Convex Optimization

Lecture Notes

Master M1 — 2025–2026

Yaë Ulrich Gaba

"Theory without practice is useless, practice without theory is blind."
— Leonhard Euler

March 25, 2026

Contents

Preface

Course Objectives

Convex optimization occupies a central place in modern applied mathematics. It provides a rigorous theoretical framework and efficient algorithms for solving a broad class of optimization problems arising in machine learning, signal processing, finance, operations research, and many other fields.

This Master-level course is intended for students with solid backgrounds in real analysis, linear algebra, and topology. It covers the theoretical foundations of convexity, optimality conditions, duality theory, and the most important algorithmic methods used in practice.

The approach adopted is both mathematically rigorous and application-oriented. Each theoretical concept is accompanied by geometric interpretations, concrete examples, and Python implementations.

Course Organization

The course is organized into ten chapters, structured in a logical progression:

1. **Convex Sets** — Fundamental definitions, topological properties, operations preserving convexity, cones, polyhedra, separation theorems.

2. **Convex Functions** — Differential characterizations, fundamental inequalities (Jensen, gradient), continuity and differentiability.

3. **Subdifferentials and Subgradients** — Nonsmooth convex analysis, subdifferential calculus, Moreau–Rockafellar rules.

4. **Fenchel–Moreau–Rockafellar Duality** — Convex conjugation, biconjugation theorem, Fenchel duality.

5. **Optimality Conditions — KKT** — Necessary and sufficient conditions, constraint qualifications.

6. **Lagrangian Duality** — Dual problem, duality gap, strong duality, Slater's theorem, economic interpretation.

7. **Gradient Descent and Variants** — Fixed and adaptive step-size, Nesterov's accelerated gradient, stochastic gradient.

8. **Proximal Methods and ADMM** — Proximal operator, ISTA/FISTA algorithms, ADMM, decomposition.

9. **Interior Point Methods** — Barrier functions, central path method, polynomial complexity.

10. **Applications** — LASSO, SVM, logistic regression, portfolio optimization.

Prerequisites

The essential prerequisites for this course are:

- **Real Analysis** — Sequences, series, continuity, differentiability in $\mathbb{R}^n$, convergence theorems.

- **Linear Algebra** — Vector spaces, matrices, eigenvalues, matrix decompositions (SVD, Cholesky), quadratic forms.

- **Topology** — Open and closed sets, compactness, connectedness in $\mathbb{R}^n$, notion of relative interior.

- **Programming** — Basic knowledge of Python (NumPy, matplotlib). The `cvxpy` library will be used for implementations.

Notation

We use the following notation throughout the course:

Notation	Description
$\mathbb{R}^n$	Euclidean space of dimension n
$\langle x, y \rangle$	Inner product $\sum_{i=1}^n x_i y_i$
$\|x\|$	Euclidean norm $\sqrt{\langle x, x \rangle}$
$\|x\|_1$	ℓ_1 norm: $\sum_i \|x_i\|$
$\|x\|_\infty$	ℓ_∞ norm: $\max_i \|x_i\|$
$B(x, r)$	Open ball centered at x with radius r
$\overline{C}$	Closure of the set C
$\operatorname{int}(C)$	Interior of the set C
$\operatorname{ri}(C)$	Relative interior of C
$\operatorname{aff}(C)$	Affine hull of C
$\operatorname{conv}(S)$	Convex hull of S
$\operatorname{cone}(S)$	Conic hull of S
$\operatorname{dom}(f)$	Effective domain of f
$\operatorname{epi}(f)$	Epigraph of f
f^*	Fenchel conjugate of f
$\partial f(x)$	Subdifferential of f at x
$\nabla f(x)$	Gradient of f at x
$\nabla^2 f(x)$	Hessian of f at x
$\arg\min_x f(x)$	Set of minimizers of f
$A \succeq 0$	A is positive semidefinite
$A \succ 0$	A is positive definite
$\mathbb{S}_+^n$	Cone of $n \times n$ PSD matrices
ι_C	Convex indicator function of C
σ_C	Support function of C
prox_f	Proximal operator of f

Conventions

- Unless otherwise stated, all spaces considered are finite-dimensional over $\mathbb{R}$.
- Convex functions considered take values in $\mathbb{R} \cup \{+\infty\}$ (proper convex functions).
- The symbol $\square$ or $\square$ marks the end of a proof.
- Exercises are classified by increasing difficulty: $(\star)$ basic, $(\star\star)$ intermediate, $(\star\star\star)$ advanced.

