

SAPIENZA
UNIVERSITÀ DI ROMA

NLP Modelling

Nonlinear programming models

Lecture notes with models, fully worked numerical examples, case studies, complete Python/Gurobi implementations and sensitivity analysis. Quadratic and nonlinear programming: convexity and KKT conditions, management cases (portfolio, prices, budget, location, queues, supply chain), quadratic variants of linear models and optimization for machine learning (SVM, least squares).

Fabio Furini
fabiofurini.github.io
Department of Computer, Control and Management Engineering — Sapienza University of Rome

Online version and downloadable material:
fabiofurini.github.io/nlp-modelling/
github.com/fabiofurini/nlp-modelling

Academic year 2026–2027

Contents

I Tools **4**

1 Introduction to the course **5**
1.1 What we will do . . . 5
1.2 The classes of models and the acronyms used . . . 5
1.3 Map of the models . . . 6
1.4 Organization of the course . . . 6
1.5 In summary . . . 6

2 Background: convexity, QP and KKT conditions **7**
2.1 Convexity and quadratic programming . . . 7
2.2 The optimality conditions: the KKT conditions . . . 8
2.3 In summary . . . 10

3 The solver for nonlinear models **11**
3.1 Functional constraints and bilinear terms . . . 11
3.2 Tolerances and scaling . . . 12
3.3 Multipliers in convex QPs . . . 12
3.4 When the duals are not there . . . 12
3.5 In summary . . . 12

II Nonlinear deterministic models **14**

4 Production with quadratic smoothing **15**
4.1 What changes with respect to the linear model . . . 15
4.2 The model . . . 16
4.3 Worked example . . . 17
4.4 Implementation . . . 17
4.5 Results . . . 17
4.6 Sensitivity analysis . . . 18
4.7 In summary . . . 19
4.8 Exercises . . . 20

5 Supply chain with congestion and sustainability **21**
5.1 Managerial motivation . . . 21
5.2 Guided construction of the model . . . 22
5.3 Worked example by hand . . . 23

5.4 Case study . . . 23
5.5 Results . . . 24
5.6 Sensitivity analysis: the price of CO_2 . . . 25
5.7 In summary . . . 26
5.8 Exercises . . . 26

6 The Markowitz portfolio **27**
6.1 Managerial motivation . . . 27
6.2 Guided construction of the model . . . 28
6.3 Worked example by hand . . . 29
6.4 Case study . . . 29
6.5 Results . . . 30
6.6 Sensitivity analysis . . . 31
6.7 In summary . . . 32
6.8 Exercises . . . 32

7 Pricing and revenue management **33**
7.1 Managerial motivation . . . 33
7.2 Guided construction of the model . . . 34
7.3 Worked example by hand . . . 35
7.4 Case study and implementation . . . 35
7.5 Multi-product version . . . 37
7.6 In summary . . . 38
7.7 Exercises . . . 38

8 Advertising budget allocation **39**
8.1 Managerial motivation . . . 39
8.2 Guided construction of the model . . . 40
8.3 Worked example by hand . . . 41
8.4 Case study and implementation . . . 41
8.5 Results and verification of the KKT conditions . . . 42
8.6 Sensitivity analysis . . . 43
8.7 In summary . . . 43
8.8 Exercises . . . 44

9 Continuous location of a service **45**
9.1 Managerial motivation . . . 45
9.2 Guided construction of the model . . . 46
9.3 Worked example by hand . . . 47
9.4 Case study . . . 47
9.5 Results . . . 48
9.6 Sensitivity analysis: the efficiency-equity frontier . . . 49
9.7 In summary . . . 49
9.8 Exercises . . . 50

10 Electric vehicle charging with a smooth profile **51**
10.1 What changes with respect to the linear model . . . 51
10.2 The model . . . 51
10.3 Worked example . . . 53
10.4 Implementation . . . 53

10.5 Results . . . 53
10.6 Sensitivity analysis . . . 54
10.7 In summary . . . 55
10.8 Exercises . . . 55

11 Service capacity and waiting times **57**
11.1 Managerial motivation . . . 57
11.2 Guided construction of the model . . . 58
11.3 Worked example by hand . . . 59
11.4 Case study and results . . . 59
11.5 Sensitivity analysis: the price of a promise . . . 60
11.6 In summary . . . 61
11.7 Exercises . . . 61

III Optimization and machine learning **62**

12 Support Vector Machine **63**
12.1 Managerial motivation . . . 63
12.2 Hard margin: the geometry . . . 64
12.3 Soft margin . . . 65
12.4 The dual formulation and the support vectors . . . 67
12.5 Kernels: non-linear boundaries from a convex QP . . . 67
12.6 Unbalanced classes and asymmetric costs . . . 68
12.7 Support Vector Regression . . . 69
12.8 In summary . . . 69
12.9 Exercises . . . 70

13 Least-squares regression **71**
13.1 What changes with respect to the linear model . . . 71
13.2 The model . . . 72
13.3 The KKT conditions are the normal equations . . . 73
13.4 Case study: least squares against absolute deviations . . . 73
13.5 Forecast for the newsvendor: mean plus stock or quantile . . . 74
13.6 Sensitivity analysis . . . 75
13.7 In summary . . . 76
13.8 Exercises . . . 76

IV Organization of the course **77**

14 Organization of the course **78**
14.1 Essential path in four laboratory sessions . . . 78
14.2 Recurring mistakes to avoid . . . 78
14.3 Course materials . . . 79

Part I

Tools

Chapter 1

Introduction to the course

Prerequisites

- From the module *LP Modelling*, Chapter 2 «Background: linear programming and sensitivity analysis»: linear models, duality and the sensitivity protocol.
- From the module *Convex Optimization*, Chapter 6 «Convex and concave optimization problems»: convex sets and functions, and convex optimization problems.

Objectives

At the end of this chapter you will be able to:

- describe the path of every chapter: management problem, model, Python and solver, interpretation;
- tell the classes QP and NLP apart, and say when a model is convex;
- find your way in the map of the models of the course.

1.1 What we will do

This course teaches how to use nonlinear programming as a *decision* tool, not only as a computing tool. Every chapter starts from a concrete management problem — how much risk to accept, which price to set, where to locate a service, how smooth a plan should be — and turns it into an optimization model, which is implemented in Python, solved with a solver and, above all, *questioned*: how much is one more unit of a resource worth? How much does a smoother plan cost? Does the solution hold if the data change?

> **The right question**
>
> At the end of every session the question is not only *"what is the optimum?"*, but *"which decision do we recommend and how robust is it?"*. The value of a model lies in the quality of the recommendation it allows us to make.

1.2 The classes of models and the acronyms used

Throughout these lecture notes the models belong to two classes, denoted by the standard international acronyms:

- **QP** (*Quadratic Programming*): quadratic objective, linear constraints;
- **NLP** (*Nonlinear Programming*): general nonlinear objective or constraints.

The acronym **LP** (*Linear Programming*) denotes linear models, which appear here only as a term of comparison. A problem is *convex* when every local minimum is also global (precise definition in chapter 2): for QP and NLP it depends on the functions, and we will point it out every time.

Notation, model format and chapter outline (motivation, model, worked example, case study, code, results, sensitivity, exercises) are the same as in LP Modelling. The numbers of every chapter are generated by the scripts in the `python/` folder (`python3 run_all.py`).

1.3 Map of the models

Ch.	Application	Class	Central concept
4	Production with quadratic smoothing	convex QP	smoothness of the plan
5	Supply chain and sustainability	convex NLP	congestion, CO_2
6	Markowitz portfolio	convex QP	risk-return
7	Pricing and revenue management	NLP	elasticity and capacity
8	Advertising budget	convex NLP	diminishing returns
9	Continuous location	convex NLP	efficiency-equity
10	EV charging with a smooth profile	convex QP	jitter of the draw
11	Capacity and waiting times	convex NLP	capacity-waiting
12	Support Vector Machine	convex QP	optimization for ML
13	Least-squares regression	convex QP	mean, outliers, ridge

Chapters 4, 10 and 13 are the quadratic variants of three cases that stay linear in LP Modelling: same data, different objective.

1.4 Organization of the course

The essential path consists of four laboratory sessions (chapter 14 for the details): (1) Markowitz and the efficient frontier; (2) production with smoothing and the multipliers of a QP; (3) nonlinear modelling (pricing or budget); (4) a project of your choice with a management presentation.

1.5 In summary

In summary

- Every chapter turns a management problem into a nonlinear model, solves it with Gurobi from Python and questions it.
- QP: quadratic objective and linear constraints; NLP: general nonlinear objective or constraints.
- Convexity decides whether a local optimum is global: for QP and NLP it depends on the functions, and it is pointed out every time.
- Notation, model format and chapter outline are those of LP Modelling.

Chapter 2

Background: convexity, quadratic programming and KKT conditions

Prerequisites

- From the module *Convex Optimization*, Chapter 3 «Convex and concave functions»: convex and concave functions and their second-order characterisation.
- From the module *Convex Optimization*, Chapter 4 «Quadratic functions»: quadratic functions and positive semidefinite matrices.
- From the module *LP Modelling*, Chapter 2 «Background: linear programming and sensitivity analysis»: LP duality, complementary slackness and shadow prices.

Objectives

At the end of this chapter you will be able to:

- recognise a convex problem and use the fact that every local minimum is global;
- decide whether a QP is convex from the matrix $\boldsymbol{Q}$;
- write the KKT conditions of a problem and use them to check a solution;
- read the multipliers as shadow prices and estimate them by perturbation.

This chapter collects the theoretical tools used throughout these lecture notes: convexity and quadratic programming and the KKT conditions. It also serves as a quick reference.

What is taken as known

Linear programming theory (duality, complementary slackness, shadow prices) is recalled in LP Modelling, "Theory: linear programming" and developed in full in the course Linear Programming. The sensitivity analysis protocol followed in every chapter is described in LP Modelling. Convex sets and functions are covered in the course Convex Optimization.

2.1 Convexity and quadratic programming

A problem $\min f(x_1, x_2, \ldots, x_n)$ over a feasible set X is **convex** if f is convex and X is convex. The property we shall use all the time:

Why convexity matters

In a convex problem every local minimum is global: optimality can be *certified*. In non-convex problems (for instance a pricing problem with a bilinear term $p \cdot q$) a local method guarantees only a local optimum; non-convex QPs can nonetheless be solved to global optimality with dedicated techniques, but the computational cost grows.

A **QP** has objective $\frac{1}{2}\sum_i \sum_j q_{ij}\, x_i x_j + \sum_j c_j x_j$ and linear constraints; it is convex if and only if the matrix $\boldsymbol{Q} = (q_{ij})$ is positive semidefinite ($\boldsymbol{Q} \succeq 0$). The example that follows shows one on a small scale; in these lecture notes QPs appear every time a cost or a risk grows more than linearly with the decision.

A 2×2 QP, worked out in full

Consider a minimization QP with a separable quadratic objective (the second variable "costs" twice as much as the first, at the margin) and a threshold constraint on the sum:

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

Here $\boldsymbol{Q} = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix} \succ 0$: the QP is convex and the optimum found is global.

- **Step 1 — solving we obtain:**

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

- **Step 2 — verification of the shadow price.** The constraint is active ($4 + 2 = 6$: the objective grows with the variables, hence the sum does not exceed what is needed) and the partial derivatives coincide:

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

  If they were different it would be worth shifting quantity from one variable to the other; the common derivative is precisely the multiplier.

- **Step 3 — verification by perturbation.** With right-hand side 7 we obtain $(\tilde{x}_1, \tilde{x}_2) = (14/3,\ 7/3)$ with

$$\tilde{f} = 32.67 \approx 24 + 8$$

  (on top of that there is the curvature term: here the shadow price is exact only at the margin, not for finite increments).

2.2 The optimality conditions: the KKT conditions

The optimality conditions of constrained nonlinear problems are known as **KKT conditions** (after Karush, Kuhn and Tucker, who formulated them). For the problem $\min f(x_1, \ldots, x_n)$ subject to $g_i(x_1, \ldots, x_n) \leq 0$ for every $i \in \{1, 2, \ldots, m\}$ and $h_j(x_1, \ldots, x_n) = 0$ for every

$j \in \{1, 2, \ldots, q\}$, the Lagrangian is

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

KKT conditions

At a regular optimal point $(\tilde{x}_1, \ldots, \tilde{x}_n)$ there exist multipliers $\lambda_i \geq 0$ and ν_j such that

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

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

If the problem is convex, the KKT conditions are also *sufficient*: a point that satisfies them is a global optimum.

The KKT conditions are complementary slackness, generalized

For an LP the KKT conditions reduce *exactly* to the optimality conditions of linear programming (LP Modelling, "Theory: linear programming"): the stationarity of the Lagrangian is equivalent to the feasibility of the dual (the multipliers λ_i are the dual variables π_i, and the multipliers of the sign constraints are the reduced costs), and the complementarity $\lambda_i \, g_i(\tilde{x}_1, \ldots, \tilde{x}_n) = 0$ is the complementary slackness condition $\bar{\pi}_i \cdot \bar{s}_i = 0$. Complementary slackness is the KKT conditions in the linear case.

The KKT conditions on the QP of the example, worked out in full

Let us take up again the QP just solved: min $x_1^2 + 2x_2^2$ subject to $x_1 + x_2 \geq 6$, $x_1, x_2 \geq 0$.

- **Step 1.** Let us rewrite the constraint in standard form: $g(x_1, x_2) = 6 - x_1 - x_2 \leq 0$.
- **Step 2.** Stationarity of the Lagrangian $L = x_1^2 + 2x_2^2 + \lambda\,(6 - x_1 - x_2)$:

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

  (the constraints $x_1, x_2 \geq 0$ will turn out to be inactive: their multipliers are null by complementarity).
- **Step 3.** Complementarity: if the threshold constraint were inactive ($\lambda = 0$) we would have $x_1 = x_2 = 0$, which violates $x_1 + x_2 \geq 6$: impossible. Hence the constraint is active: $\lambda/2 + \lambda/4 = 6 \Rightarrow \lambda = 8$, $\tilde{x}_1 = 4$, $\tilde{x}_2 = 2$, $\tilde{f} = 24$ — the same numbers found by elementary means in the QP example, now obtained with the general procedure.
- **Step 4.** Reading of the multiplier: with right-hand side $6 + \varepsilon$ the optimal value grows by about $\lambda\varepsilon = 8\varepsilon$. Verification: with right-hand side d the solution is $\tilde{x}_1 = 2d/3$, $\tilde{x}_2 = d/3$, hence

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

Line-by-line explanation

The KKT multipliers generalize the shadow prices of the LP: they measure the marginal variation of the optimal value with respect to the right-hand side of the constraint. In the laboratory, when the duals are not available (general nonlinear models), we estimate the multiplier *by perturbation*: the right-hand side is increased by ε, the problem is re-optimized and the incremental ratio is computed. This numerical verification is itself a recurring exercise.

2.3 In summary

In summary

- In a convex problem every local minimum is global; a QP is convex if and only if $\boldsymbol{Q} \succeq 0$.
- The KKT conditions are stationarity of the Lagrangian, feasibility and complementarity; in convex problems they are sufficient.
- In the linear case the KKT conditions reduce to complementary slackness.
- The multipliers are shadow prices, and they can also be estimated by perturbation.

Exercise 2.1 — KKT

Solve with the KKT conditions: $\min\ (x-3)^2 + (y-1)^2$ subject to $x + 2y \le 2$, $x, y \ge 0$. Say which constraints are active at the optimum and interpret the multipliers.

Exercise 2.2 — convexity

For which values of the parameter γ is the function $f(x_1, x_2) = x_1^2 + \gamma x_1 x_2 + x_2^2$ convex? (Write Q and impose $Q \succeq 0$.)

Chapter 3

The solver for nonlinear models

Prerequisites

- From the module *LP Modelling*, Chapter 3 «The solver: building, solving, interpreting»: the general `gurobipy` commands: building, solving, reading the solution.
- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex QPs and the KKT multipliers.

Objectives

At the end of this chapter you will be able to:

- write nonlinear functions with functional constraints $y = f(x)$ and ratios with bilinear constraints;
- choose tolerances and reformulations that give good scaling;
- read the multipliers of a convex QP with `Pi`;
- estimate a shadow price by perturbation when the solver gives no duals.

This chapter collects the `gurobipy` commands specific to nonlinear models: functional and bilinear constraints, tolerances, multipliers of QPs. The general commands (installation and licence, the five steps to build a model, the log, the attributes of the solution) are in LP Modelling, "Implementation: linear models" and in the solver guide of that course.

A single solver for the whole course: Gurobi covers LP and QP natively (including non-convex ones, with `m.Params.NonConvex = 2`) and, from version 12, also solves *globally* models with general nonlinear functions.

3.1 Functional constraints and bilinear terms

Nonlinear functions — logarithms, exponentials, powers — enter the model through *functional constraints*: an auxiliary variable y is introduced and $y = f(x)$ is imposed. Ratios such as $1/(\mu - \lambda)$ are instead reformulated with an auxiliary variable and a bilinear constraint ($w\,(\mu - \lambda) = 1$).

```
m.Params.FuncNonlinear = 1      # functions handled exactly (globally)
z = m.addVar(lb=-GRB.INFINITY)
m.addGenConstrLog(g, z)         # z = log(g);  also Exp, Pow, Sin, ...
```

```
m.addQConstr(w * v == 1)        # bilinear terms: NonConvex = 2 is
    required
```

3.2 Tolerances and scaling

Two practical devices, both used in the scripts of these lecture notes:

- **tolerances**: for accurate marginal analyses (differences between two nearby optima) tighten `MIPGap`, `FeasibilityTol` and `OptimalityTol` to 10^{-9};
- **scaling**: reformulate to avoid tiny quantities — for instance $q \le A\,p^{-\varepsilon}$ is written $q \cdot r \le A$ with $r = p^{\varepsilon}$, which keeps the numbers in a healthy range.

3.3 Multipliers in convex QPs

In convex QPs with linear constraints Gurobi returns the multipliers as in LPs: the attribute `Pi` of each constraint is the KKT multiplier of chapter 2, with Gurobi's sign convention (change of the optimal value per unit of right-hand side). Quadratic constraints need `m.Params.QCPDual = 1`, and the attribute is `QCPi`. Unlike in LPs, the multiplier is exact only at the margin: for finite increments the optimal value also changes because of the curvature, so the check by perturbation uses a small ε.

3.4 When the duals are not there

For models with functional or bilinear constraints the solver does not provide the duals: shadow prices and multipliers are estimated *by perturbation* (increase the right-hand side by ε, re-optimize, take the incremental ratio), as for the KKT multipliers of chapter 2.

Why a single solver

Same syntax, same interpretation checklist and — above all — a *certified global* optimum even in non-convex problems: with a local solver every result would have to be accompanied by the question "is it only a local optimum?". Convex problems with a purely quadratic or conic structure (constraints such as $d_x^2 + d_y^2 \le d^2$) require nothing special.

3.5 In summary

In summary

- Nonlinear functions enter through functional constraints $y = f(x)$; ratios through bilinear constraints with `NonConvex = 2`.
- Marginal analyses need tight tolerances; a good reformulation improves the scaling.
- In a convex QP the multipliers are read with `Pi`; without duals, shadow prices are estimated by perturbation.
- A single solver for every model: the same syntax, and a certified global optimum.

