP per un incremento grande: è esatto solo al margine, perché il valore ottimo ha curvatura.
4. Dimenticare `NonConvex = 2` con termini bilineari, o credere convesso un problema che non lo è.
5. Aggiornare il RHS di un vincolo che contiene costanti a sinistra.
6. Ottimizzare un solo obiettivo quando il problema ne ha due (costo e regolarità, rischio e rendimento).
7. Scegliere gli iperparametri (C, γ, λ) guardando il test set.

8. Riportare sei cifre decimali da stime che ballano alla seconda.

14.3 Materiali del corso

`python/`	uno script per capitolo; `esegui_tutti.py` rigenera dati, figure e notebook
`notebooks/`	gli stessi script come notebook, da aprire in Colab
`dati/`	tutti i dati dei casi di studio in CSV
guida al solver	in Modellazione PL
`docs/pdf/`	questa dispensa e le slide dei capitoli

Riproducibilità: i due comandi che rifanno dati, figure e notebook.

```
python3 -m pip install gurobipy matplotlib pandas
python3 python/esegui_tutti.py
```