- `algorithme` blocks present pseudo-code for the methods studied.
- `codeblock` blocks contain executable Python code.

Guiding Thread

The guiding thread of this course is the general convex optimization problem:

$$\min_{x \in \mathbb{R}^n} \; f(x) \quad \text{subject to} \quad g_i(x) \leq 0, \; i = 1, \ldots, m, \quad h_j(x) = 0, \; j = 1, \ldots, p,$$

where f and the g_i are convex and the h_j are affine. We study successively:

- the *geometry* of the feasible set (Chapters 1–2),
- the *analysis* of objective functions (Chapters 2–4),
- *optimality conditions* and *duality* (Chapters 5–6),
- *algorithms* for solving (Chapters 7–9),
- concrete *applications* (Chapter 10).

Pedagogical Approach

Each chapter follows a similar structure:

1. **Motivation** — Why this topic matters.
2. **Theory** — Definitions, theorems with complete proofs.
3. **Geometric Interpretation** — TikZ illustrations and intuitions.
4. **Examples** — Detailed examples and counterexamples.
5. **Implementation** — Python/cvxpy code.
6. **Exercises** — Problems of varying difficulty.

Main References

1. S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
2. R.T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.
3. J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*, Springer, 2001.
4. Y. Nesterov, *Introductory Lectures on Convex Optimization*, Springer, 2004.
5. A. Beck, *First-Order Methods in Optimization*, SIAM, 2017.

6. N. Parikh and S. Boyd, *Proximal Algorithms*, Foundations and Trends in Optimization, 2014.

7. H.H. Bauschke and P.L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, 2017.

8. D.P. Bertsekas, *Convex Optimization Algorithms*, Athena Scientific, 2015.

The Author
March 2026

Chapter 1

Convex Sets

Imagine stretching a rubber band between two pins stuck in a piece of cardboard. The band follows a straight line. Now imagine that the accessible region of the cardboard is constrained: if, no matter where you place the two pins *within* the region, the rubber band stays entirely inside it, then that region is *convex*. This is an idea as old as geometry itself — Euclid knew about convex polygons — but its consequences for optimization are profound and were only fully exploited in the twentieth century, by pioneers like Minkowski, Fenchel, Rockafellar, and Boyd.

1.1 Introduction and Motivation

Convex sets are the fundamental building blocks of convex optimization. An optimization problem is called convex when one minimizes a convex function over a convex set. The geometry of these sets determines the structure of solutions and guides algorithm design.

Intuition

A set is convex if, for every pair of points in the set, the line segment connecting them lies entirely within the set. This simple property has profound consequences: every local minimum is global, and hyperplane separation techniques become available.

1.2 Fundamental Definitions

Definition 1.1 (Convex Set)**.** A set $C \subseteq \mathbb{R}^n$ is *convex* if, for all $x, y \in C$ and all $\theta \in [0,1]$:

$$\theta x + (1-\theta)y \in C.$$

Definition 1.2 (Convex Combination)**.** A point x is a *convex combination* of points $x_1, \ldots, x_k$ if there exist $\theta_1, \ldots, \theta_k \geq 0$ with $\sum_{i=1}^{k} \theta_i = 1$ such that:

$$x = \sum_{i=1}^{k} \theta_i x_i.$$

Definition 1.3 (Convex Hull)**.** The *convex hull* of a set $S \subseteq \mathbb{R}^n$, denoted $\operatorname{conv}(S)$, is the set of all convex combinations of points in S:

$$\operatorname{conv}(S) = \left\{ \sum_{i=1}^{k} \theta_i x_i \;\middle|\; x_i \in S,\ \theta_i \geq 0,\ \sum_{i=1}^{k} \theta_i = 1,\ k \in \mathbb{N}^* \right\}.$$

Theorem 1.4 (Carathéodory)**.** *Let $S \subseteq \mathbb{R}^n$. Every point in* $\operatorname{conv}(S)$ *can be written as a convex combination of at most $n+1$ points in S.*