Exercise 3.1

Implement the QP of chapter 2, min $x_1^2 + 2x_2^2$ with $x_1 + x_2 \geq 6$, print `Pi` and `Slack` of the constraint and verify the multiplier by re-optimizing with right-hand side 6.01.

Exercise 3.2

Solve with Gurobi's functional constraints the one-variable problem max $5\log(1+x) - x$ with $x \in [0, 10]$ and compare with the analytical solution $(5/(1+x) - 1 = 0 \Rightarrow x = 4)$.

Part II

Nonlinear deterministic models

Chapter 4

Production with quadratic smoothing

Class: convex QP **Script:** `python/lab04_production_quadratic.py`
Online page

Prerequisites

- From the module *LP Modelling*, Chapter 4 «Multi-period production and inventory»: the linear production and inventory model, with its data.
- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex quadratic programs, the KKT conditions and the multipliers as shadow prices.
- From this module, Chapter 3 «The solver for nonlinear models»: quadratic objectives and the multipliers of a QP in `gurobipy`.

Objectives

At the end of this chapter you will be able to:

- add a convex quadratic term to a linear model;
- read the multipliers of a QP and verify them by perturbation;
- trace the cost-smoothness frontier as the weight varies.

4.1 What changes with respect to the linear model

The case is the one of LP Modelling, "Production and inventory": three products, six months, one shared resource, with the same data and the same description. The minimum-cost plan of that model changes pace abruptly from one month to the next: total production goes from 219.5 to 389.0 units, with jumps of up to 81.4 units in a month. Every jump has a cost that the linear model does not see: overtime, shifts to reorganize, suppliers to realign.

Here we add to the objective a *quadratic* penalty on the jumps of total production. A jump twice as large costs four times as much: the model prefers many small adjustments to a few large changes of pace.

4.2 The model

What changes. Data, variables and constraints are those of the linear model: n products indexed by i, m months indexed by t, demand d_{it}, costs c_{it} and h_{it}, unit hours a_i, capacity b_t, initial stock s_{i0}; variables x_{it} (production) and s_{it} (end-of-month stock). A single datum is added:

Symbol	Type	Meaning
γ	$\in \mathbb{Q}_{\geq 0}$	weight of the jumps of total production (€ per unit^2)

Using the same variables, a QP model for the problem is the following:

QP model of production with smoothing

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

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

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

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

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

Description of the objective function and of the constraints of the QP model:

- the quadratic objective function (4.1a) adds to the production and holding cost of the linear model the weight γ times the square of every jump of total production between two consecutive months;
- the linear constraints (4.1b) are the inventory balances ($n\,m$ linear constraints);
- the linear constraints (4.1c) limit the machine hours of every month (m linear constraints);
- the constraints (4.1d) and (4.1e) define the variables of the model.

Why the QP is convex

Every jump $\sum_i x_{it} - \sum_i x_{i,t-1}$ is a linear function of the variables, and the square of a linear function is convex. A sum of convex functions with weights $\gamma \geq 0$ stays convex, and the constraints are linear: the QP is convex, every local minimum is global and the KKT conditions of chapter 2 are necessary and sufficient. With $\gamma = 0$ we recover the linear model.

4.3 Worked example

One product, two months

Demand $d_1 = 60$ and $d_2 = 100$, production cost $c = 10$, holding $h = 1$, zero initial stock, ample capacity. The linear model produces to demand: $x_1 = 60$, $x_2 = 100$, cost 1600, jump 40.

- **Step 1 — solving we obtain**, with $\gamma = 0.5$:

  $$\tilde{x}_1 = 79.75, \quad \tilde{x}_2 = 80.25, \quad \tilde{s}_1 = 19.75, \quad \tilde{z} = 1619.875, \quad \tilde{\pi}_1 = 9.5, \quad \tilde{\pi}_2 = 10.5,$$

  where $\tilde{\pi}_1$ and $\tilde{\pi}_2$ are the multipliers of the two balances.

- **Step 2 — one variable.** From the balances $x_1 = 60 + s_1$ and $x_2 = 100 - s_1$, so the jump is $40 - 2s_1$ and the objective is

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

- **Step 3 — stationarity.** $f'(s_1) = 1 - 4\gamma\,(40 - 2s_1) = 0$ gives $s_1 = 20 - 1/(8\gamma)$: with $\gamma = 0.5$ we find $s_1 = 19.75$, as the solver does. One unit in stock costs $h = 1$ and reduces the jump by 2: stock is built until the marginal saving on the penalty equals the holding cost.

- **Step 4 — the threshold.** $f'(0) = 1 - 160\gamma$: for $\gamma \le 1/160 = 0.00625$ the derivative at $s_1 = 0$ is not negative and the optimum keeps no stock. With $\gamma = 0.004$ the solver indeed gives $\tilde{s}_1 = 0$, $\tilde{z} = 1606.4$, $\tilde{\pi} = (9.68;\ 10.32)$.

- **Step 5 — the multipliers.** With positive stock the KKT conditions give $\tilde{\pi}_2 - \tilde{\pi}_1 = h = 1$: one more unit of demand in month 2 costs one holding period more than one in month 1. With $\gamma = 0.004$ the difference is $0.64 < h$, and indeed keeping stock does not pay.

4.4 Implementation

With respect to the linear script only the objective changes; the full version is in the script named at the top of the chapter:

```
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]      # capacity multipliers
```

4.5 Results

```
gamma = 0.0: objective 32,889.33  linear cost 32,889.33  largest jump 81.4
   total production:  219.5  301.0  341.4  374.3  389.0  324.8
```

```
gamma = 0.5: objective 33,003.10  linear cost 33,000.96  largest jump  1.2
   total production:  323.3  324.3  325.5  326.1  325.9  324.8
gamma = 2.0: objective 33,004.74  linear cost 33,004.14  largest jump  0.4
```

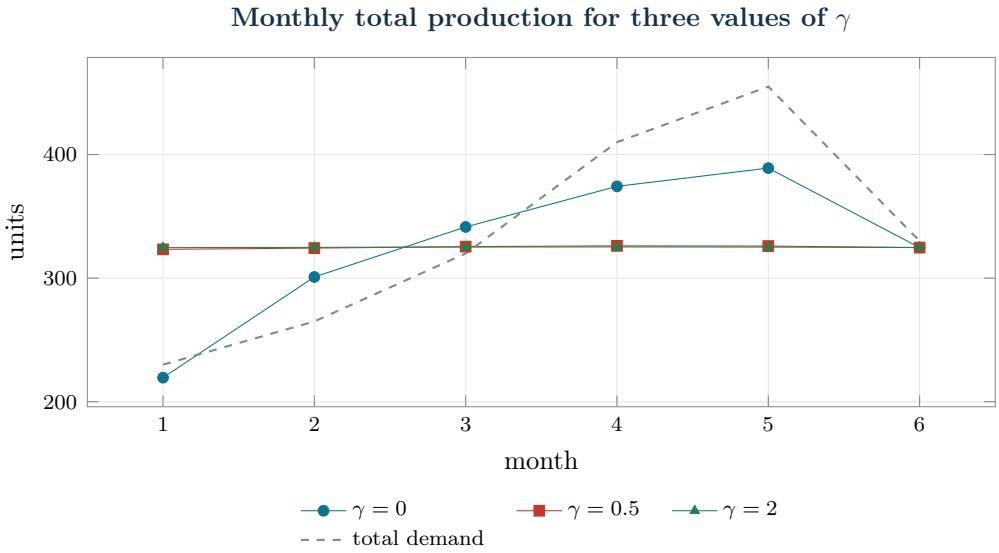


Figure 4.1: Monthly total production: with $\gamma = 0$ (linear model) the profile follows demand; already with $\gamma = 0.5$ it becomes almost flat, around 325 units per month.

The smooth plan costs € 111.63 of production and holding more than the linear plan (+0.34%). With $\gamma = 0.5$ the plan by product is:

```
month          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
stock prod. 1: 153.3  212.6  194.7  110.4  0.0  0.0
stock prod. 3:   0.0    0.0   23.5   23.9  5.2  0.0
```

Managerial insight

In the linear model the stock built in advance was that of product 3, the most expensive in hours. With smoothing it is mostly product 1 that goes to stock, up to 212.6 units: the penalty counts *units*, not hours, and product 1 gives more units per machine hour ($a_1 = 0.9$). A new term in the objective can change the structure of the solution, not only its numbers.

4.6 Sensitivity analysis

Capacity multipliers

```
month 1: Pi(gamma=0) =  0.000   Pi(gamma=0.5) =  0.000 EUR/hour
month 2: Pi(gamma=0) = -0.762   Pi(gamma=0.5) =  0.000 EUR/hour
month 3: Pi(gamma=0) = -1.524   Pi(gamma=0.5) =  0.000 EUR/hour
month 4: Pi(gamma=0) = -2.286   Pi(gamma=0.5) = -0.667 EUR/hour
month 5: Pi(gamma=0) = -3.048   Pi(gamma=0.5) = -1.333 EUR/hour
month 6: Pi(gamma=0) = -3.810   Pi(gamma=0.5) = -3.134 EUR/hour
Check: +1    hour  in month 6 -> difference quotient -3.060
```

```
Check: +0.01 hours in month 6 -> difference quotient -3.133
```

The multiplier of a QP is exact only at the margin

In the linear model one more hour in month 6 lowers the cost by exactly the shadow price, as long as we stay within the validity range. In the QP the optimal value has curvature: with one more hour the change is −3.060, with 0.01 hours the difference quotient is −3.133, that is the multiplier −3.134. With smoothing the capacity of the first three months is no longer active: the levelled plan does not saturate the hours at the start of the horizon.

The cost-smoothness frontier

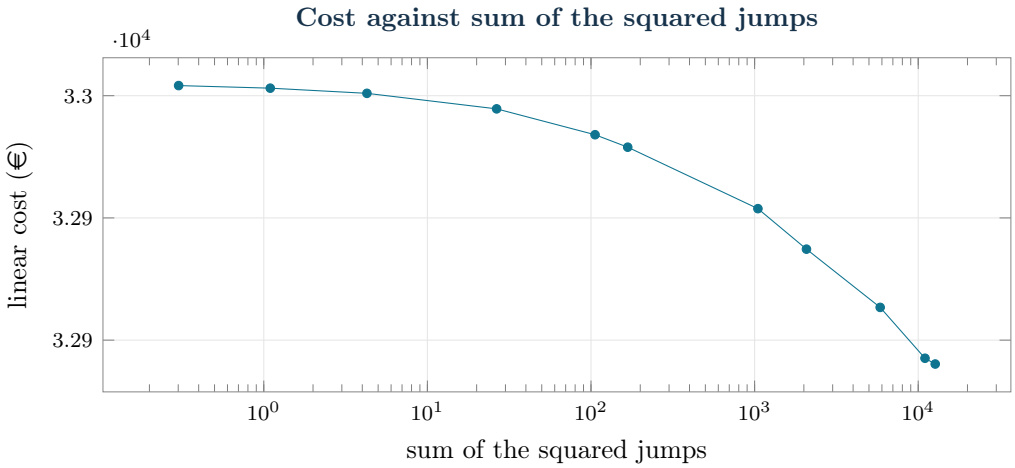


Figure 4.2: Every point is the optimal plan for a value of γ between 0.001 and 2: the smoothest plans on the left, the most expensive at the top.

```
gamma = 0.001:  cost 32,890.23   sum of jumps^2 12,703.7   largest jump 72.5
gamma = 0.010:  cost 32,937.22   sum of jumps^2  2,074.7   largest jump 26.0
gamma = 0.050:  cost 32,978.93   sum of jumps^2    167.7   largest jump  7.4
gamma = 0.500:  cost 33,000.96   sum of jumps^2      4.3   largest jump  1.2
```

Managerial insight

The first steps towards smoothness cost very little: with $\gamma = 0.01$ the sum of the squared jumps falls by more than six times (from 13,699 to 2,075) for € 48. An almost flat plan costs about € 112 in all over a horizon of 33,000: if a change of pace costs the company more than a few euros, levelling pays.

4.7 In summary

In summary

- A convex quadratic term added to an LP gives a convex QP.
- The weight γ traces the cost-smoothness frontier: the first steps cost little.
- QP multipliers are read with `Pi`, but they hold at the margin.

- A new term can change the structure of the solution.

4.8 Exercises

Exercise 4.1 — by hand

In the worked example set $\gamma = 0.05$: compute s_1, x_1, x_2, the optimal value and the multipliers of the balances, then check with the solver.

Exercise 4.2 — check

With $\gamma = 0.5$ verify by perturbation the capacity multiplier of month 5, with increments of 1 and of 0.01 hours, and compare with the check of month 6 in the text.

Exercise 4.3 — smoothing by product

Penalize the jumps of every product, $\gamma \sum_{i=1}^{n} \sum_{t=2}^{m} (x_{it} - x_{i,t-1})^2$, with $\gamma = 0.5$. Compare cost and profiles with the smoothing of total production.

Exercise 4.4 — linear smoothing

Replace the square with the absolute value, $\gamma \sum_{t=2}^{m} |v_t - v_{t-1}|$ with $v_t = \sum_i x_{it}$, written as an LP with two non-negative variables per jump. With $\gamma = 5$, what shape does the profile take? Why?

Exercise 4.5 — target level

Penalize the deviation from a fixed level, $\gamma \sum_{t=1}^{m} (v_t - 325)^2$, with $\gamma = 0.5$. Compare cost and profile with the smoothing of the jumps.

Chapter 5

Supply chain with congestion and sustainability

Class: LP / convex NLP

Script: `python/lab05_supplychain.py`
Online page

Prerequisites

- From the module *LP Modelling*, Chapter 2 «Background: linear programming and sensitivity analysis»: shadow prices, reduced costs and their ranges.
- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex quadratic programs, the KKT conditions and the multipliers as shadow prices.

Objectives

At the end of this chapter you will be able to:

- formulate a minimum-cost flow problem with conservation constraints;
- model congestion with convex quadratic terms;
- draw a cost-emissions frontier with an internal CO_2 price;
- understand why the solutions of LPs jump from one vertex to another.

5.1 Managerial motivation

Products cross a network of plants, hubs and markets. Every leg has a cost, a capacity and a CO_2 footprint; the effective cost grows rapidly when a leg approaches saturation. Three managerial questions: how should the flows be routed at minimum cost? what changes if we penalize congestion? how much must CO_2 be worth for the "clean" routes to become convenient?

This is the everyday problem of whoever manages distribution and logistics: choosing the routes, deciding how much to load on each leg, evaluating investments in capacity and, more and more often, accounting for the carbon footprint of the shipments. The mathematical structure — the network flow — is among the most reusable ones in the whole of operations research.

5.2 Guided construction of the model

The problem in words. *We decide* how many units travel on each leg of the network. *The objective* is to minimize the total transport cost (in the variants: plus congestion or CO_2). *The constraints* impose that the flows add up at every node — the plants do not ship more than they produce, the hubs retain nothing, the markets receive exactly their demand — and that no leg exceeds its own capacity.

Data (input of the model)

Symbol	Type	Meaning
$G = (N, A)$	directed graph	nodes N and arcs $A \subseteq N \times N$; we write $\|N\|$ and $\|A\|$ for the number of nodes and of arcs
b_i	$\in \mathbb{Q}$	balance of node $i \in N$: supply if > 0, demand if < 0, transit if $= 0$
u_{ij}	$\in \mathbb{Q}_{>0}$	capacity of arc $(i, j) \in A$ (units)
c_{ij}	$\in \mathbb{Q}_{\geq 0}$	transport cost of one unit on arc (i, j) (€)
e_{ij}	$\in \mathbb{Q}_{\geq 0}$	emissions per unit transported on (i, j) (kg CO_2)

Decision variables

We introduce one non-negative variable for each of the $|A|$ arcs:

$$x_{ij} = \text{units of product sent on arc } (i, j), \qquad \forall (i, j) \in A.$$

Using these variables, an LP model for the problem is the following:

Minimum-cost flow LP model

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

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

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

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

Description of the objective function and of the constraints of the LP model:

- the linear objective function (5.1a) minimizes the total transport cost: every unit that crosses arc (i, j) pays the unit cost c_{ij};
- the linear constraints (5.1b) are the flow conservations: at every node, what goes out minus what comes in equals the balance b_i — positive at the plants, negative at the markets, zero at the hubs ($|N|$ linear constraints); the sum of the balances over all the nodes must be ≥ 0, otherwise the model is infeasible;
- the linear constraints (5.1c) ensure that no leg carries more than its own capacity ($|A|$ linear constraints);
- the constraints (5.1d) define the variables of the model.

Variants on the objective (the constraints do not change):

$$\underbrace{\textstyle\sum_{(i,j)\in A}\left(c_{ij}\,x_{ij}+\alpha\,c_{ij}\,x_{ij}^2/u_{ij}\right)}_{\text{congestion (convex)}} \qquad \underbrace{\textstyle\sum_{(i,j)\in A}(c_{ij}+\tau\,e_{ij})\,x_{ij}}_{\text{internal } CO_2 \text{ price}}$$

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

where $\alpha \in \mathbb{Q}_{\ge 0}$ weighs congestion and $\tau \in \mathbb{Q}_{\ge 0}$ (€/kg) is the internal price of CO_2.

Line-by-line explanation

Congestion. The term $\alpha\,c_{ij}\,x_{ij}^2/u_{ij}$ is convex and grows with the utilization of the arc: it represents overtime, queues at the terminals, insurance premiums. Its effect is to *spread* the flows over several routes instead of saturating the cheapest one.
The price of CO_2. No emissions constraint is needed: it is enough to add them to the cost with the internal price τ. By varying τ one draws the whole cost-emissions frontier ("weight" approach; the "constraint" alternative imposes $\sum_{(i,j)\in A} e_{ij}\,x_{ij} \le \bar{e}$ and reads the dual).

5.3 Worked example by hand

Two competing routes

A market asks for 100 units that can be served by two routes: route 1 with $c_1 = 2$ € and $u_1 = 80$; route 2 with $c_2 = 5$ € and abundant capacity.
Pure LP. The cheap route is filled and the rest overflows onto the other one: $x_1 = 80$, $x_2 = 20$, cost $2 \cdot 80 + 5 \cdot 20 = 260$ €. Shadow price of demand: serving the 101st unit would cost 5 € (route 1 is saturated); shadow price of capacity u_1: +1 unit of capacity saves $5 - 2 = 3$ €.
With congestion ($\alpha = 1$): cost of route $1 = 2x_1 + 2x_1^2/80$. The marginal cost of route 1, $2 + 4x_1/80$, has to be equalized with that of route 2 (otherwise it is worth shifting flow): $2 + x_1/20 = 5 \Rightarrow x_1 = 60$, $x_2 = 40$. The cheap route *is no longer saturated*: the model leaves a safety margin, exactly as a prudent planner would do.