Proof. Let $x \in \operatorname{conv}(S)$. Then $x = \sum_{i=1}^{k} \theta_i x_i$ with $\theta_i > 0$, $\sum \theta_i = 1$, $x_i \in S$. If $k > n+1$, the vectors $x_2 - x_1, \ldots, x_k - x_1$ number $k-1 > n$ and are therefore linearly dependent. There exist $\mu_2, \ldots, \mu_k$, not all zero, such that $\sum_{i=2}^{k} \mu_i (x_i - x_1) = 0$. Set $\mu_1 = -\sum_{i=2}^{k} \mu_i$. Then $\sum_{i=1}^{k} \mu_i x_i = 0$ and $\sum_{i=1}^{k} \mu_i = 0$.

For $t \in \mathbb{R}$, $x = \sum_{i=1}^{k} (\theta_i - t\mu_i) x_i$. Choose $t^* = \min_{i:\mu_i > 0} \theta_i / \mu_i$. Then $\theta_i - t^* \mu_i \geq 0$ for all i, and at least one coefficient vanishes, reducing the number of terms. Iterate until $k \leq n+1$. □

1.3 Examples of Convex Sets

Example 1.5 (Classical Convex Sets)**.** The following sets are convex:

1. **Hyperplane**: $\{x \in \mathbb{R}^n \mid \langle a, x \rangle = b\}$ with $a \neq 0$.
2. **Halfspace**: $\{x \in \mathbb{R}^n \mid \langle a, x \rangle \leq b\}$.
3. **Euclidean ball**: $B(x_0, r) = \{x \mid \|x - x_0\| \leq r\}$.
4. **Ellipsoid**: $\{x \mid (x - x_0)^T P^{-1} (x - x_0) \leq 1\}$, $P \succ 0$.
5. **Polyhedron**: $\{x \mid Ax \leq b,\ Cx = d\}$.
6. **Simplex**: $\Delta_n = \{x \in \mathbb{R}^n_+ \mid \sum x_i = 1\}$.
7. **PSD cone**: $\mathbb{S}^n_+ = \{X \in \mathbb{S}^n \mid X \succeq 0\}$.

Verification for the ball. Let $x, y \in B(x_0, r)$ and $\theta \in [0, 1]$. By the triangle inequality:

$$\|\theta x + (1-\theta) y - x_0\| = \|\theta(x - x_0) + (1-\theta)(y - x_0)\| \leq \theta \|x - x_0\| + (1-\theta) \|y - x_0\| \leq \theta r + (1-\theta) r = r.$$

□

1.4 Convex Cones

Definition 1.6 (Cone)**.** A set $K \subseteq \mathbb{R}^n$ is a *cone* if for all $x \in K$ and all $\lambda \geq 0$, we have $\lambda x \in K$.

Definition 1.7 (Convex Cone)**.** A cone K is *convex* if K is a convex set, which is equivalent to: for all $x, y \in K$ and $\lambda, \mu \geq 0$:

$$\lambda x + \mu y \in K.$$

Definition 1.8 (Dual Cone)**.** The *dual cone* of K is:

$$K^* = \{y \in \mathbb{R}^n \mid \langle y, x \rangle \geq 0,\ \forall x \in K\}.$$

Proposition 1.9 (Properties of the Dual Cone)**.**

1. K^* is a closed convex cone.
2. If $K_1 \subseteq K_2$, then $K_2^* \subseteq K_1^*$.

3. If K is a closed convex cone, then $K^{**} = K$.

Example 1.10 (Classical Dual Cones).

- The dual cone of $\mathbb{R}^n_+$ is $\mathbb{R}^n_+$ (self-dual).
- The dual cone of the Lorentz cone $\mathcal{L}^n = \{(x,t) \in \mathbb{R}^{n-1} \times \mathbb{R} \mid \|x\| \leq t\}$ is $\mathcal{L}^n$ itself.
- The dual cone of $\mathbb{S}^n_+$ is $\mathbb{S}^n_+$.