5.4 Case study

Network with 2 plants (S1, S2), 2 hubs (H1, H2) and 4 markets (M1–M4); 12 arcs with capacities, costs and emissions that are *not* proportional to the costs: the cheap legs are "by road" (polluting), the expensive ones "by rail" (clean). Data in `data/supplychain_archi.csv`; supply $(260, 240)$, total demand 450.

```
m = gp.Model("supply_chain")
x = m.addVars(A, name="x", ub=U)    # ub directly on the arc
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:    # convex quadratic term
    obj += gp.quicksum(congestione * c[a] * x[a] * x[a] / U[a]
```

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

5.5 Results

```
LP minimum cost : cost 3.385.00 EUR  emissions 2.730 kg  max utilization 100%
  saturated arcs: S1->H1, S2->H2, H2->M3
  shadow prices of demand: M1 8.00  M2 9.50  M3 10.00  M4 10.50 EUR/unit
  reduced costs of arcs at zero (threshold = SAObjLow):
    S2->H1 +2.00 (threshold 5.00)   H1->M4 +0.50 (threshold 5.50)
    H2->M1 +4.50 (threshold 1.50)   H2->M2 +1.00 (threshold 3.00)  EUR/unit
Congestion a=1  : cost 3.702.81 EUR  emissions 2.530 kg  max utilization  90%
Minimax         : minimum possible maximum utilization 58.4%
```

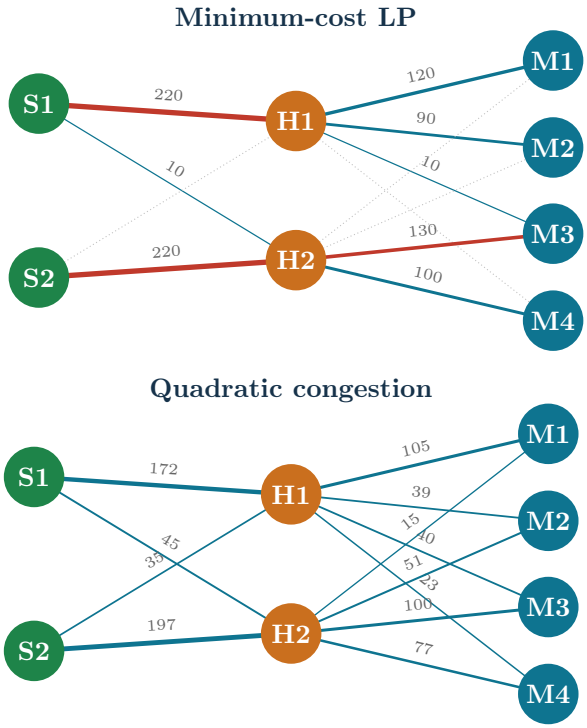


Figure 5.1: The optimal flows on the network (plants S in green, hubs H in orange, markets M in teal; the thickness of the arcs is proportional to the flow, red indicates a saturated arc, the number is the flow in units). Above: minimum-cost LP, which saturates three arcs. Below: model with quadratic congestion, which spreads the flows over several routes and saturates no arc.

> **Managerial insight**
>
> The shadow prices of demand (8–10.50 €/unit) are the *marginal costs of service* of the four markets: this is the number to compare with the commercial margin before accepting new orders, and the correct basis for an internal transfer price. M4 is the most expensive market to serve: +31% with respect to M1.

The reduced costs of the unused arcs, with their range

Four arcs stay at zero and their reduced cost (attribute `x[a].RC`) is their *convenience threshold*: route S2 → H1 (cost 7 €/unit, `RC` = +2) would enter the optimal solution only if its cost dropped below 7 − 2 = 5 €/unit — the number to bring to the table in a negotiation with the carrier. As always, next to the reduced cost we quote its range of validity `SAObjLow`–`SAObjUp`: for a variable at zero it is [threshold, +∞), that is, the cost of the arc may grow as much as it likes (it stays unused) and the threshold itself is `SAObjLow`. Route H1 → M4 is one step away from being used (`RC` = +0.50): a discount of 50 cents is enough. The positive variables all have zero reduced cost.

5.6 Sensitivity analysis: the price of CO_2

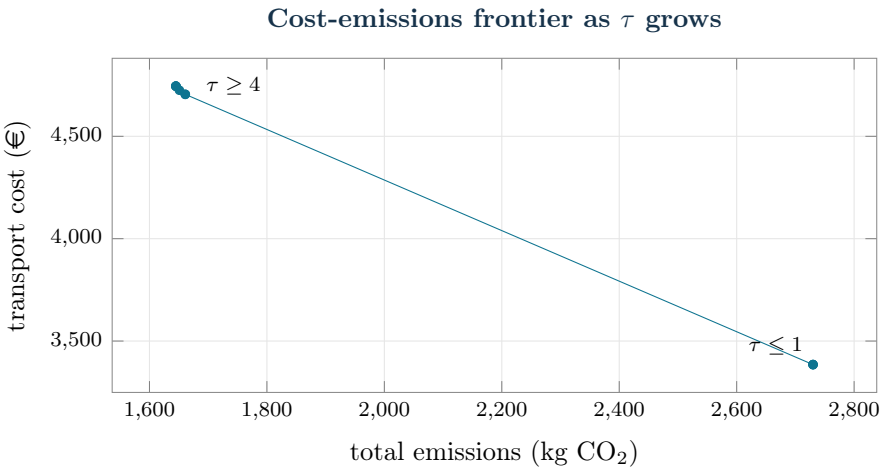


Figure 5.2: Cost-emissions frontier: every point is the optimal solution for an internal price of CO_2 (τ); at the bottom right the "by road" configuration (cheap, 2730 kg), at the top left the "by rail" configuration (expensive, 1645 kg). The frontier proceeds by jumps between vertices of the LP.

```
tau = 0.0 ... 1.0 : cost 3385  emissions 2730   (routes by road)
tau = 1.24        : cost 4025  emissions 2173   (intermediate vertex)
tau = 1.26        : cost 4265  emissions 1981
tau = 1.5         : cost 4705  emissions 1661   ("rail" configuration)
tau = 4.0 and up  : cost 4745  emissions 1645   (stable)
```

Why the frontier proceeds by jumps

The solutions of an LP lie at the *vertices* of the feasible polyhedron: as τ grows the optimum stays put on one vertex, then it *jumps* to the next one. A bisection on the model shows that up to $\tau = 1$ nothing changes, that between $\tau = 1.0$ and $\tau = 1.5$ the network crosses intermediate vertices (at $\tau = 1.24$: 2173 kg) and that the main jump happens exactly at $\tau = 1.25$ €/kg. Managerially: an internal price of CO_2 below the threshold does not change the routes; just above it, it restructures the whole network at once. The model with congestion, being strictly convex, would instead move continuously.

The "pure" minimax ($\min z$) is degenerate: any routing with utilization $\leq z^*$ is optimal, even an extremely expensive one. In practice one always uses a combination (cost $+\rho \cdot z$) or one constrains z and minimizes the cost.

5.7 In summary

In summary

- The shadow prices of demand are the marginal costs of serving the markets; the reduced costs of unused arcs are convenience thresholds.
- Congestion modelled with quadratic terms lowers the maximum utilization at a higher cost.
- Varying the CO_2 price τ traces the cost-emissions frontier: in an LP the optimum jumps between vertices, a strictly convex model moves continuously.
- Pure minimax is degenerate: it must be combined with the cost.

5.8 Exercises

Exercise 5.1 — check

Verify by perturbation the shadow price of the demand of M4 (raise it to 101) and that of the capacity of arc H2→M3.

Exercise 5.2 — failure of a hub

Reduce by 50% the capacities of the arcs entering and leaving H2: how much does the failure cost? which markets suffer the most? (Compare the shadow prices before/after.)

Exercise 5.3 — the rail threshold

Reproduce with a bisection on τ the threshold $\tau = 1.25$ €/kg found in the text, and list which arcs change flow when crossing it. Compare the threshold with the current ETS price of CO_2.

Exercise 5.4 — emissions constraint

Reformulate with the constraint $\sum_{(i,j)\in A} e_{ij}\, x_{ij} \leq \bar{e}$ for $\bar{e} = 2000$ kg and verify that the dual of the constraint coincides with the "weight" τ that produces the same emissions.

Exercise 5.5 — barrier

Replace the congestion term with the barrier $\alpha\, x_{ij}/(u_{ij} - x_{ij})$ and compare: which formulation keeps the flows further away from saturation?

Chapter 6

The Markowitz portfolio

Class: convex QP

Script: `python/lab06_markowitz.py`
Online page

Prerequisites

- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex quadratic programs, the KKT conditions and the multipliers as shadow prices.
- From the module *Convex Optimization*, Chapter 4 «Quadratic functions»: quadratic forms and covariance matrices as positive semidefinite matrices.

Objectives

At the end of this chapter you will be able to:

- formulate and solve a convex QP;
- build and read the efficient frontier;
- understand the effect of mandate constraints;
- be wary of the estimates of the expected returns.

6.1 Managerial motivation

How much should we invest in each asset in order to balance expected return and risk? The Markowitz model is the most famous QP in history: it minimizes the variance of the portfolio subject to a minimum return. The result is not a single number but an *efficient frontier*: the complete menu of risk-return trade-offs among which the decision maker chooses according to their own degree of risk aversion.

The problem was born in finance (Markowitz, 1952, Nobel prize) but its structure — allocating a budget among risky and correlated alternatives — reappears everywhere: mixes of R&D projects, customer portfolios, even the allocation of production capacity across volatile markets. It is also the natural bridge between optimization and statistics: the data of the model are *estimates*, with all the consequences that this entails.

6.2 Guided construction of the model

The problem in words. *We decide* the share of capital to invest in each asset. *The objective* is to minimize the risk of the portfolio, measured by the variance of its return. *The constraints*: invest the whole capital (the shares sum to 1), guarantee a minimum expected return and respect the minimum and maximum share of each asset.

Data (input of the model)

Symbol	Type	Meaning
n	$\in \mathbb{Z}_{\geq 1}$	number of assets; the assets are indexed by $i \in \{1, 2, \ldots, n\}$
μ_i	$\in \mathbb{Q}$	expected return of asset i (per year)
q_{ij}	$\in \mathbb{Q}$	covariance between the returns of assets i and j; the matrix $\boldsymbol{Q} = (q_{ij})$ is positive semidefinite ($\boldsymbol{Q} \succeq 0$)
$\bar{r}$	$\in \mathbb{Q}$	minimum required expected return
ℓ_i, u_i	$\in \mathbb{Q}$, $0 \leq \ell_i \leq u_i \leq 1$	minimum and maximum share that can be invested in asset i

Decision variables

We introduce the following n non-negative variables:

$$x_i = \text{share of capital invested in asset } i, \qquad \forall i \in \{1, 2, \ldots, n\}.$$

Using these variables, a QP model for the problem is the following:

Minimum-variance QP model with a minimum return

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

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

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

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

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

Description of the objective function and of the constraints of the QP model:

- the quadratic objective function (6.1a) minimizes the variance of the return of the portfolio; it contains the covariances q_{ij}, and this is where diversification comes from: combining weakly correlated assets lowers the overall risk below that of the individual components;
- the linear constraint (6.1b) imposes that the expected return of the portfolio reaches the threshold $\bar{r}$: it is the "dial" with which the efficient frontier is traced out, and its multiplier tells us how much variance each additional point of return costs (one linear constraint);

- the linear constraint (6.1c) imposes that the whole capital is invested (one linear constraint);
- the linear constraints (6.1d) are the caps per asset, typical regulatory or mandate constraints (n linear constraints);
- the constraints (6.1e) define the variables of the model (with $\ell_i = 0$: short selling is forbidden).

Equivalent formulations:

$$\max \sum_{i=1}^{n} \mu_i x_i - \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} x_i x_j \qquad \text{(mean-variance)},$$

and $\max \sum_{i=1}^{n} \mu_i x_i$ with variance $\leq \bar{\sigma}^2$ (maximum return with bounded risk). By varying $\bar{r}$ (or λ, or $\bar{\sigma}$) the same frontier is traced out.

Line-by-line explanation

Convexity. Every covariance matrix is positive semidefinite, hence the QP is convex and the optimum that is found is a *certified global* one.

6.3 Worked example by hand

Two uncorrelated assets

Volatilities $\sigma_1 = 20\%$, $\sigma_2 = 30\%$, zero correlation. Minimize the variance with $x_1 + x_2 = 1$ (without the return constraint).

- **Step 1.** Substituting $x_2 = 1 - x_1$: $f(x_1) = 0.04\, x_1^2 + 0.09\,(1 - x_1)^2$.
- **Step 2.** $f'(x_1) = 0.08\, x_1 - 0.18\,(1 - x_1) = 0 \;\Rightarrow\; x_1 = \dfrac{0.09}{0.04 + 0.09} = \dfrac{9}{13} \approx 69.2\%$.
- **Step 3.** Variance of the portfolio: $f(9/13) = 0.04 \cdot 0.479 + 0.09 \cdot 0.095 = 0.0277$, that is a volatility of 16.6%: **less risky than either of the two assets**. This is the diversification effect in its purest form: the optimal weights are inversely proportional to the variances.

6.4 Case study

Eight sector ETFs (*Exchange Traded Fund*: listed funds that replicate a basket of securities of a given sector), 60 monthly returns simulated with a single-market-factor model (file `data/markowitz_rendimenti.csv`); $\boldsymbol{\mu}$ and $\boldsymbol{Q}$ are *estimated* from these data, as in practice. Annualized statistics:

```
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")           # shares, 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 Results

```
Global minimum variance : return 11.06%, volatility  6.03%
Equally weighted (1/n)  : return 10.74%, volatility 10.13%
Min-variance composition: ENE 5.2%  IND 1.9%  SAN 11.5%  CON 22.3%  UTL 58.3%
```

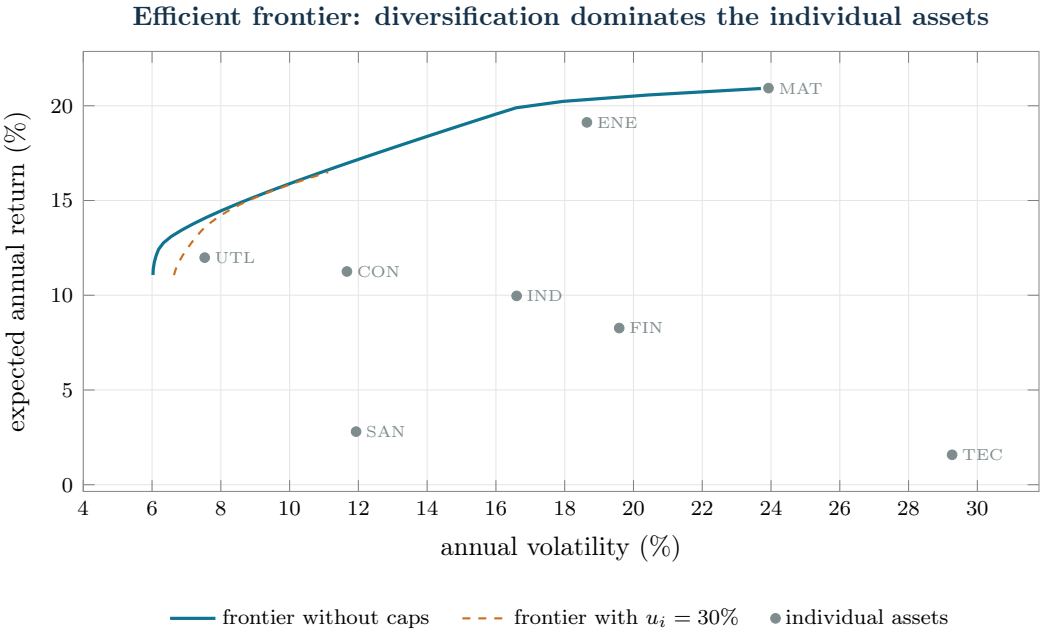


Figure 6.1: Risk-return plane: the eight assets (grey dots), the efficient frontier without caps (teal line), the frontier with a 30% cap per asset (dashed orange), the minimum-variance portfolio (red star) and the equally weighted $1/n$ portfolio (purple diamond), which turns out to be dominated.

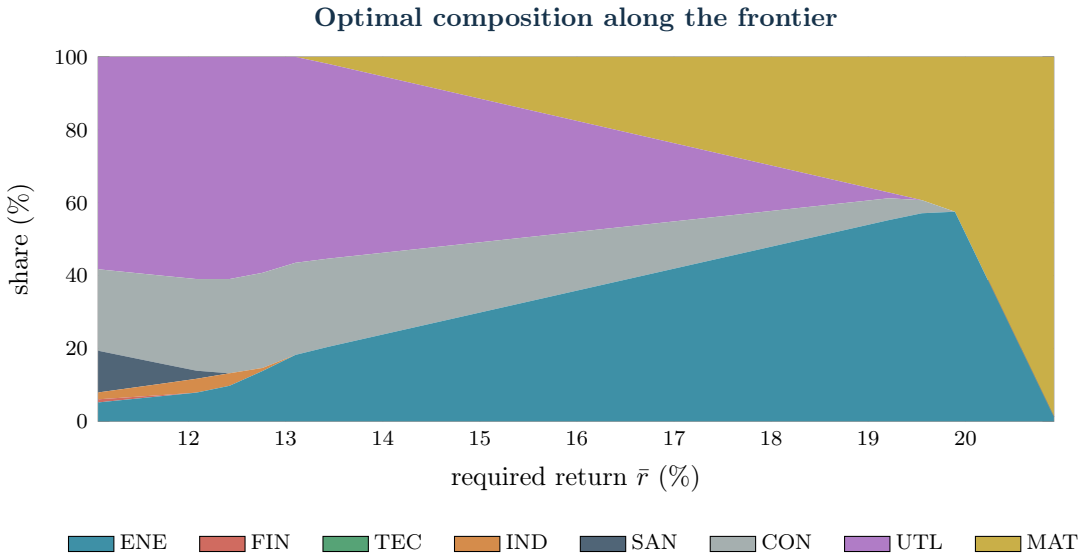


Figure 6.2: Composition of the optimal portfolio (stacked areas, in %) as the required return $\bar{r}$ grows: on the left the defensive mix dominates (UTL, CON, SAN), towards the right the portfolio concentrates progressively on the high-return assets (MAT, ENE) up to the single-asset portfolio at the extreme.

Managerial insight

Three messages from the figure. (1) The minimum-variance portfolio (volatility 6%) is much less risky than the best individual asset (UTL, 7.5%) while still returning 11%. (2) The equally weighted $1/n$ portfolio — the "I know nothing" strategy — is clearly dominated: the same region of return but almost twice the volatility. (3) The cap $u_i = 30\%$ cuts off the upper part of the frontier: the cost of the mandate constraints can be *seen* as the distance between the two curves.

6.6 Sensitivity analysis