1.5 Operations Preserving Convexity

Theorem 1.11 (Convexity-Preserving Operations). *The following operations preserve convexity:*

1. ***Intersection**: if C_α is convex for all $\alpha \in I$, then $\bigcap_{\alpha \in I} C_\alpha$ is convex.*
2. ***Affine image**: if C is convex and $f(x) = Ax + b$, then $f(C) = \{Ax + b \mid x \in C\}$ is convex.*
3. ***Affine preimage**: if C is convex, then $f^{-1}(C) = \{x \mid Ax + b \in C\}$ is convex.*
4. ***Minkowski sum**: if C_1, C_2 are convex, then $C_1 + C_2 = \{x + y \mid x \in C_1, y \in C_2\}$ is convex.*
5. ***Projection**: if $C \subseteq \mathbb{R}^n \times \mathbb{R}^m$ is convex, then $\{x \in \mathbb{R}^n \mid \exists y, (x,y) \in C\}$ is convex.*
6. ***Perspective function**: if C is convex and $P(x,t) = x/t$ for $t > 0$, then $P(C)$ is convex.*

Proof of (1). Let $x, y \in \bigcap_\alpha C_\alpha$ and $\theta \in [0,1]$. For each α, $x, y \in C_\alpha$, so $\theta x + (1-\theta)y \in C_\alpha$ by convexity of C_α. Thus $\theta x + (1-\theta)y \in \bigcap_\alpha C_\alpha$. □

Remark 1.12. The union of two convex sets is generally not convex. For example, $\{0\} \cup \{1\}$ is not convex in $\mathbb{R}$. Arbitrary intersection preserves convexity, but union does not.

1.6 Separation Theorems

Theorem 1.13 (Hyperplane Separation). *Let $C, D \subseteq \mathbb{R}^n$ be two nonempty disjoint convex sets. There exists $a \in \mathbb{R}^n$, $a \neq 0$, and $b \in \mathbb{R}$ such that:*

$$\langle a, x \rangle \leq b \; \forall x \in C \quad \text{and} \quad \langle a, x \rangle \geq b \; \forall x \in D.$$

Theorem 1.14 (Strict Separation). *Let C be a nonempty closed convex set and $x_0 \notin C$. There exists $a \in \mathbb{R}^n$, $a \neq 0$, and $b \in \mathbb{R}$ such that:*

$$\langle a, x_0 \rangle > b > \langle a, x \rangle \quad \forall x \in C.$$

Proof. Let $\bar{x} = \text{proj}_C(x_0)$ be the projection of x_0 onto C (which exists and is unique since C is closed and convex). Set $a = x_0 - \bar{x}$ and $b = \langle a, \bar{x} \rangle + \frac{1}{2}\|a\|^2$.

For all $x \in C$, the projection characterization gives:

$$\langle x_0 - \bar{x}, x - \bar{x} \rangle \leq 0,$$

that is, $\langle a, x \rangle \leq \langle a, \bar{x} \rangle$. Moreover:

$$\langle a, x_0 \rangle = \langle a, \bar{x} \rangle + \|a\|^2 > \langle a, \bar{x} \rangle + \tfrac{1}{2}\|a\|^2 = b > \langle a, \bar{x} \rangle \geq \langle a, x \rangle.$$

□

Intuition

Geometrically, the separation theorem states that one can always place a hyperplane between two disjoint convex sets. This is the higher-dimensional analogue of the fact that two disjoint intervals on the real line can be separated by a point.

1.7 Supporting Hyperplanes

Definition 1.15 (Supporting Hyperplane)**.** Let $C \subseteq \mathbb{R}^n$ be a convex set. A *supporting hyperplane* of C at a boundary point $x_0 \in \partial C$ is a hyperplane $H = \{x \mid \langle a, x\rangle = b\}$ such that:

$$\langle a, x_0\rangle = b \quad \text{and} \quad \langle a, x\rangle \leq b \; \forall x \in C.$$

Theorem 1.16 (Existence of Supporting Hyperplanes)**.** *Let C be a nonempty convex set with nonempty interior, and $x_0 \in \partial C$. Then there exists a supporting hyperplane of C at x_0.*

1.8 Support Function and Indicator Function

Definition 1.17 (Support Function)**.** The *support function* of a set C is:

$$\sigma_C(y) = \sup_{x \in C} \langle y, x\rangle.$$

Definition 1.18 (Convex Indicator Function)**.** The *indicator function* (in the sense of convex analysis) of C is:

$$\iota_C(x) = \begin{cases} 0 & \text{if } x \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

Proposition 1.19. If C is a nonempty closed convex set, then ι_C is convex, proper, and lower semicontinuous. Moreover, $\sigma_C = \iota_C^*$ (the Fenchel conjugate of ι_C).

1.9 Projection onto a Closed Convex Set

Theorem 1.20 (Convex Projection)**.** *Let $C \subseteq \mathbb{R}^n$ be a nonempty closed convex set. For every $y \in \mathbb{R}^n$, there exists a unique $\bar{x} \in C$ such that:*

$$\bar{x} = \arg\min_{x \in C} \|x - y\|^2.$$

This point, denoted $\operatorname{proj}_C(y)$*, is characterized by:*

$$\bar{x} \in C \quad \textit{and} \quad \langle y - \bar{x}, x - \bar{x}\rangle \leq 0, \quad \forall x \in C.$$

Proof. **Existence.** Let $d = \inf_{x \in C} \|x - y\|$. Take $(x_k) \subset C$ with $\|x_k - y\| \to d$. By the parallelogram identity:

$$\|x_k - x_l\|^2 = 2\|x_k - y\|^2 + 2\|x_l - y\|^2 - 4\left\|\tfrac{x_k + x_l}{2} - y\right\|^2.$$

Since $\frac{x_k+x_l}{2} \in C$ (convexity), $\left\|\frac{x_k+x_l}{2} - y\right\| \geq d$, so:

$$\|x_k - x_l\|^2 \leq 2\|x_k - y\|^2 + 2\|x_l - y\|^2 - 4d^2 \to 0.$$

(x_k) is Cauchy, hence converges to $\bar{x} \in C$ (since C is closed).

Uniqueness. If $\bar{x}_1, \bar{x}_2$ are two projections, the same argument shows $\|\bar{x}_1 - \bar{x}_2\| = 0$.

Characterization. $\bar{x}$ minimizes $\varphi(x) = \|x - y\|^2$ over C if and only if for all $x \in C$ and $t \in (0, 1]$: $\varphi(\bar{x} + t(x - \bar{x})) \geq \varphi(\bar{x})$, which gives $\langle y - \bar{x}, x - \bar{x}\rangle \leq 0$. □

Proposition 1.21 (Non-expansiveness of Projection)**.** The projection proj_C is non-expansive:

$$\|\mathrm{proj}_C(x) - \mathrm{proj}_C(y)\| \leq \|x - y\|, \quad \forall x, y \in \mathbb{R}^n.$$

In particular, proj_C is continuous (Lipschitz with constant 1).

1.10 Polyhedra and Representations

Definition 1.22 (Polyhedron)**.** A *polyhedron* is the intersection of finitely many halfspaces:

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}, \quad A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m.$$

A bounded polyhedron is called a *polytope*.

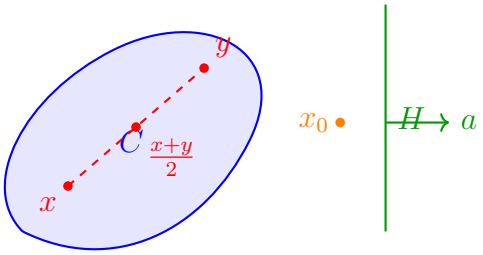
Theorem 1.23 (Minkowski–Weyl)**.** *A set $P \subseteq \mathbb{R}^n$ is a polytope if and only if it is the convex hull of finitely many points:*

$$P = \mathrm{conv}(\{v_1, \ldots, v_k\}).$$

More generally, every polyhedron admits a representation of the form:

$$P = \mathrm{conv}(\{v_1, \ldots, v_k\}) + \mathrm{cone}(\{d_1, \ldots, d_l\}).$$