```
r_min =  6%: vol 6.03%   (constraint NOT active: same as min variance)
r_min =  8%: vol 6.03%   (same)
r_min = 10%: vol 6.03%   (same)
r_min = 12%: vol 6.10%   d(variance)/d(r_min) ~ 0.0221
```

A constraint that costs nothing

Up to $\bar{r} = 11\%$ the return constraint is *inactive*: the minimum-variance portfolio already returns 11.06%, hence asking for "at least 8%" costs nothing and the multiplier is zero. Only beyond 11.06% does the constraint bite, and every additional point of return is paid for in variance (multiplier ≈ 0.022 at $\bar{r} = 12\%$). Recognizing inactive constraints avoids "paying" for constraints that are free.

In the simulated data the asset TEC has a true α of 11% per year, but over 60 months the *estimated* return is 1.6%: the noise dominates. This is the most important practical lesson of Markowitz: **the estimates of the expected returns are far more unstable**

than those of the covariances, and optimized portfolios chase the estimation errors. Remedies used in practice: the constraints u_i, shrinkage of the estimates, or optimizing risk only (minimum variance).

6.7 In summary

In summary

- Markowitz minimizes the variance of the portfolio subject to a minimum expected return: a convex QP.
- Along the efficient frontier one moves from the defensive mix to the high-return assets; mandate constraints have a visible cost.
- Below the minimum-variance return the return constraint is inactive and its multiplier is zero; beyond it every point of return costs variance.
- The estimates of expected returns are fragile: caps, shrinkage and minimum variance are the remedies.

6.8 Exercises

Exercise 6.1 — check

Verify by hand the example with two assets and $\rho = 0.5$: the formula becomes $x_1 = (\sigma_2^2 - \rho\sigma_1\sigma_2)/(\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2)$. Is diversification still worthwhile?

Exercise 6.2 — frontier

Recompute the frontier with $u_i = 20\%$: how much maximum return is lost? Draw the three frontiers ($u = 1,\ 0.3,\ 0.2$) in the same plot.

Exercise 6.3 — stability of the estimates

Run the script again with a different random seed (`default_rng(1)`) and compare the optimal compositions for the same $\bar{r}$: how much do they change? Which operational implication do you draw from this?

Exercise 6.4 — tracking error

The benchmark is the equally weighted portfolio $x_i^B = 1/8$. Minimize the tracking error $\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}(x_i - x_i^B)(x_j - x_j^B)$ with expected return $\geq 12\%$: how far must one depart from the benchmark?

Exercise 6.5 — transaction costs

Starting from the equally weighted portfolio, add the quadratic cost $\kappa \sum_{i=1}^{n}(x_i - 1/8)^2$ with $\kappa = 0.5$: how does the composition change with respect to the "clean" optimum?

Chapter 7

Pricing and revenue management

Class: concave / non-convex NLP

Script: `python/lab07_pricing.py`
Online page

Prerequisites

- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex quadratic programs, the KKT conditions and the multipliers as shadow prices.
- From this module, Chapter 3 «The solver for nonlinear models»: bilinear constraints and `NonConvex = 2`; shadow prices estimated by perturbation.

Objectives

At the end of this chapter you will be able to:

- model demand as a function of the price and understand when profit is concave;
- handle a non-convex bilinear objective;
- evaluate capacity correctly: an additional seat is not worth the full margin;
- recognize corner optima, where no derivative vanishes.

7.1 Managerial motivation

Which price maximizes the profit of a concert, of a hotel, of a subscription, when demand decreases as the price grows? Here the price is not a datum but a *variable*: demand becomes endogenous and the profit $p \cdot q$ introduces a bilinear term. This is the chapter in which the laboratory meets non-convexity for the first time — and learns how to handle it.

Setting the price is perhaps the business decision with the highest impact on the margin: one percentage point of price is often worth more than many points of volume. Revenue management — born in the airlines and today standard in hotels, events and rentals — adds the capacity constraint: the seats left unsold tonight cannot be recovered tomorrow.

7.2 Guided construction of the model

The problem in words. *We decide* the selling price and the quantity to sell. *The objective* is to maximize the profit, that is revenue minus variable costs. *The constraints*: we cannot sell more than the market asks for at that price, nor more than the available capacity.

Data (input of the model)

Symbol	Type	Meaning
$d(\cdot)$	function $\mathbb{Q}_{\geq 0} \to \mathbb{Q}_{\geq 0}$	expected demand at price p, decreasing in p
c	$\in \mathbb{Q}_{\geq 0}$	unit marginal cost (€)
k	$\in \mathbb{Q}_{>0}$	available capacity (seats, rooms, units)

Decision variables

We introduce the following 2 non-negative variables:

$$\begin{cases} p = \text{selling price decided by the firm (€)} \\ q = \text{quantity sold (units)} \end{cases}$$

Using these variables, a pricing model for the problem is the following:

Pricing model with a capacity constraint

$$\begin{aligned} \max \quad & p\,q - c\,q && \text{(7.1a)} \\ \text{subject to} \quad & q \leq d(p), && \text{(7.1b)} \\ & q \leq k, && \text{(7.1c)} \\ & p \geq 0, && \text{(7.1d)} \\ & q \geq 0. && \text{(7.1e)} \end{aligned}$$

Description of the objective function and of the constraints of the model:

- the objective function (7.1a) maximizes the profit $p\,q - c\,q = (p - c)\,q$, revenue minus variable cost; it contains the product of the two variables p and q (a *bilinear* term, non-convex), and this is where all the difficulty of the chapter comes from;
- constraint (7.1b) imposes that we do not sell more than the market asks for at price p (one constraint, linear if d is linear);
- the linear constraint (7.1c) imposes that we do not sell more than the available capacity; when it is active, it is this constraint that determines the optimal price (one linear constraint);
- constraints (7.1d) and (7.1e) define the variables of the model.

Demand functions used in the case study, with their respective data (all positive rationals): linear $d(p) = a - b\,p$; constant elasticity $d(p) = \theta\,p^{-\varepsilon}$ with $\varepsilon > 1$; logistic $d(p) = m/\left(1+e^{-\alpha+\beta p}\right)$.

Line-by-line explanation

Why $q \le d(p)$ and not $q = d(p)$? At the optimum the inequality (7.1b) is active by itself (selling less than what is possible at that price is never worthwhile if $p > c$), but writing it as $\le$ keeps the feasible region simpler and the model feasible even when $d(p) > k$.
Where is the non-convexity? In the product $p \cdot q$ of the objective. With linear demand, substituting $q = a - b\,p$, the profit $(p-c)(a-b\,p)$ is a concave parabola: the *reduced* problem is easy. Gurobi nevertheless solves the bilinear version to global optimality (`NonConvex=2`); for general non-linear demands (constant elasticity, logistic) the function constraints of Gurobi are enough (`addGenConstrPow`, `addGenConstrExp`): same solver, still a global optimum.

7.3 Worked example by hand

Linear demand, with and without capacity

Concert: $d(p) = 1200 - 5p$ (hence $a = 1200$, $b = 5$), marginal cost $c = 20$ €, theatre with $k = 400$ seats.

- **Step 1 — without capacity.** Profit $\Pi(p) = (p - 20)(1200 - 5p)$; $\Pi'(p) = 1200 - 10p + 100 = 0$ gives $p^\circ = \dfrac{a/b + c}{2} = \dfrac{240 + 20}{2} = 130$ €, with $q^\circ = 1200 - 650 = 550$ tickets.
- **Step 2 — capacity bites.** $q^\circ = 550 > 400$: the constraint is active, hence exactly k units are sold and the price that empties the theatre is $p^* = (a - k)/b = (1200 - 400)/5 = 160$ €. Profit: $(160 - 20) \cdot 400 = 56.000$ €.
- **Step 3 — how much is an additional seat worth?** As k grows, the optimal price falls by $1/b$: $\dfrac{d\Pi}{dk} = \underbrace{(p^* - c)}_{140} + k\,\underbrace{\dfrac{dp^*}{dk}}_{-1/5} = 140 - 80 = 60$ €. An additional seat is *not* worth the full margin (140 €): in order to fill it the price must be lowered for everybody. A classic mistake in the evaluation of capacity investments.

7.4 Case study and implementation

The same data as in the example (this is the rare case in which the realistic instance is already small), plus the constant-elasticity variant ($\theta = 6 \cdot 10^6$, $\varepsilon = 2.2$), the logistic one ($m = 900$, $\alpha = 6$, $\beta = 0.045$) and the multi-product one.

```
m = gp.Model("pricing_lineare")
m.Params.NonConvex = 2              # bilinear objective 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()
# no duals in non-convex QPs: sensitivity by perturbation
v_cap.RHS = K + 1; m.optimize()
```

```
Gurobi:  p* = 160.00 EUR, q* = 400, profit = 56.000.00 EUR
Marginal value of a seat: 59.80 EUR (theory: 60.00)
Constant elasticity: p* = 79.11 EUR  (optimum at the point where d(p) = k)
Logistic           : p* = 138.29 EUR, profit = 47.316.83 EUR
```

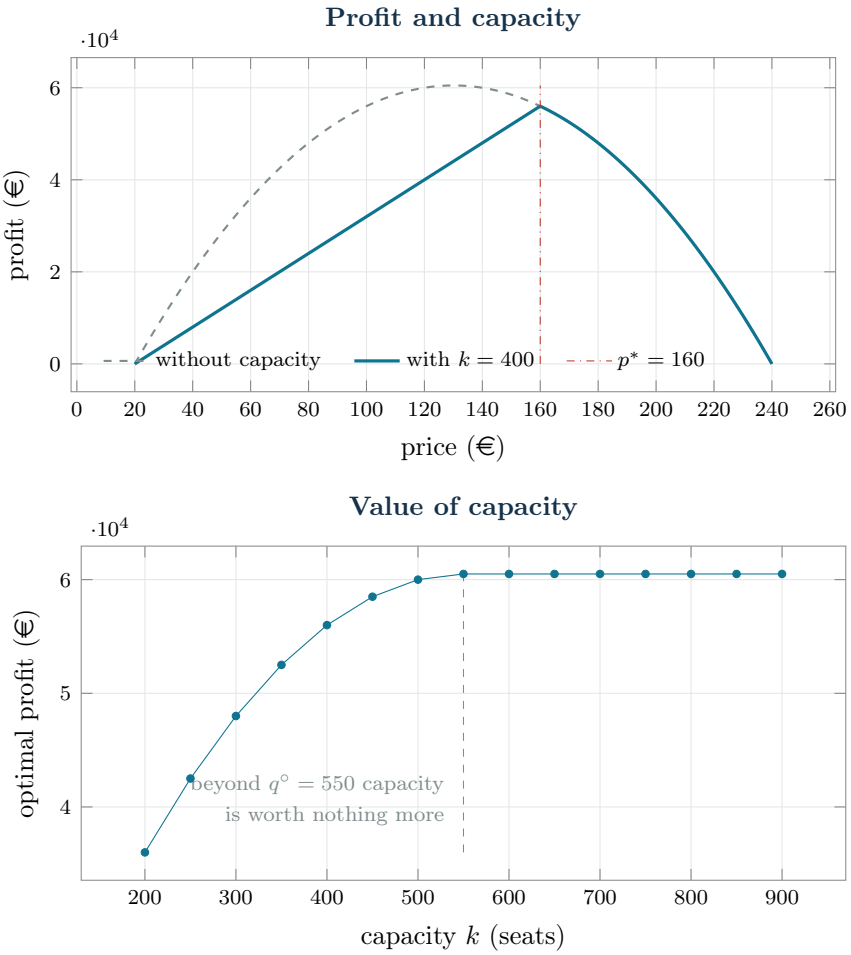


Figure 7.1: On the left: profit as a function of the price for linear demand, without the constraint (dashed grey parabola) and with capacity $k = 400$ (teal line): the constraint cuts the parabola and moves the optimum to $p^* = 160$ €. On the right: optimal profit as capacity grows; beyond $q^\circ = 550$ seats the curve becomes flat and additional capacity is worth nothing more.

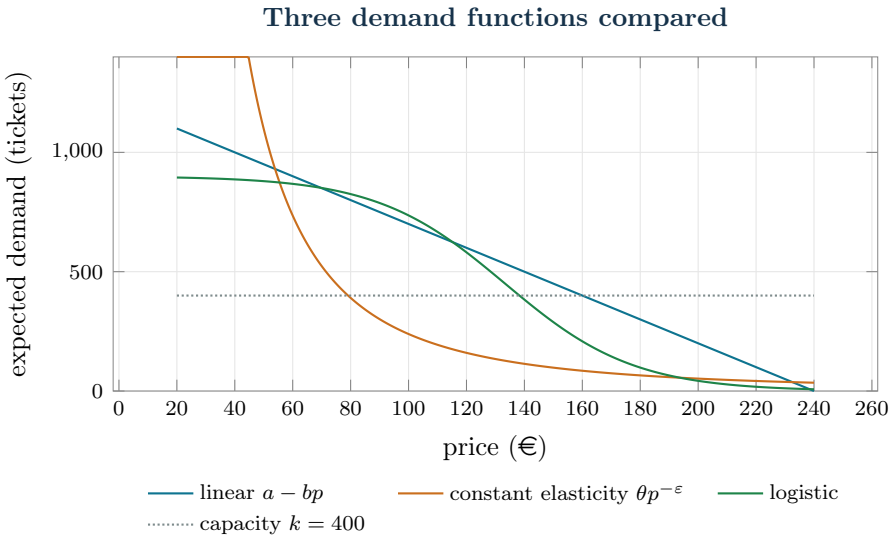


Figure 7.2: The three demand functions of the case study compared (expected tickets as a function of the price) with capacity $k = 400$ (dotted line): the linear one cuts the axis, the constant-elasticity one has a long tail, the logistic one describes the gradual transition between high and low demand.

The optimum at a corner point

With constant elasticity the *unconstrained* optimum would be $p = c\varepsilon/(\varepsilon - 1) = 36.67$ €, but at that price demand would exceed capacity by far. For p below 79.11 € we sell k anyway (so it is worthwhile to raise the price); above it, the profit $(p - c)\,\theta p^{-\varepsilon}$ decreases. The optimum $p^* = 79.11 = (\theta/k)^{1/\varepsilon}$ lies at the *corner point* where $d(p) = k$: it makes no derivative vanish. The moral: never look for the optimum among stationary points only when there are constraints.

7.5 Multi-product version

Two ticket categories with substitution (raising the price of the stalls pushes part of the demand towards the balcony): $d_1(p_1, p_2) = 500 - 2p_1 + 0.6\,p_2$, $d_2(p_1, p_2) = 900 + 0.8\,p_1 - 4p_2$, costs $(30, 15)$ €, capacities $(150, 300)$ seats.

```
stalls   : p1* = 234.04 EUR   q1* = 150/150 (full)
balcony  : p2* = 196.81 EUR   q2* = 300/300 (full)
total profit: 85.148.94 EUR
```

Managerial insight

Both categories fill up, but the prices are not "the ones that empty" each hall taken on its own: the model exploits substitution, keeping the stalls expensive in order to push demand towards the balcony. With substitute products prices must be decided *jointly*: optimizing them one at a time leaves money on the table.

With general demand functions the profit may fail to be concave: a local solver may stop at a local optimum. Practice of the laboratory: study the domain and the derivatives of the

function *before* trusting the solver, try several starting points, and always plot the profit over the relevant interval.

7.6 In summary

In summary

- With linear demand profit is concave; when capacity is active the price rises to fill it exactly.
- An additional seat is worth $p^* - c - k/b$, not the full margin: capacity has value only while it is active.
- Gurobi solves the bilinear model to global optimality with `NonConvex = 2`; without duals, the value of a seat is estimated by perturbation.
- With substitute products prices must be decided jointly; with non-concave profit: study the function, several starting points, a plot.

7.7 Exercises

Exercise 7.1 — check

Verify with Gurobi the marginal value of a seat for $k = 300$ (from the table: 100 €) and obtain it also with the formula $p^* - c - k/b$.

Exercise 7.2 — marginal cost

How do p^*, q^* and the value of a seat change if the marginal cost rises to $c = 60$ €? Redo the three steps of the example by hand.

Exercise 7.3 — reference price

Add the reputational penalty $-\gamma\,(p - 120)^2$ with $\gamma = 2$: which price balances profit and reputation? Does the problem remain concave?

Exercise 7.4 — elasticity

For the constant-elasticity demand, plot p^* as a function of $\varepsilon \in [1.3; 3]$ with $k = 400$: for which ε does the capacity constraint remain active?

Exercise 7.5 — multi-product

In the two-category model, by how much does the profit increase with 50 additional stalls seats? And with 50 additional balcony seats? Which expansion would you recommend, at equal cost?

Chapter 8

Advertising budget allocation

Class: convex NLP **Script:** `python/lab08_budget.py`
Online page

Prerequisites

- From the module *Convex Optimization*, Chapter 3 «Convex and concave functions»: concave functions and diminishing returns.
- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex quadratic programs, the KKT conditions and the multipliers as shadow prices.
- From this module, Chapter 3 «The solver for nonlinear models»: functional constraints for nonlinear functions in `gurobipy`.

Objectives

At the end of this chapter you will be able to:

- model diminishing returns with concave functions;
- solve convex NLPs with the function constraints of Gurobi;
- verify the KKT conditions numerically;
- build the budget value curve and use λ to negotiate the budget.

8.1 Managerial motivation

How should a campaign of 100.000 € be split among social, search, TV and influencers, when every channel shows diminishing marginal returns? The first euros spent on a channel yield a lot, the following ones less and less: the response is a concave function of the spend, and the optimum follows an elegant economic rule that this chapter proves and verifies numerically.

The allocation of a budget among uses with diminishing returns is a universal scheme: marketing over several channels, funds among projects, time among activities, fertilizer among fields. Theory says something strong and verifiable: at the optimum the marginal returns are equal to each other — and this is exactly what the solver will return, digit by digit.

8.2 Guided construction of the model

The problem in words. *We decide* how much to spend on each advertising channel. *The objective* is to maximize the total expected response of the campaign. *The constraints*: the overall spend cannot exceed the budget, and each channel has a maximum useful investment.

Data (input of the model)

Symbol	Type	Meaning
n	$\in \mathbb{Z}_{\geq 1}$	number of advertising channels; the channels are indexed by $i \in \{1, 2, \ldots, n\}$
b	$\in \mathbb{Q}_{>0}$	total budget (thousands of €)
$r_i(\cdot)$	increasing concave function	expected response of channel i as a function of the spend; in the case study $r_i(x) = a_i \log(1 + k_i\, x)$ with given $a_i, k_i \in \mathbb{Q}_{>0}$
u_i	$\in \mathbb{Q}_{>0}$	maximum useful or allowed investment in channel i

Decision variables

We introduce the following n non-negative variables:

$$x_i = \text{spend on channel } i \text{ (thousands of €)}, \qquad \forall i \in \{1, 2, \ldots, n\}.$$

Using these variables, a convex NLP model for the problem is the following:

Allocation model with concave response

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

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

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

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

Description of the objective function and of the constraints of the model:

- the concave objective function (8.1a) maximizes the total response of the campaign; it is separable by channel and is a sum of concave functions, hence the problem has a certifiable global optimum;
- the linear constraint (8.1b) imposes that the overall spend does not exceed the budget: it is the only constraint that links the channels to each other — without it every channel would go to its own cap (one linear constraint);
- the linear constraints (8.1c) are the caps per channel: the limit beyond which the channel absorbs no further useful spend (n linear constraints);
- the constraints (8.1d) define the variables of the model.

The economic condition at the optimum

By the KKT conditions, at the optimum there exists $\lambda \geq 0$ (shadow price of the budget) such that, for every channel that is not stuck at the bounds 0 or u_i:

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

In words: **the last euro invested yields the same in all the active channels**. If this were not the case, moving one euro from the channel with the low marginal return to the one with the high marginal return would improve the response: we would not be at the optimum. The channels at the cap u_i may have a marginal return $> \lambda$ (we would like to invest more in them but we cannot); those at zero have an initial marginal return $< \lambda$ (they are not worth even the first euro).

8.3 Worked example by hand

Two logarithmic channels

$r_1(x) = 100\log(1+0.2x)$, $r_2(x) = 60\log(1+0.1x)$, budget $b = 50$ (thousands of €), without caps.

- **Step 1 — condition of equal marginal returns.**

$$\frac{100 \cdot 0.2}{1+0.2x_1} = \frac{60 \cdot 0.1}{1+0.1x_2} \quad \Longrightarrow \quad \frac{20}{1+0.2x_1} = \frac{6}{1+0.1x_2}.$$

- **Step 2 — budget.** $x_2 = 50 - x_1$. Substituting: $20\,(1+0.1(50-x_1)) = 6\,(1+0.2x_1)$, that is $20\,(6-0.1x_1) = 6+1.2x_1 \Rightarrow 120 - 2x_1 = 6 + 1.2x_1 \Rightarrow x_1 = \dfrac{114}{3.2} = 35.6$, $x_2 = 14.4$.
- **Step 3 — shadow price.** $\lambda = 20/(1+0.2\cdot 35.6) = 2.46$: one more euro of budget would yield ≈ 2.46 units of response — the same number, whichever channel it is fed into.

8.4 Case study and implementation

Four channels, $r_i(x) = a_i\log(1+k_ix)$, budget $b = 100$ (thousands of €); data in `data/budget_canali.csv`. The problem is concave and we solve it with the function constraints of Gurobi: an auxiliary variable $z_i = \log(g_i)$ with $g_i = 1 + k_ix_i$, so that the objective becomes linear in the z_i (certified global optimum):

```
m.Params.FuncNonlinear = 1          # log handled exactly
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 Results and verification of the KKT conditions

```
Total response: 1.449.8 (thousands of useful contacts)
    channel |  spend |   cap | response |  marginal
     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

Check: +1000 EUR of budget -> response +8.244 ~ lambda
```

Managerial insight

The theoretical condition can be seen in the numbers: the four marginal returns coincide to the fourth digit ($\lambda = 8.2693$), and the actual increase in response obtained with one thousand euros more (8.244) confirms the reading of the multiplier. A managerial note: TV receives *less* than social despite its higher cap — what matters is not the size of the channel but the speed at which it saturates (k_i).

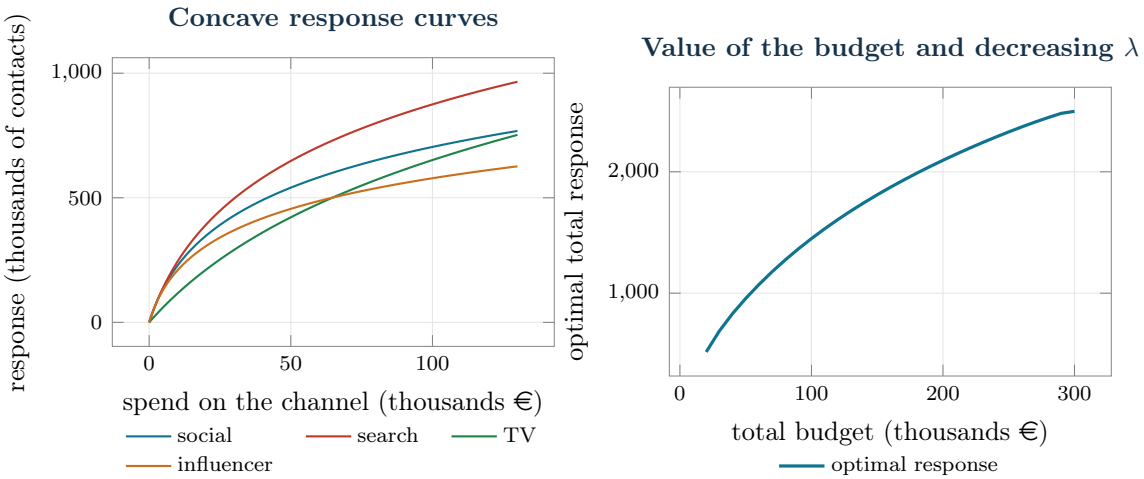


Figure 8.1: On the left: the response curves $r_i(x) = a_i \log(1 + k_i x)$ of the four channels — concave, with different saturation speeds. On the right: the optimal total response as the budget grows (value curve), concave: every tranche of budget yields less than the previous one.

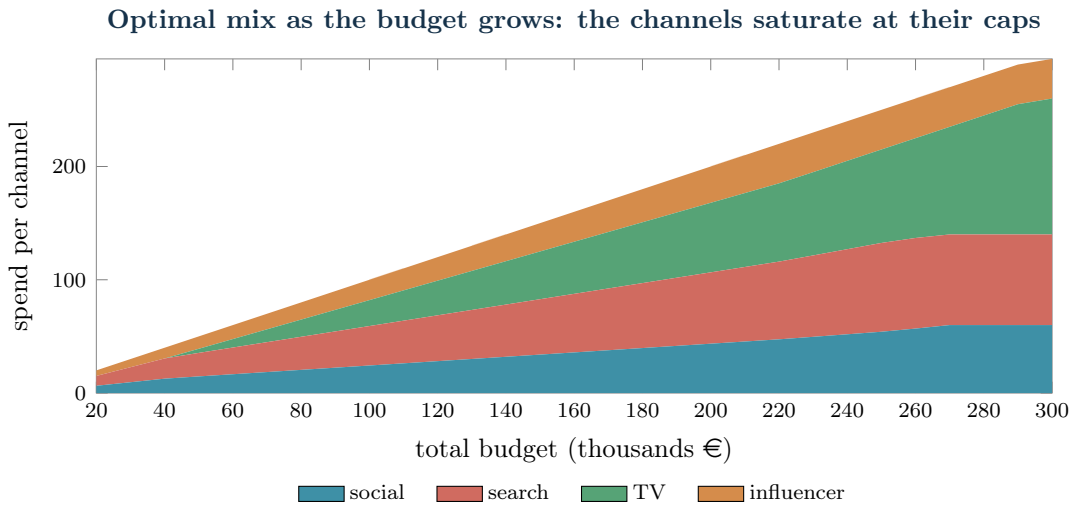


Figure 8.2: Optimal spend mix (stacked areas, thousands of €) as the total budget grows: the channels enter and grow together while keeping their marginal returns equal; the influencer channel saturates first at its cap (35), followed by the others.

8.6 Sensitivity analysis

```
b =  20: response   515.9   lambda = 18.971
b =  60: response 1.070.4   lambda = 10.909
b = 100: response 1.449.8   lambda =  8.244
b = 180: response 1.988.2   lambda =  5.538
b = 260: response 2.370.2   lambda =  4.049
b = 300: response 2.498.1   lambda =  0.000  (all channels at their caps)
```

Line-by-line explanation

The budget value curve is concave: λ falls from 19 to 4 as the budget grows. At $b = 300$ all the channels are at their caps ($60 + 80 + 120 + 35 = 295$): the budget stops being the scarce resource and λ collapses to zero. For a CFO: the *curve* of λ is the quantitative argument for deciding how much budget to grant to marketing — it is funded as long as λ exceeds the value of one euro invested elsewhere.

8.7 In summary

In summary

- Logarithmic responses are concave: maximizing their sum under a budget is a convex problem with a certified global optimum.
- Gurobi handles the logarithm with a functional constraint after the reformulation $z_i = \log(1 + k_i x_i)$.
- At the optimum the marginal returns of the channels below their caps coincide with λ, as the KKT conditions predict.
- The response grows with the budget in a concave way and λ decreases: its curve is

the tool for negotiating the budget.

8.8 Exercises

Exercise 8.1 — check

Solve the example by hand with $b = 80$: where does the additional budget end up? How does λ change?

Exercise 8.2 — exponential response

Replace the response of TV with $r(x) = 520\,(1 - e^{-0.025x})$ and repeat the optimization: does the mix change? The function has a natural ceiling: is the bound u_i still needed?

Exercise 8.3 — excluded channel

Add a fifth channel with initial marginal return $a_5 k_5 = 5 < \lambda$: verify that the optimum leaves it at zero and interpret its "reduced cost" $a_5 k_5 - \lambda$.

Exercise 8.4 — fair coverage

The budget serves three customer segments with different coverage per channel. Maximize the coverage of the worst segment ($\max z$ with $z \le \text{coverage}_g$): how much does fairness cost in terms of total response?

Chapter 9

Continuous location of a service

Class: convex NLP

Script: `python/lab09_location.py`

Online page

Prerequisites

- From the module *Convex Optimization*, Chapter 1 «Convex sets»: convex sets, on which the geographic constraints are built.
- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex quadratic programs, the KKT conditions and the multipliers as shadow prices.
- From this module, Chapter 3 «The solver for nonlinear models»: functional constraints and tolerances in `gurobipy`.

Objectives

At the end of this chapter you will be able to:

- formulate three location objectives and understand when they coincide;
- write Euclidean distances with conic constraints and solve them with Gurobi from good starting points;
- handle convex geographic constraints;
- build an efficiency-equity frontier with the ε-constraint method.

9.1 Managerial motivation

Where should a fast-charging station, a logistics micro-hub, a health-care post be placed in order to be "close" to demand? The answer depends on what close means: *mean* distance (efficiency), *maximum* distance (equity towards the most unlucky user) or *squared* distance (which leads to the centroid). Three objectives, three different points on the map: geometry makes the trade-off visible as in no other model.

Location decisions are among the most expensive ones to correct: a hub in the wrong place stays in the wrong place for years. The continuous version of the problem (free coordinates on the plane) is the ideal laboratory for a theme that reappears in all public and private decisions: the conflict between *efficiency* (serving well on average) and *equity* (leaving nobody behind).

9.2 Guided construction of the model

The problem in words. *We decide* the coordinates at which the new facility is placed. *The objective* — and this is the real managerial choice — may be to minimize the mean distance weighted by the users, or the distance of the farthest user, or the squared distance. *The constraints*, when present, delimit the feasible area (a rectangular zone, a maximum radius from an infrastructure).

Data (input of the model)

Symbol	Type	Meaning
n	$\in \mathbb{Z}_{\geq 1}$	number of demand points (districts), indexed by $i \in \{1, 2, \ldots, n\}$
(a_i, b_i)	$\in \mathbb{Q}^2$	coordinates of district i (km)
w_i	$\in \mathbb{Q}_{\geq 0}$	weight of district i: population, demand or priority
(a_0, b_0), r	$\in \mathbb{Q}^2$, $\in \mathbb{Q}_{>0}$	centre and radius of the possible geographic constraint (maximum distance from an infrastructure)

Decision variables

We introduce the following 2 free variables (they may take any real value) and, for the minimax version, a non-negative auxiliary variable:

$$\begin{cases} x = \text{east coordinate of the new facility (km)} \\ y = \text{north coordinate of the new facility (km)} \\ z = \text{maximum distance from the farthest district (minimax only)} \end{cases}$$

Using these variables, the models for the three classical objectives are the following:

Weber model (weighted mean distance)

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

$$\text{subject to} \quad x \gtreqless 0, \tag{9.1b}$$

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

Minimax model (maximum distance)

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

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

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

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

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

Description of the objective functions and of the constraints of the two models:

- the convex objective function (9.1a) minimizes the total distance weighted by the districts (efficiency); constraints (9.1b) and (9.1c) define the variables, which are free;
- the objective function (9.2a) minimizes the distance of the farthest district (equity); the convex constraints (9.2b) impose that every district is at most at distance z from the facility (n constraints); constraints (9.2c)–(9.2e) define the variables;
- a third variant, **quadratic**: minimize $\sum_{i=1}^{n} w_i\big[(x-a_i)^2+(y-b_i)^2\big]$, which has a closed-form solution (the weighted centroid).

Typical convex geographic constraints, to be added to any variant: rectangular zone $x^L \le x \le x^U$, $y^L \le y \le y^U$; maximum distance from an infrastructure $(x-a_0)^2+(y-b_0)^2 \le r^2$.

Line-by-line explanation

Quadratic ⇒ centroid. Setting the gradient to zero: $x^* = \sum_{i=1}^{n} w_i\, a_i / \sum_{i=1}^{n} w_i$ (and analogously for y): the only case with a closed formula, useful as a starting point for the others.
Weber. It has no closed form; the function is not differentiable at the points (a_i, b_i). At the optimum the weighted unit "pulls" of the districts cancel out: $\sum_{i=1}^{n} w_i\, \boldsymbol{u}_i = \mathbf{0}$, where $\boldsymbol{u}_i$ is the unit vector from the facility towards district i.
Minimax. The optimum is the centre of the *smallest enclosing circle* of all the points: it depends only on the extreme districts and ignores the weights — this is why it protects the outskirts.

9.3 Worked example by hand

Three customers on a road

Customers at km 0, 4 and 10 with weights 1, 1 and 3.
Weber in 1D = weighted median. The cost is $f(x) = |x| + |x-4| + 3|x-10|$. Moving by δ to the left from $x = 10$, the cost of the first two customers decreases by 2δ but that of the third increases by 3δ: it gets worse. From the right, symmetrically, it gets worse again. Optimum: $x^* = 10$, the point that leaves *at least half of the total weight* ($5/2 = 2.5$) on both sides — the weighted median.
Centroid. $x = (1\cdot 0 + 1\cdot 4 + 3\cdot 10)/5 = 6.8$: sensitive to *how far* the points are, not only to which side they are on.
Minimax. Midpoint of the extremes: $(0+10)/2 = 5$, weights ignored.
Three reasonable objectives, three different answers: 10, 6.8, 5. The choice of the objective IS the managerial decision.

9.4 Case study

Twelve districts with weights between 5 and 25 thousand inhabitants; the data are in the file `localizzazione_quartieri.csv` of the folder `data/`. Technical constraint: the station must lie within $r = 2$ km of the substation at $(7, 6)$.

The modelling trick: a variable $d_k \ge$ the Euclidean distance from district k, imposed with the (convex) conic constraint $d_x^2 + d_y^2 \le d_k^2$; Weber minimizes $\sum_k w_k d_k$, the minimax a variable z with $d_k \le z$.

```
d = m.addVars(n)                              # d_k >= distance from district
    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])    # cone
# Weber:    min sum w_k d_k
# minimax: min z  with  d_k <= z
# geographic constraint: (px-7)^2 + (py-6)^2 <= R^2
```

9.5 Results

```
Weighted centroid : (4.897, 5.397)
Weber             : (4.886, 5.310)  mean dist. 3.344 km  max 6.314 km
Minimax           : (5.496, 5.029)  mean dist. 3.391 km  max 5.680 km
Constrained Weber : (5.083, 5.429)  cost +0.2% (constraint active, optimum on
    boundary)
```

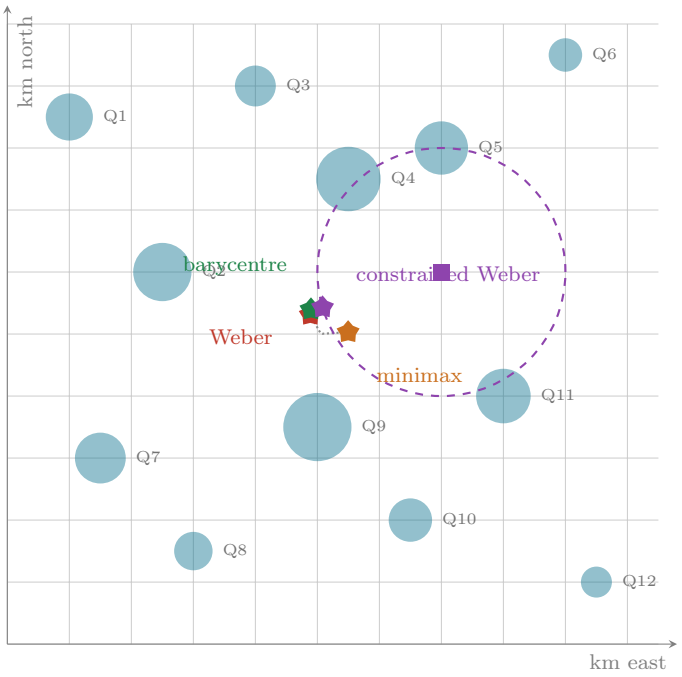


Figure 9.1: Map of the city: the teal circles are the 12 districts (area proportional to the population); the stars indicate the four optimal locations (Weber in red, minimax in orange, centroid in green, constrained Weber in purple); the dashed purple circle is the constraint of 2 km from the electrical substation; the dotted line is the trajectory of the compromise solutions between minimax and Weber.

> **Managerial insight**
>
> Weber and the centroid almost coincide here (the weights are evenly distributed), but the minimax moves more than half a kilometre towards the south-east in order to protect the peripheral districts. The constraint of the substation costs very little (+0.2% of total distance): discovering that a feared constraint is almost free is a managerial result as

important as the optimum itself.

9.6 Sensitivity analysis: the efficiency-equity frontier

We minimize the mean distance while imposing a cap D on the maximum distance (the ε-constraint method), and we vary D between the two extremes:

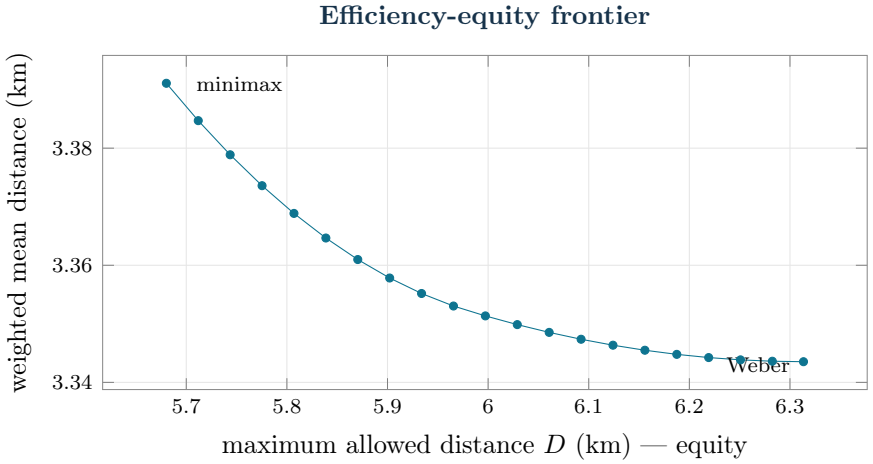


Figure 9.2: Efficiency-equity frontier (ε-constraint): minimum weighted mean distance attainable for every cap D imposed on the maximum distance. The extremes are the minimax solution (on the left) and the Weber point (on the right); the almost flat curve indicates that equity costs very little in terms of efficiency.

```
cap D = 5.680 km: mean 3.391   (minimax solution)
cap D = 5.997 km: mean 3.351
cap D = 6.314 km: mean 3.344   (Weber solution)
```

Line-by-line explanation

The frontier is almost flat: guaranteeing 0.63 km less to the farthest district costs only 47 metres of mean distance. When the frontier is this flat, the equitable solution is "almost free" — a very strong argument in a public discussion. The opposite case (a steep frontier) would signal a genuine efficiency-equity conflict.

9.7 In summary

In summary

- Weber minimizes the weighted mean distance, minimax the maximum distance, the quadratic objective gives the centroid.
- Euclidean distances are written with convex conic constraints.
- The choice of the objective is the managerial decision; a feared constraint can be almost free.

- The efficiency-equity frontier measures how much equity costs.

9.8 Exercises

Exercise 9.1 — check

Compute by hand the weighted centroid of the 12 districts from the CSV data and compare it with the output of the script.

Exercise 9.2 — a weight that changes everything

Raise the weight of Q12 (the district at the bottom right) from 5 to 40: how far do Weber and minimax move? Which of the two is more sensitive to the weights, and why?