1.11 Geometric Interpretation



1.12 Python Implementation

Simplex projection and convexity verification

```
import numpy as np
import matplotlib.pyplot as plt
```

```python
def project_simplex(v):
    """Project onto the standard simplex."""
    n = len(v)
    u = np.sort(v)[::-1]
    cumsum = np.cumsum(u)
    rho = np.max(np.where(u + (1 - cumsum) / (np.arange(n) + 1) > 0))
    theta = (cumsum[rho] - 1) / (rho + 1)
    return np.maximum(v - theta, 0)

def is_convex_hull_member(point, vertices):
    """Check if point is in conv(vertices) via LP."""
    import scipy.optimize as opt
    k = len(vertices)
    V = np.array(vertices).T  # n x k
    n = V.shape[0]
    c = np.zeros(k)
    A_eq = np.vstack([V, np.ones((1, k))])
    b_eq = np.append(point, 1.0)
    bounds = [(0, None)] * k
    res = opt.linprog(c, A_eq=A_eq, b_eq=b_eq, bounds=bounds)
    return res.success

# Demonstration: simplex projection
np.random.seed(42)
v = np.random.randn(3)
p = project_simplex(v)
print(f"Original point: {v}")
print(f"Projection:     {p}")
print(f"Sum = {p.sum():.6f}, min = {p.min():.6f}")
```

1.13 Exercises

Exercise 1.1 ($\star$). Show that the intersection of two halfspaces is convex.

Exercise 1.2 ($\star$). Show that the image of a convex set under a linear map is convex.

Exercise 1.3 ($\star\star$). Let $C = \{x \in \mathbb{R}^n \mid \|x\|_1 \leq 1\}$. Show that C is a polytope and determine its vertices.

Exercise 1.4 ($\star\star$). Let C be a closed convex set and $x_0 \notin C$. Prove that there exists a unique closest point $\bar{x} \in C$ to x_0. Show that $\langle x_0 - \bar{x}, x - \bar{x} \rangle \leq 0$ for all $x \in C$.

Exercise 1.5 ($\star\star$). Compute the dual cone of the Lorentz cone $\mathcal{L}^n = \{(x, t) \in \mathbb{R}^{n-1} \times \mathbb{R} \mid \|x\| \leq t\}$.

Exercise 1.6 ($\star\star\star$). Prove Carathéodory's theorem geometrically in dimension 2, then generalize the proof to dimension n.

Exercise 1.7 ($\star\star\star$). Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuous. Show that the sublevel set $\{x \mid f(x) \leq \alpha\}$ is closed. Give an example showing it is not necessarily convex even if f is continuous.

Chapter 2

Convex Functions

2.1 Introduction

"The great watershed in optimization is not between linearity and nonlinearity, but between convexity and nonconvexity," wrote R. Tyrrell Rockafellar in 1993. This deceptively simple observation captures a profound truth: convex functions enjoy a property so powerful that it transforms the entire landscape of optimization. Every local minimum is automatically global. There are no hidden valleys, no deceptive plateaus, no traps. If you find a point where the function cannot decrease locally, you have found the best point, period.

The theory of convex functions was forged over more than a century, from the inequalities of Jensen (1906) and the geometric insights of Minkowski, through the duality framework of Fenchel in the 1940s, to the monumental treatise of Rockafellar in 1970. What emerges is a remarkably elegant structure: convex functions can be characterised algebraically (via inequalities), analytically (via derivatives), and geometrically (via epigraphs)—and all three viewpoints converge to the same theory. This chapter builds that theory from the ground up.

2.2 Definitions and First Properties

Definition 2.1 (Convex Function)**.** A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is *convex* if its effective domain $\mathrm{dom}(f) = \{x \mid f(x) < +\infty\}$ is convex and if for all $x, y \in \mathrm{dom}(f)$ and all $\theta \in [0, 1]$:

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y).$$

Definition 2.2 (Strict Convexity)**.** f is *strictly convex* if the above inequality is strict for $x \neq y$ and $\theta \in (0, 1)$.

Definition 2.3 (Strong Convexity)**.** f is *strongly convex* with parameter $m > 0$ (or m-strongly convex) if $f(x) - \frac{m}{2}\|x\|^2$ is convex, i.e., for all x, y and $\theta \in [0, 1]$:

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y) - \frac{m}{2}\theta(1-\theta)\|x-y\|^2.$$

Convexity Hierarchy

$$\text{strongly convex} \Rightarrow \text{strictly convex} \Rightarrow \text{convex}.$$

The reverse implications are false in general.