Exercise 9.3 — two facilities

With *two* stations and the assignment of every district to the nearest one, does the problem remain convex? Try an alternating approach (assign–optimize) and discuss why it only finds local optima.

Exercise 9.4 — Manhattan distance

Replace the Euclidean distance with $|x - a_i| + |y - b_i|$ and show that the Weber problem becomes an LP (introduce auxiliary variables for the absolute values). Solve it with Gurobi and compare the optimum.

Exercise 9.5 — multiplier of the radius

Estimate by perturbation the multiplier of the constraint $r = 2$ km (solve with $r = 2.1$): how much is every additional 100 m of radius worth, in km of total distance?

Chapter 10

Electric vehicle charging with a smooth profile

Class: convex QP **Script:** `python/lab10_ev_charging_quadratic.py`
Online page

Prerequisites

- From the module *LP Modelling*, Chapter 5 «Smart charging of electric vehicles»: the linear charging model, with its data.
- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex quadratic programs, the KKT conditions and the multipliers as shadow prices.

Objectives

At the end of this chapter you will be able to:

- penalize the changes of an aggregate quantity with a quadratic term;
- read the energy-requirement multiplier in a QP;
- trace the frontier between spend and jitter of the draw.

10.1 What changes with respect to the linear model

The case is the one of LP Modelling, "Smart charging of electric vehicles": six vans to charge overnight, hourly prices, base load of the building and a 120 kW meter, with the same data. Minimum-cost charging chases the cheap hours and creates a "jittery" draw: total draw rises to 103.4 kW and changes by up to 54.9 kW from one hour to the next. Ramps this steep strain transformer and meter and, in many tariffs, are paid for.

Here we add to the objective a *quadratic* penalty on the hourly jumps of the total draw. The linear model only limited the peak; the quadratic term smooths the whole profile.

10.2 The model

What changes. Data, variables and constraints are those of the linear model: n vehicles indexed by v, m hours indexed by t (with $\Delta t = 1$ h), prices π_t, availability a_{vt}, energy e_v, powers $\bar{p}_v$, base load b_t, meter k, efficiency η; variables x_{vt} (power to vehicle v in hour t). A single datum is added:

Symbol	Type	Meaning
γ	$\in \mathbb{Q}_{\geq 0}$	weight of the jumps of the total draw (€ per kW^2)

We denote by $z_t = b_t + \sum_{v=1}^{n} x_{vt}$ the total draw in hour t: it is a linear expression of the variables, not a new variable. Using the same variables, a QP model for the problem is the following:

QP model of charging with a smooth profile

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

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

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

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

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

Description of the objective function and of the constraints of the QP model:

- the quadratic objective function (10.1a) adds to the energy spend the weight γ times the square of every hourly jump of the total draw;
- the linear constraints (10.1b) guarantee the energy required by every vehicle (n linear constraints);
- the linear constraints (10.1c) limit the draw at the meter (m linear constraints);
- the linear constraints (10.1d) limit the power of every vehicle and set it to zero when the vehicle is absent ($n\,m$ linear constraints);
- the constraints (10.1e) define the variables of the model.

The base load enters the penalty

The jumps are those of the *total* draw, base load included: even with zero charging the building profile has its own ramps, and the model can use charging to soften them. The QP is convex for the same reason as in chapter 4: a sum of squares of linear functions, linear constraints.

10.3 Worked example

One vehicle, two hours

We need $e = 10$ kWh ($\eta = 1$), prices $\pi_1 = 0.10$ and $\pi_2 = 0.20$ €/kWh, maximum power $\bar{p} = 12$ kW, zero base load. The linear model charges everything in the cheap hour: $x_1 = 10$, spend € 1.00.

- **Step 1 — solving we obtain**, with $\gamma = 0.01$:

$$\tilde{x}_1 = 6.25, \quad \tilde{x}_2 = 3.75, \quad \tilde{z} = 1.4375, \quad \tilde{\lambda} = 0.15,$$

  where $\tilde{\lambda}$ is the multiplier of the energy requirement; the spend is € 1.375.

- **Step 2 — one variable.** The energy constraint is active: with $x_1 = 5 + d$ and $x_2 = 5 - d$ the objective is

$$f(d) = 1.5 - 0.1\,d + 4\gamma\,d^2,$$

  with minimum at $d = 1/(80\gamma)$: with $\gamma = 0.01$ we find $d = 1.25$, as the solver does. This holds as long as $x_2 \geq 0$, that is for $\gamma \geq 1/400$; below that threshold everything is charged in hour 1.

- **Step 3 — the multiplier.** Stationarity with respect to x_1 and x_2:

$$0.10 - 2\gamma\,(x_2 - x_1) = \lambda = 0.20 + 2\gamma\,(x_2 - x_1) \;\Rightarrow\; \lambda = \tfrac{0.10+0.20}{2} = 0.15.$$

  One more kWh costs the *average* of the two prices, for every $\gamma \geq 1/400$: the solver gives $\tilde{\lambda} = 0.15$ also with $\gamma = 0.1$.

In the linear model the marginal kWh cost the price of the marginal hour (0.10); with the penalty it is split between the two hours to keep the jump small.

10.4 Implementation

With respect to the linear script only the objective changes; the full version is in the script named at the top of the chapter:

```
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 Results

```
gamma = 0.000: spend 20.36  peak 103.4 kW  largest jump 54.9  sum jumps^2
    7268.4
gamma = 0.001: spend 21.76  peak  77.3 kW  largest jump 15.4  sum jumps^2
    1188.4
gamma = 0.010: spend 24.67  peak  68.0 kW  largest jump 10.0  sum jumps^2
    298.2
```

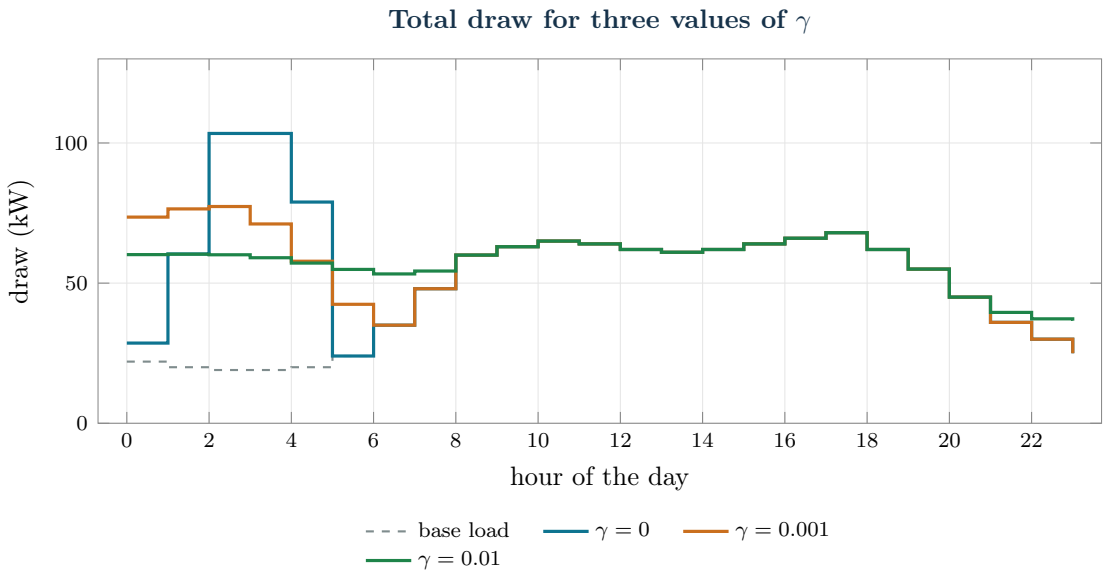


Figure 10.1: Total draw hour by hour: the linear model ($\gamma = 0$) concentrates charging in the cheapest hours; as γ grows charging spreads over the night and the morning ramps soften.

Managerial insight

With $\gamma = 0.01$ the smooth profile costs € 4.31 more per night and brings the peak to 68 kW, the lowest possible. The linear compromise cost $+\rho$ peak of LP Modelling reached the same peak spending less, because it only looks at the maximum; the QP also pays to soften every ramp. Choosing between the two models means deciding what really costs the company: the peak or the changes.

10.6 Sensitivity analysis

Energy-requirement multipliers

```
            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
Check (g = 0.01): e_1 + 1 kWh -> 0.0917;   e_1 + 0.01 kWh -> 0.0916
```

A single price for all vehicles

In the linear model every vehicle has its own marginal kWh, the price of its marginal hour divided by η. In the QP all the multipliers coincide: the penalty links the vehicles through the total draw, and one more kWh for any vehicle is obtained by redistributing the charging of all of them. The check by perturbation with 0.01 kWh confirms the value 0.0916.

The spend-jitter frontier

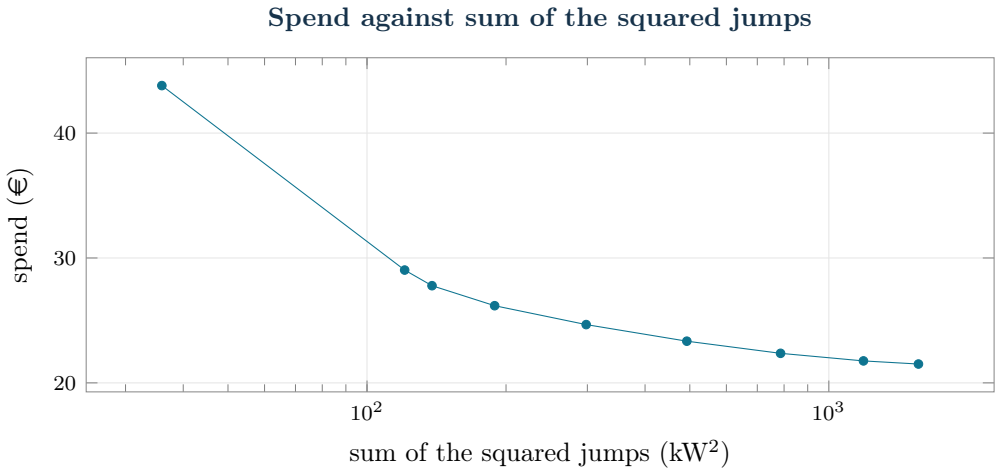


Figure 10.2: Every point is the optimal charging for a value of γ between 0.0005 and 0.5.

```
gamma = 0.0005: spend 21.51   sum jumps^2 1563.4   peak 82.3 kW
gamma = 0.0050: spend 23.34   sum jumps^2  492.6   peak 68.0 kW
gamma = 0.1000: spend 29.03   sum jumps^2  120.7   peak 68.0 kW
gamma = 0.5000: spend 43.80   sum jumps^2   36.0   peak 68.0 kW
```

Managerial insight

The frontier is steep at first: with € 1.15 more per night ($\gamma = 0.0005$) the sum of the squared jumps falls from 7,268 to 1,563. Beyond $\gamma = 0.1$ every further smoothing costs a lot: the base load has ramps of its own, and to follow them charging must move to expensive hours.

10.7 In summary

In summary

- The quadratic term penalizes every change of the draw, not only the peak.
- The marginal kWh is split between hours: in the example it costs the average price.
- In the QP the multipliers of the vehicles coincide.
- The frontier is steep at first: the first smoothing steps cost little.

10.8 Exercises

Exercise 10.1 — by hand

In the worked example set $\gamma = 0.05$: compute x_1, x_2, the spend, the optimal value and the energy multiplier, then check with the solver.

Exercise 10.2 — check

With $\gamma = 0.001$ verify by perturbation the energy multiplier of V3, with increments of 1 and of 0.01 kWh.

Exercise 10.3 — charging only

Penalize the jumps of charging only, $\gamma \sum_{t=2}^{m} \left(\sum_v x_{vt} - \sum_v x_{v,t-1} \right)^2$, with $\gamma = 0.01$. How do spend, peak and largest jump of the total draw change?

Exercise 10.4 — penalty on power

Replace the penalty on the jumps with $\gamma \sum_{t=1}^{m} z_t^2$ (losses growing with the square of the power), with $\gamma = 0.001$. What shape does the total draw take in the night hours?

Exercise 10.5 — meter

With $\gamma = 0.001$ reduce the meter to $k = 70$ kW. How much does the objective increase? Which grid multiplier becomes non-zero?

Chapter 11

Service capacity and waiting times

Class: convex NLP (M/M/1 queue) **Script:** `python/lab11_queues.py`

Online page

Prerequisites

- From the module *Convex Optimization*, Chapter 3 «Convex and concave functions»: convex functions of one variable and their minimization.
- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex quadratic programs, the KKT conditions and the multipliers as shadow prices.
- From this module, Chapter 3 «The solver for nonlinear models»: functional constraints and the estimate of shadow prices by perturbation.

Objectives

At the end of this chapter you will be able to:

- use the M/M/1 formulas inside an optimization problem;
- derive the optimum analytically and confirm it numerically;
- put a price on a service promise;
- protect the sizing against uncertainty about demand.

11.1 Managerial motivation

How much capacity should be assigned to a customer service, to a service desk, to a cloud service? More capacity costs money; too little capacity makes the waiting times explode. Queueing theory provides the quantitative relation between capacity and waiting, and optimization turns it into a decision. The central message is counter-intuitive for anyone who reasons "by efficiency": **the optimal utilization is not 100%**.

Every service with random arrivals — call centres, emergency rooms, help desks, the API of a cloud service — lives with the same dilemma: capacity costs, and so does waiting. Queueing theory gives the exact relation between the two; optimization chooses the right point. The chapter is short but its message (the optimal utilization is not 100%) is by itself worth a discussion in a board meeting.

11.2 Guided construction of the model

The problem in words. *We decide* how much service capacity to buy. *The objective* is to minimize the total cost: the capacity that is bought plus the value of the time the customers spend in the system. *The constraint* of a structural nature is only one, the stability of the queue ($\mu > \lambda$); in the variants the service promise $w \le w_{\max}$ is added.

We use the simplest queueing model, denoted by the standard notation **M/M/1**: *Markovian* arrivals (a Poisson process), *Markovian* service times (exponential) and *one* server. For this system the closed formulas reported in the table hold; what is new here is treating the capacity μ as an optimization variable.

Data (input of the model)

Symbol	Type	Meaning
λ	$\in \mathbb{Q}_{>0}$	mean arrival rate of the customers (requests/hour)
c	$\in \mathbb{Q}_{>0}$	cost of one unit of capacity (€ per unit of μ per hour)
h	$\in \mathbb{Q}_{>0}$	value of one hour spent by a customer in the system (€)

Decision variable and derived quantities

We introduce one continuous variable:

$$\mu = \text{service capacity (service rate, requests/hour)},$$

with the stability requirement $\mu > \lambda$. From M/M/1 queueing theory we obtain the mean time in the system $w(\mu) = 1/(\mu - \lambda)$ (in the literature: W) and the mean number of customers in the system $l(\mu) = \lambda/(\mu - \lambda)$ (Little's law; in the literature: L).

Using this variable, the model for the problem is the following:

Total cost model: capacity + waiting

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

$$\text{subject to} \quad \mu > \lambda. \tag{11.1b}$$

Description of the objective function and of the constraint of the model:

- the objective function (11.1a) is the sum of two terms: the first, $c\,\mu$, buys the capacity; the second, $h \cdot l(\mu)$, monetizes the total time the customers spend in the system; it is convex on $(\lambda, +\infty)$;
- the constraint (11.1b) defines the variable and guarantees the stability of the queue: with $\mu \le \lambda$ the line grows without bound.

Setting the derivative of the objective to zero gives the closed-form solution:

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

Line-by-line explanation

The formula has the classical structure "capacity = demand + safety stock": the mean demand λ is covered and a margin $\sqrt{h\lambda/c}$ is added, which grows with the value of the customers' time and decreases with the cost of capacity. The margin grows like $\sqrt{\lambda}$: large systems enjoy economies of scale in the "buffer" (pooling).

11.3 Worked example by hand

A service desk with 8 customers per hour

$\lambda = 8$/h, $c = 2$ € per unit of capacity, $h = 4$ €/h.

- **Step 1.** $\mu^* = 8 + \sqrt{4 \cdot 8/2} = 8 + 4 = 12$ customers/hour.
- **Step 2.** Utilization $\rho = 8/12 = 66.7\%$; mean time $w = 1/(12-8) = 15$ minutes; cost $C = 2 \cdot 12 + 4 \cdot 8/4 = 32$ €/h.
- **Step 3 — why not cut the capacity?** With $\mu = 9$ ($\rho = 89\%$, "more efficient"): $C = 18 + 32/1 = 50$ €/h, much worse. The apparent waste (33% of unused capacity) is exactly what keeps the queues short.

11.4 Case study and results

Customer service: $\lambda = 42$ requests/hour, $c = 3$ €, $h = 1.5$ €.

```
Analytical: mu* = 42 + sqrt(1.5*42/3) = 46.583   cost 153.495 EUR/h
Gurobi    : mu* = 46.582 (bilinear w(mu-lam) = 1, NonConvex=2: matches)
At optimum: rho = 90.2%   w = 13.1 minutes
```

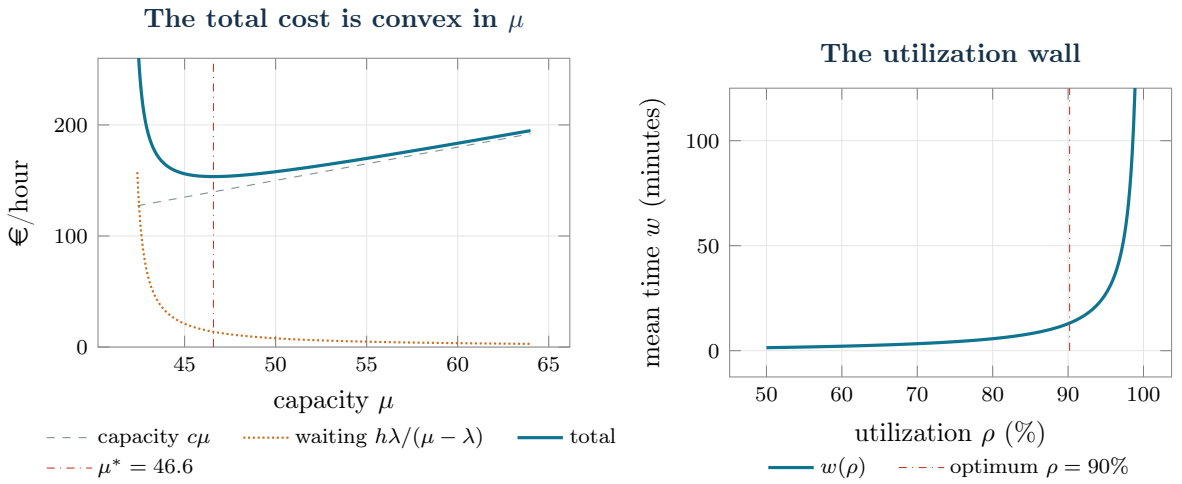


Figure 11.1: On the left: the two components of the cost (capacity, increasing; waiting, decreasing) and the total cost, convex, with its minimum at $\mu^* = 46.6$. On the right: the "utilization wall" — the mean time in the system w as a function of the utilization ρ: beyond 90% the waiting time explodes hyperbolically.

Managerial insight

The plot on the right is perhaps the most important one in these notes to show to a manager: between $\rho = 90\%$ and $\rho = 99\%$ the waiting time is multiplied by ten. "Saturating the resources" and "giving good service" are objectives in mathematical, not organizational, conflict: no reorganization removes the hyperbola $w = 1/(\mu - \lambda)$.

11.5 Sensitivity analysis: the price of a promise

If marketing promises a "mean time below $w_{\max}$", we need $\mu \geq \lambda + 1/w_{\max}$: the constraint becomes active when it is tighter than the economic optimum (13.1 minutes).

```
w_max = 12 min: mu = 47.00   cost 153.60   (almost free)
w_max =  9 min: mu = 48.67   cost 155.45
w_max =  6 min: mu = 52.00   cost 162.30
w_max =  4 min: mu = 57.00   cost 175.20
w_max =  3 min: mu = 62.00   cost 189.15
w_max =  2 min: mu = 72.00   cost 218.10   (+42% on the base cost)
```

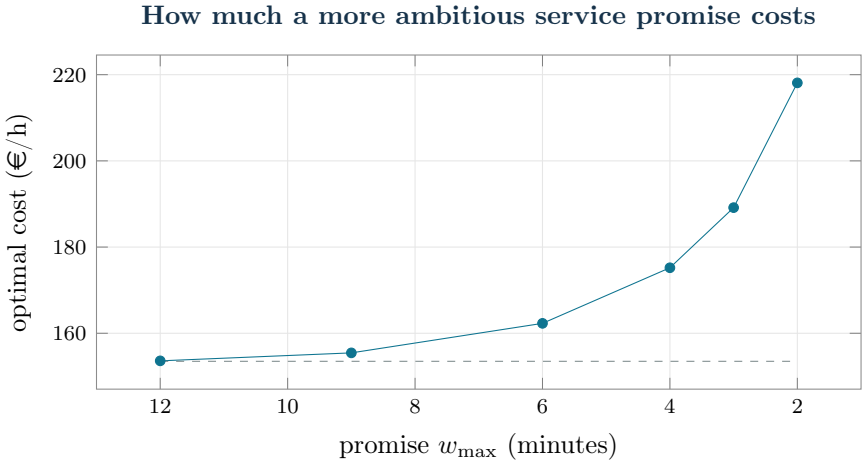


Figure 11.2: Optimal cost as a function of the service promise $w_{\max}$ (reversed axis: towards the right the promise becomes more ambitious). The dashed line is the cost without any promise: up to 12 minutes the promise is almost free, below 4 minutes every additional minute cut costs explosively.

The marginal cost of the promise

The shadow price of the constraint grows explosively: going from 12 to 9 minutes costs 1.85 €/h, from 3 to 2 minutes costs 28.95 €/h. When the constraint is active, $\mu = \lambda + 1/w_{\max}$ and $C = c\lambda + c/w_{\max} + h\lambda\, w_{\max}$, from which $dC/dw_{\max} = -c/w_{\max}^2 + h\lambda$ (with $w_{\max}$ in hours): at $w_{\max} = 6$ minutes it is equal to $-300 + 63 = -237$ €/h per hour of promise — exactly the finite-difference estimate of the script. This is the number to give to marketing *before* it signs the SLA (*Service Level Agreement*, the contractual commitment on the service level).

Robustness

If λ is uncertain in $[36, 48]$, sizing on the nominal optimum $\mu^* = 46.58$ is **unstable**: with $\lambda = 48 > \mu^*$ the queue diverges (infinite cost). Minimizing the cost of the worst case gives

$\mu_{\text{rob}} = 52.90$ with a worst-case cost of 173.39 €/h: an insurance premium of $\approx 13\%$ is paid in order to be protected from a disaster. Capacity decisions must always be checked against the *maximum* plausible demand, not only against the mean.

11.6 In summary

In summary

- The total cost $c\mu + h\lambda/(\mu - \lambda)$ is convex, with optimum $\mu^* = \lambda + \sqrt{h\lambda/c}$: demand plus a safety margin.
- The optimal utilization is not 100%: close to saturation the waiting time explodes.
- A promise $w \leq w_{\max}$ becomes $\mu \geq \lambda + 1/w_{\max}$, and its shadow price explodes for ambitious promises.
- Sizing on the mean demand can make the queue unstable: the robust solution costs more and protects from disaster.

11.7 Exercises

Exercise 11.1 — check

Derive on your own the formula $dC/dw_{\max} = -c/w_{\max}^2 + h\lambda$ of the previous explanation and verify it numerically at $w_{\max} = 4$ and 9 minutes by finite differences.

Exercise 11.2 — quadratic penalty

Replace the linear waiting cost with $h\,[1/(\mu - \lambda)]^2$: derive μ^* analytically and compare the optimal utilization with the linear case.

Exercise 11.3 — two queues vs one

Two separate service desks each receive $\lambda = 21$; alternatively, a single system receives $\lambda = 42$ with aggregate capacity. At equal capacity cost, which configuration gives shorter waiting times? (This is the quantitative argument for pooling.)

Exercise 11.4 — capacity-service frontier

Plot the frontier (cost, w) by varying $w_{\max}$ continuously and identify the point beyond which every additional promised minute cut costs more than 10 €/h.

Part III

Optimization and machine learning

Chapter 12

Support Vector Machine: optimization for machine learning

Class: convex QP

Script: `python/lab12_svm.py`
Online page

Prerequisites

- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex quadratic programs, the KKT conditions and the multipliers as shadow prices.
- From the module *Convex Optimization*, Chapter 6 «Convex and concave optimization problems»: convex optimization problems and their optimality conditions.
- From this module, Chapter 3 «The solver for nonlinear models»: quadratic objectives in `gurobipy`.

Objectives

At the end of this chapter you will be able to:

- formulate hard and soft margin classification as QPs;
- derive and solve the dual, identifying the support vectors;
- obtain non-linear boundaries with kernels while staying in the convex world;
- handle unbalanced classes with asymmetric costs and evaluate with the right metrics.

12.1 Managerial motivation

The Support Vector Machine connects convex optimization directly to machine learning: the classifier is obtained by solving a QP, with a clear geometric reading. The managerial applications are immediate: credit risk, churn, predictive maintenance, quality control, fraud. In this chapter we do *not* use ML libraries: every model is a QP written and solved with Gurobi, because the goal is to understand *what* a classifier optimizes.

A modelling note: the labels $y_i \in \{-1, +1\}$ are *observed data*, not variables — the variables (coefficients, intercept, slacks, multipliers) are all continuous.

This chapter closes the circle: the same tools used for production, logistics and finance — convex QPs, duality, multipliers — are the engine of one of the most elegant classifiers of machine learning. Understanding the SVM as an optimization problem, and not as a black box from a library, gives full control over what the model is really doing.

12.2 Hard margin: the geometry

The problem in words. *We decide* the coefficients of a hyperplane that separates two classes of observations. *The objective* is to maximize the margin, that is, the distance between the hyperplane and the closest points of the two classes (a classifier with a wide margin generalizes better). *The constraints* require every observation to lie on the right side of the hyperplane — in the soft margin, up to a slack paid in the objective.

Data (input of the model)

Symbol	Type	Meaning
n	$\in \mathbb{Z}_{\geq 1}$	number of observations of the training set, indexed by $i \in \{1, 2, \dots, n\}$
p	$\in \mathbb{Z}_{\geq 1}$	number of features, indexed by $j \in \{1, 2, \dots, p\}$
x_{ij}	$\in \mathbb{Q}$	value of feature j for observation i; $\boldsymbol{x}_i \in \mathbb{Q}^p$ is the vector of observation i
y_i	$\in \{-1, +1\}$	*observed* class label of i (a datum, not a variable)
C	$\in \mathbb{Q}_{>0}$	weight of the margin violations (soft margin only); upper case to follow the machine learning literature, an exception to the convention that reserves capital letters for sets and matrices

Decision variables

We introduce the following $p + 1$ free variables:

$$\begin{cases} w_j = \text{coefficient } j \text{ of the separating hyperplane} \\ b = \text{intercept of the hyperplane} \end{cases} \qquad \forall j \in \{1, 2, \dots, p\},$$

which define the hyperplane $\sum_{j=1}^{p} w_j \, x_j + b = 0$.

Using these variables, the hard margin QP model is the following:

Hard margin QP model

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

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

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

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

Description of the objective function and of the constraints of the QP model:

- the convex quadratic objective function (12.1a) minimizes $\frac{1}{2}\|\boldsymbol{w}\|^2$; since the geometric margin between the supporting hyperplanes is $2/\|\boldsymbol{w}\|$, minimizing it is equivalent to *maximizing the margin*;
- the linear constraints (12.1b) impose that every observation lies on the right side of the hyperplane, at a distance of at least one unit in the scale of $\boldsymbol{w}$ (n linear constraints);
- constraints (12.1c) and (12.1d) define the variables of the model, all free.

Two points, everything by hand

$\boldsymbol{x}_1 = (0.0)$ with $y_1 = -1$ and $\boldsymbol{x}_2 = (2.2)$ with $y_2 = +1$.

- **Step 1 — constraints.** $-(b) \geq 1$ and $2w_1 + 2w_2 + b \geq 1$, that is, $b \leq -1$ and $2(w_1 + w_2) \geq 1 - b \geq 2$.
- **Step 2 — symmetry and optimum.** To minimize $w_1^2 + w_2^2$ with $w_1 + w_2 \geq 1$ it is best to take $w_1 = w_2 = t$ and the inequalities active: $4t + b = 1$, $b = -1 \Rightarrow t = 1/2$. Hence $\boldsymbol{w} = (0.5; 0.5)$, $b = -1$: the hyperplane $0.5x_1 + 0.5x_2 = 1$ is the perpendicular bisector of the segment between the two points.
- **Step 3 — margin.** $2/\|\boldsymbol{w}\| = 2/\sqrt{0.5} = 2\sqrt{2} = 2.83$: exactly the distance between the two points. Both constraints are active: both points are *support vectors*. With real overlapping data, however, the hard margin is infeasible: the soft margin is needed.

12.3 Soft margin

With real overlapping data the hard margin is infeasible: one adds n non-negative slack variables,

$$\xi_i = \text{margin violation of observation } i, \qquad \forall i \in \{1, 2, \ldots, n\},$$

where $\xi_i > 1$ corresponds to a misclassification.

Soft margin QP model (primal)

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

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

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

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

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

Description of the objective function and of the constraints of the QP model:

- the convex quadratic objective function (12.2a) balances two terms: the simplicity of the model ($\frac{1}{2}\|\boldsymbol{w}\|^2$, wide margin) and the cost of the violations on the training set ($C\sum_{i=1}^{n}\xi_i$); C is the price assigned to the violations;

- the linear constraints (12.2b) require correct classification *up to the slack* ξ_i, paid in the objective (n linear constraints);
- constraints (12.2c), (12.2d) and (12.2e) define the variables of the model: w_j and b free, ξ_i non-negative.

Case study: credit risk

90 customers, two standardized indicators (capital soundness, punctuality of payments), 55 creditworthy and 35 defaulting, partially overlapping (`data/svm_clienti.csv`).

```
C =  0.05: w = (0.678, 0.474)  b = -0.148 | margin 2.418 | errors 2 | in the
    margin 22
C =  1.00: w = (1.433, 0.611)  b = -0.236 | margin 1.284 | errors 2 | in the
    margin  7
C = 20.00: w = (5.058, 2.338)  b = -2.936 | margin 0.359 | errors 0 | in the
    margin  3
```

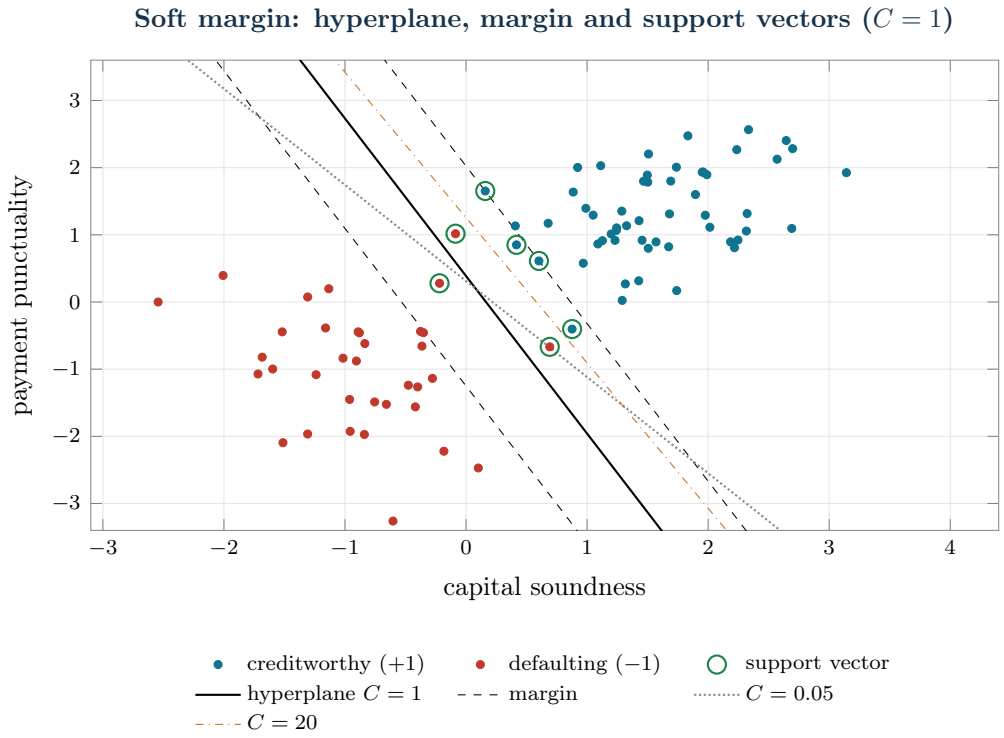


Figure 12.1: The 90 customers in the plane of the two indicators: creditworthy (teal), defaulting (red), support vectors circled in green. The black line is the optimal hyperplane for $C = 1$ with its margin lines (dashed); the grey and orange lines are the hyperplanes for $C = 0.05$ (wide margin) and $C = 20$ (narrow margin).

Managerial insight

A small C buys a wide margin while accepting 22 points inside the margin; a large C chases every point of the training set (zero errors, a very narrow margin): this is the **regularization–fit** trade-off, and it must be chosen with validation data, never by looking at the test set. Note that the three lines are similar but not equal: the slope changes because the "paid for" points change.

12.4 The dual formulation and the support vectors

By associating a multiplier α_i with each classification constraint (12.2b), one obtains n new variables:

$$\alpha_i = \text{multiplier of the classification constraint of observation } i, \qquad \forall i \in \{1, 2, \ldots, n\}.$$

Dual QP model of the soft margin

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

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

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

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

Description of the objective function and of the constraints of the dual model:

- the concave quadratic objective function (12.3a) depends on the data *only* through the inner products $\boldsymbol{x}_i' \boldsymbol{x}_j$: this is the observation that opens the door to the kernel;
- the linear constraint (12.3b) follows from stationarity with respect to the intercept b (one linear constraint);
- constraints (12.3c) bound each multiplier by the price C of the violations (n linear constraints);
- constraints (12.3d) define the variables of the model.

Primal-dual relation: $\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \, y_i \, \boldsymbol{x}_i$ and $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}'\boldsymbol{x} + b)$.

```
w from the dual = (1.433, 0.611)   b = -0.236     (identical to the primal)
dual value 4.7868 = primal value (strong duality)
alpha = 0         : 83 points  -> do NOT affect the boundary
0 < alpha < C     :  2 points  -> support vectors exactly on the margin
alpha = C         :  5 points  -> inside the margin or misclassified
```

Why they are called support vectors

The solution depends *only* on the 7 points with $\alpha_i > 0$: it is they that "support" the hyperplane; the other 83 could disappear without changing anything. The KKT complementarity organizes everything: $\alpha_i = 0 \Rightarrow$ point beyond the margin; $0 < \alpha_i < C \Rightarrow$ exactly on the margin (from these one recovers b); $\alpha_i = C \Rightarrow$ the constraint "pays" a slack $\xi_i > 0$. Moreover, the dual depends on the data only through the inner products $\boldsymbol{x}_i' \boldsymbol{x}_j$: this observation opens the door to the kernel.

12.5 Kernels: non-linear boundaries from a convex QP

By replacing $\boldsymbol{x}_i' \boldsymbol{x}_j$ with a kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ — for example the Gaussian RBF (*Radial Basis Function*) kernel $K(\boldsymbol{x}, \boldsymbol{z}) = e^{-\gamma \|\boldsymbol{x} - \boldsymbol{z}\|^2}$ — one classifies in a transformed space without ever

building it: $f(\boldsymbol{x}) = \operatorname{sign}\big(\sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\big)$. The boundary in the original space can be arbitrarily curved, but the training problem *remains a convex QP* if the kernel is valid (positive semidefinite matrix).

Case study: "ring-shaped" anomalies

90 process observations: normal operation in the centre, the anomalies arranged in a ring — no straight line can separate them.

```
gamma = 0.1: training errors  1   support vectors 20
gamma = 0.7: training errors  0   support vectors 15
gamma = 5.0: training errors  0   support vectors 61  <- "memorizes" the points
```

RBF kernel: the level-0 boundary for three values of γ

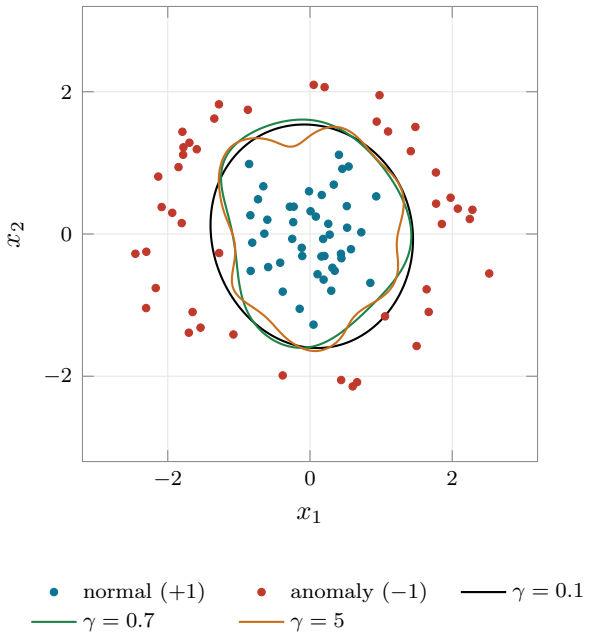


Figure 12.2: The "ring-shaped" dataset (normal operation in the centre, anomalies in a ring) and the decision boundaries of the RBF kernel for three values of γ: with $\gamma = 0.1$ the boundary is a regular circle, with $\gamma = 5$ it breaks up around the individual points (overfitting).

> With $\gamma = 5$ the boundary breaks up into islands around the individual points and the support vectors rise to 61 out of 90: the model is memorizing the training set (overfitting), while still remaining a convex QP — convexity of *training* and the ability to *generalize* are different properties. C and γ must be chosen together by cross-validation.

12.6 Unbalanced classes and asymmetric costs

In managerial applications the important class is rare and an error on the rare cases costs more. It is enough to weight the slacks: $C_{-}\sum_{i:\, y_i=-1} \xi_i + C_{+}\sum_{i:\, y_i=+1} \xi_i$.

```
equal costs (C=1)           : precision 1.00   recall defaulters 0.94
10x cost on the defaulters  : precision 0.97   recall defaulters 1.00
```

Managerial insight

With the 10× cost the boundary moves back towards the "healthy" class: no defaulter escapes (recall 100%) at the price of a few more false alarms (precision from 100% to 97%). The weights must reflect the *decision costs* (how much a loan granted to someone who will not repay costs vs. a good customer turned down), not the frequencies of the classes.

12.7 Support Vector Regression

SVR estimates a continuous function while ignoring the errors within a tube of width ε: $\min \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$ with $|y_i - (\boldsymbol{w}'\boldsymbol{x}_i + b)| \le \varepsilon + \xi_i^{(*)}$ for every i.

```
SVR line: demand = -10.86 * price + 220.32    (true: -11p + 220)
Tube eps = 8: 14 points out of 40 outside the tube -> only they determine the
    line
```

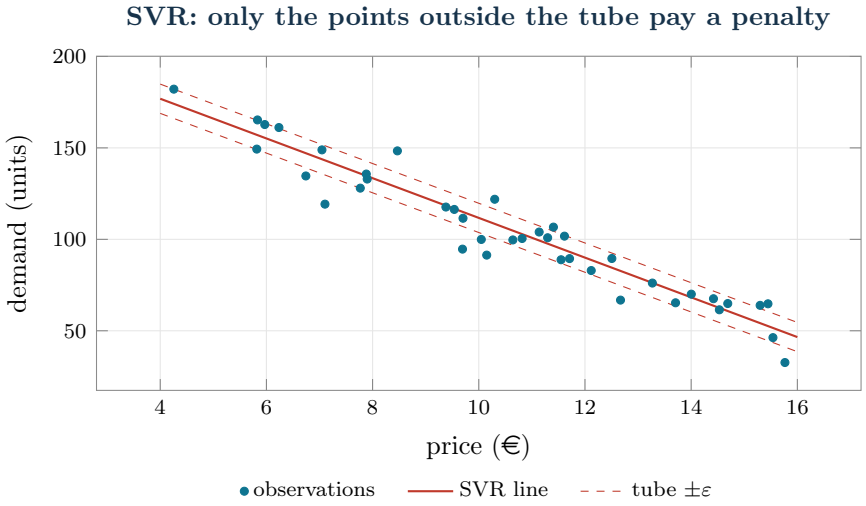


Figure 12.3: Support Vector Regression on the price–demand relation: the 40 observations, the estimated line and the tolerance tube $\pm\varepsilon = 8$: only the 14 points outside the tube pay a penalty and determine the line.

12.8 In summary

In summary

- Maximizing the margin is a convex QP; the soft margin pays the violations with a price C.
- In the dual only the support vectors have non-zero multipliers, and they alone determine the boundary.
- A kernel replaces the inner product in the dual: non-linear boundaries, still a convex QP; C and γ are chosen by cross-validation.
- With unbalanced classes two prices C_- and C_+ reflect the decision costs; one evaluates with precision and recall.

12.9 Exercises

Exercise 12.1 — check

Solve by hand the hard margin with $\boldsymbol{x}_1 = (0.0)$, $y_1 = -1$ and $\boldsymbol{x}_2 = (4.0)$, $y_2 = +1$, and check with Gurobi ($\boldsymbol{w} = (0.5;\ 0)$, $b = -1$, margin 4).

Exercise 12.2 — complementarity

For $C = 1$, list from the script the 7 points with $\alpha_i > 0$ and check numerically the three complementarity conditions ($\alpha_i = 0 \Rightarrow \xi_i = 0$ and margin ≥ 1; and so on).

Exercise 12.3 — validation curve

Split the 90 customers into training (60) and validation (30); plot the validation error for $C \in \{0.01;\ 0.1;\ 1;\ 10;\ 100\}$ and choose C. Is the training error monotone in C? And the validation error?

Exercise 12.4 — decision threshold

For the same model ($C = 1$), move the threshold: classify as defaulting if $\boldsymbol{w}'\boldsymbol{x} + b < \theta$ with $\theta \in [-1, 1]$. Plot precision and recall as a function of θ: where is the point that maximizes profit if an undetected defaulter costs 10 and a false alarm 1?

Exercise 12.5 — one-class

Implement the One-Class SVM on the ring-shaped dataset using only the normal points: $\min \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \xi_i - \rho$ subject to $\boldsymbol{w}'\phi(\boldsymbol{x}_i) \geq \rho - \xi_i$ (RBF kernel, $\gamma = 0.7$). Which fraction of anomalies does it detect for $\nu \in \{0.05;\ 0.1;\ 0.2\}$?

Chapter 13

Least-squares regression: estimating parameters with a QP

Class: convex QP **Script:** `python/lab13_least_squares.py`
Online page

Prerequisites

- From the module *LP Modelling*, Chapter 10 «Robust and quantile regression: estimating parameters with an LP»: regression with absolute deviations and quantiles as an LP, on the same data.
- From this module, Chapter 2 «Background: convexity, quadratic programming and KKT conditions»: convex quadratic programs, the KKT conditions and the multipliers as shadow prices.
- From the module *Linear Algebra*, Chapter 6 «System of linear equations»: linear systems, which give the normal equations.

Objectives

At the end of this chapter you will be able to:

- write least squares as a QP;
- read the KKT conditions as the normal equations;
- compare least squares, absolute deviations and quantiles on the same data;
- shrink the coefficients with a quadratic penalty (ridge).

13.1 What changes with respect to the linear model

The case is the one of LP Modelling, "Robust and quantile regression": sixty days of sales of a bakery, with temperature, price and weekend, five of them during two transport strikes. Data and description are the same. There the line is estimated by minimizing the sum of the *absolute* deviations, with an LP.

Here we minimize the sum of the *squared* deviations: it is least-squares regression, the most used in practice, and it is a convex QP. The comparison with robust and quantile regression shows what is gained and what is lost by changing the objective.

13.2 The model

What changes. The data are those of the linear model: n observations indexed by i, p features indexed by j, values x_{ij} and observed values y_i. The quantile τ is no longer needed. We introduce $p+1$ free variables for the function to estimate, w_j (coefficient of feature j) and b (intercept), and one free variable per observation:

$$r_i = \text{residual of observation } i, \qquad \forall i \in \{1, 2, \dots, n\}.$$

Using these variables, the QP model of least squares is the following:

QP model of least squares

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

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

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

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

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

Description of the objective function and of the constraints of the QP model:

- the quadratic objective function (13.1a) adds up the squares of the residuals;
- the linear constraints (13.1b) define the residual of every observation as the difference between the datum and the forecast (n linear constraints);
- the constraints (13.1c), (13.1d) and (13.1e) define the variables of the model, all free.

One variable per residual, no positive parts

In the linear model every deviation is split into a positive and a negative part, to pay its absolute value. The square does not distinguish the sign: a free residual r_i is enough. The objective $\sum_i r_i^2$ is convex and the constraints are linear: the KKT conditions are necessary and sufficient.

Without features: the mean comes back

With $p = 0$ only the intercept is left. Seven days of sales: 88, 96, 104, 112, 120, 132, 148 items (the same as in the example of the linear model).

- **Step 1 — solving we obtain**

  $$\tilde{b} = 114.2857 = \tfrac{800}{7}, \qquad \tilde{z} = 2619.4286,$$

  with residuals $\tilde{r} = (-26.29;\ -18.29;\ -10.29;\ -2.29;\ 5.71;\ 17.71;\ 33.71)$ and multipliers $\tilde{\pi}_i = 2\,\tilde{r}_i$.

- **Step 2 — stationarity in b.** The Lagrangian $\sum_i r_i^2 - \sum_i \pi_i(b + r_i - y_i)$ gives $2r_i = \pi_i$

and $\sum_i \pi_i = 0$, hence $\sum_i r_i = 0$: the estimate is the **mean** 800/7, and the residuals balance out.

- **Step 3 — the comparison.** On the same seven numbers robust regression ($\tau = 1/2$) gives the median 112, quantile regression with $\tau = 9/13$ gives 120.
- **Step 4 — an outlier.** If the last value goes from 148 to 248, the mean rises by 14.29 items (to 128.57), the median stays at 112.

13.3 The KKT conditions are the normal equations

With features, stationarity of the Lagrangian with respect to r_i, b and w_j gives

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

The optimal residuals sum to zero and are orthogonal to every feature: these are the **normal equations** of statistics, obtained here as KKT conditions. The multiplier $\tilde{\pi}_i$ says how much the sum of squares grows if the datum y_i increases by one unit: twice the residual. In the linear model the duals were $\pm 1/2$ or an interior value; here they are proportional to the residuals, and a distant datum weighs as much as it is distant.

13.4 Case study: least squares against absolute deviations

We estimate the relation with temperature alone ($p = 1$) on the 60 days.

```
QP (least squares)          : demand = 202.34 - 3.61 * temperature
LP (absolute deviations)    : demand = 209.77 - 3.63 * temperature
QP without the strike days  : demand = 213.58 - 3.72 * temperature
Sum of squares 68,286.6   root mean squared error 33.74 items
Mean absolute deviation on the 55 normal days: QP 14.81  LP 13.08  clean QP
    13.05
Share of the sum of squares due to the 5 strike days: 71.4%
KKT check: sum of the residuals 0.000; sum of residuals * temperature 0.000
```

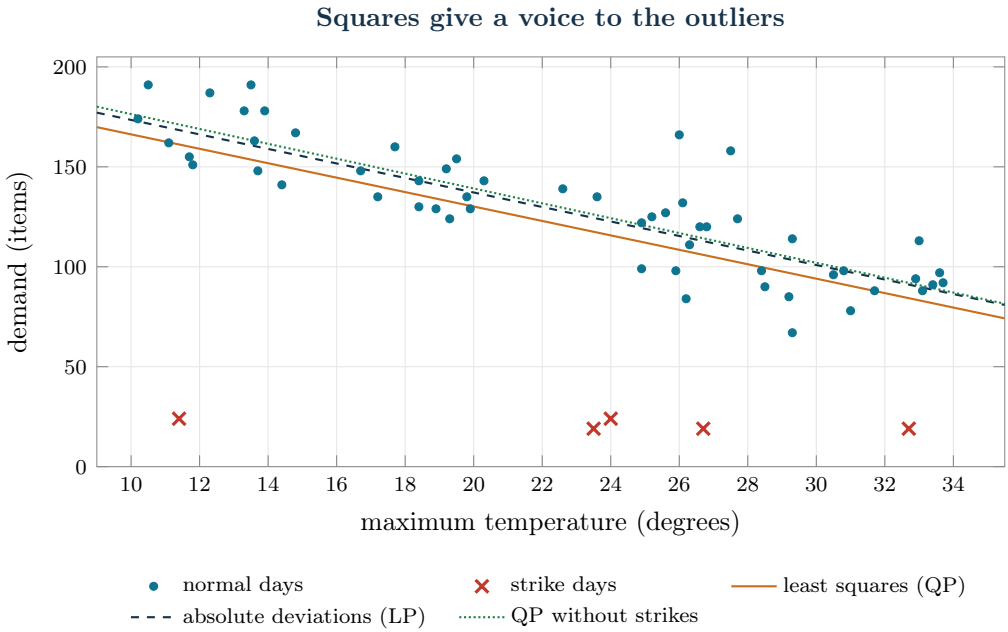


Figure 13.1: The 60 days of the bakery: the five strike days pull the least-squares line (orange) down by more than 11 items on the intercept; the line of absolute deviations (blue) stays close to the least-squares line computed without the strikes (green).

Managerial insight

Five days out of sixty produce 71% of the sum of squares: residuals of −137 items weigh as much as thousands of small residuals. Least squares must therefore come with a check for outliers; the robust regression of LP Modelling does not need one. In return, on clean data least squares has a closed-form solution (the normal equations) and estimates the mean, which is the right quantity when the errors are symmetric.

13.5 Forecast for the newsvendor: mean plus stock or quantile

On the 55 normal days, with three features (temperature, price, weekend):

```
Least squares: demand = 272.21 - 3.84 temp - 22.52 price + 31.57 weekend
               (standard deviation of the residuals 9.40)
9/13 quantile: demand = 281.19 - 3.96 temp - 23.21 price + 31.47 weekend
Typical day (28 degrees, price 3.00, weekend):
  forecast mean (QP)                    128.7 items
  mean + z*sigma, z = 0.5024            133.5 items
  forecast 9/13 quantile (LP)           132.1 items
```

Two roads to the same quantity

The newsvendor needs the 9/13 quantile of demand. Least squares gives the mean; to reach the quantile a stock $z\,\sigma$ is added, with z the quantile of the standard normal, assuming normal residuals. Quantile regression estimates the quantile directly, with no assumption on the shape of the errors. Here the two roads differ by 1.4 items; on the 55 days the share below the "mean $+\,z\sigma$" estimate is 0.709, close to the target 0.692.

13.6 Sensitivity analysis

A single outlying day

We move down the demand of the hottest normal day (33.7 degrees, demand 92) and estimate again on the 55 normal days.

shift	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

> **Managerial insight**
>
> The least-squares estimate moves *in proportion* to the error: every item less on the outlying day changes the slope by about 0.004. The LP estimate changes once, when the point goes below the line, and then stops moving: beyond that point only the sign of the residual matters. A hot day, far from the centre of the data, has a high *leverage* on least squares.

Ridge regression

Adding $\lambda \sum_{j=1}^{p} w_j^2$ to the objective gives *ridge* regression, still a convex QP. On seven standardized features (the three useful ones and four with no link to demand):

lambda	temp	price	weekend	follow.	rain	stock	day	rmse	zeros
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

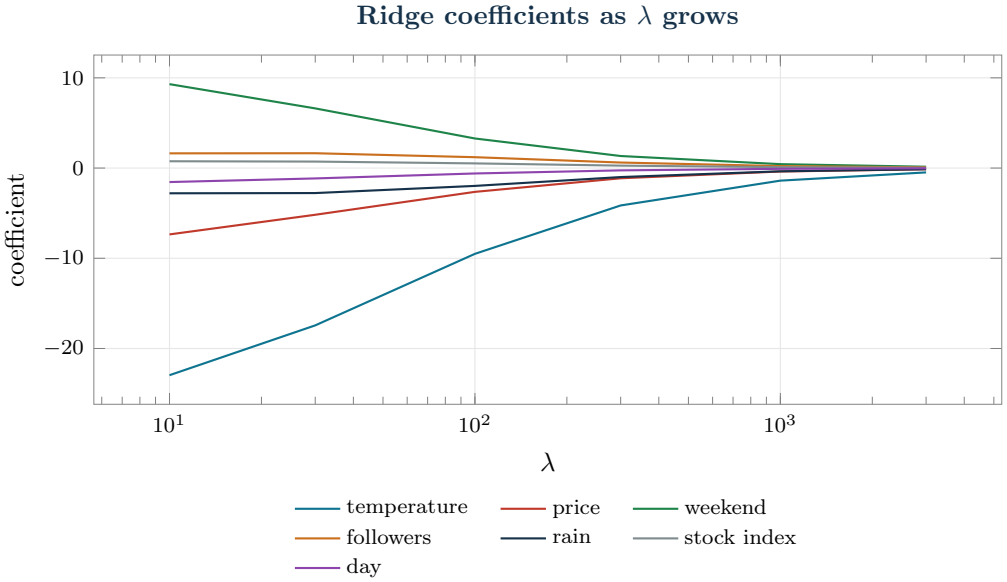


Figure 13.2: The ridge coefficients all shrink towards zero as λ grows, but none becomes zero.

Managerial insight

The quadratic penalty shrinks the coefficients but never sets them to zero: its derivative $2\lambda w_j$ vanishes exactly at $w_j = 0$, so there is no "fixed cost" pushing a coefficient exactly to zero. The budget on the absolute values of LP Modelling, instead, leaves out the useless features. To *select*, use the linear penalty; to *stabilize* estimates with correlated features, the quadratic one.

13.7 In summary

In summary

- Least squares is a convex QP, and its KKT conditions are the normal equations.
- It estimates the mean: excellent with symmetric errors, fragile with outliers, where absolute deviations hold better.
- For the newsvendor one uses the mean plus a stock, or the quantile directly.
- Ridge stabilizes the coefficients; a linear budget on them selects the features.

13.8 Exercises

Exercise 13.1 — by hand

With the seven sales of the example, compute by hand the sum of the squared residuals for $b = 112$ (the median) and for $b = 800/7$ (the mean). How much larger is the first?

Exercise 13.2 — normal equations

Write and solve the normal equations $\boldsymbol{A}'\boldsymbol{A}\,\boldsymbol{c} = \boldsymbol{A}'\boldsymbol{y}$ for the line with temperature alone on the 55 normal days, and check that they coincide with the solution of the QP.

Exercise 13.3 — weights

Minimize the weighted sum $\sum_i \omega_i r_i^2$ with $\omega_i = 0.01$ on the strike days and 1 on the others, on the 60 days with temperature alone. Where does the line end up?

Exercise 13.4 — Huber

Replace the square with the Huber loss: r_i^2 for $|r_i| \le 20$ and $40|r_i| - 400$ beyond. Write it as a QP with $r_i = q_i + u_i - v_i$, $|q_i| \le 20$, objective $\sum_i (q_i^2 + 40u_i + 40v_i)$, and estimate the line with temperature alone on the 60 days.

Exercise 13.5 — validation

Estimate least squares and absolute deviations with the three features on the first 40 days (strikes included) and compare the mean absolute deviation on the remaining 20 days, strikes excluded.

Part IV

Organization of the course

Chapter 14

Organization of the course

14.1 Essential path in four laboratory sessions

	Content	Learning objectives
Lab 1	Markowitz (ch. 6)	build a convex QP; trace a frontier; discuss the fragility of the estimates
Lab 2	Production with smoothing (ch. 4)	add a quadratic term to a linear model; read the multipliers of a QP; verify them by perturbation
Lab 3	Pricing *or* budget (chs. 7–8)	model nonlinear functions; study concavity; verify the KKT conditions numerically
Lab 4	Project of your choice	supply chain, location, EV charging, queues, SVM or least squares; management presentation of the results

Chapters 12–13 can replace session 3 in an edition oriented towards machine learning, or become in-depth topics for individual projects.

14.2 Recurring mistakes to avoid

1. Reading `.X` or `.Pi` without checking `m.Status`.
2. Forgetting `lb=-GRB.INFINITY` on free variables (b of the SVM, the residuals of least squares).
3. Using the multiplier of a QP for a large increment: it is exact only at the margin, because the optimal value has curvature.
4. Forgetting `NonConvex = 2` with bilinear terms, or believing a problem convex when it is not.
5. Updating the RHS of a constraint that contains constants on the left.
6. Optimizing a single objective when the problem has two (cost and smoothness, risk and return).
7. Choosing the hyperparameters (C, γ, λ) by looking at the test set.
8. Reporting six decimal digits of estimates that wobble at the second.

14.3 Course materials

`python/`	one script per chapter; `run_all.py` regenerates data, figures and notebooks
`notebooks/`	the same scripts as notebooks, to be opened in Colab
`data/`	all the data of the case studies in CSV
solver guide	in LP Modelling
`docs/pdf/`	these lecture notes and the slides of the chapters

Reproducibility: the two commands that redo data, figures and notebooks.

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