2.3 Epigraph and Characterization

Definition 2.4 (Epigraph)**.** The *epigraph* of a function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is:

$$\text{epi}(f) = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \mid f(x) \leq t\}.$$

Theorem 2.5 (Epigraphic Characterization)**.** *f is convex if and only if* $\text{epi}(f)$ *is a convex subset of* $\mathbb{R}^{n+1}$.

Proof. ($\Rightarrow$) Let $(x_1, t_1), (x_2, t_2) \in \text{epi}(f)$ and $\theta \in [0, 1]$. Then:

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2) \leq \theta t_1 + (1-\theta) t_2,$$

so $(\theta x_1 + (1-\theta)x_2, \theta t_1 + (1-\theta)t_2) \in \text{epi}(f)$.

($\Leftarrow$) Let $x_1, x_2 \in \text{dom}(f)$. Then $(x_1, f(x_1)), (x_2, f(x_2)) \in \text{epi}(f)$. By convexity of the epigraph:

$$(\theta x_1 + (1-\theta)x_2, \theta f(x_1) + (1-\theta) f(x_2)) \in \text{epi}(f),$$

which means $f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2)$. □

2.4 Sublevel Sets

Definition 2.6 (Sublevel Set)**.** The *α-sublevel set* of f is:

$$S_\alpha = \{x \in \text{dom}(f) \mid f(x) \leq \alpha\}.$$

Proposition 2.7. If f is convex, then all its sublevel sets are convex.

Converse is False

The converse is false: $f(x) = e^x$ has convex sublevel sets (intervals), but $g(x) = \sqrt{|x|}$ also has convex sublevel sets without being convex. A function with convex sublevel sets is called *quasiconvex*.

2.5 Differential Characterizations

2.5.1 First-Order Condition

Theorem 2.8 (First-Order Condition)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable on an open convex set Ω. Then f is convex on Ω if and only if:*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \quad \forall x, y \in \Omega.$$

Proof. ($\Rightarrow$) By convexity, for $t \in (0, 1]$:

$$f(x + t(y - x)) \leq f(x) + t(f(y) - f(x)).$$

So $\frac{f(x+t(y-x))-f(x)}{t} \leq f(y) - f(x)$. Taking $t \to 0^+$: $\langle \nabla f(x), y - x \rangle \leq f(y) - f(x)$.
($\Leftarrow$) Let $x, y \in \Omega$ and $z = \theta x + (1 - \theta)y$. Then:

$$f(x) \geq f(z) + \langle \nabla f(z), x - z \rangle,$$
$$f(y) \geq f(z) + \langle \nabla f(z), y - z \rangle.$$

Multiplying by θ and $1 - \theta$ respectively and summing:

$$\theta f(x) + (1 - \theta)f(y) \geq f(z) + \langle \nabla f(z), \theta x + (1 - \theta)y - z \rangle = f(z).$$

□

Intuition

The first-order condition means that the graph of f always lies above its tangent hyperplanes. Geometrically, the surface $z = f(x)$ curves upward.

2.5.2 Second-Order Condition

Theorem 2.9 (Second-Order Condition)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable on an open convex set Ω. Then:*

1. *f is convex on $\Omega \Leftrightarrow \nabla^2 f(x) \succeq 0$ for all $x \in \Omega$.*
2. *f is strongly convex with parameter $m \Leftrightarrow \nabla^2 f(x) \succeq mI$ for all $x \in \Omega$.*
3. *If $\nabla^2 f(x) \succ 0$ for all x, then f is strictly convex (the converse is false).*

Proof of (1), $\Rightarrow$. By the first-order condition, for all $h \in \mathbb{R}^n$ and small enough $t > 0$:

$$f(x + th) \geq f(x) + t\langle \nabla f(x), h \rangle.$$

By Taylor expansion:

$$f(x + th) = f(x) + t\langle \nabla f(x), h \rangle + \frac{t^2}{2} h^T \nabla^2 f(x) h + o(t^2).$$

Hence $h^T \nabla^2 f(x) h + o(t^2)/t^2 \geq 0$. Letting $t \to 0$: $h^T \nabla^2 f(x) h \geq 0$. □

2.6 Jensen's Inequality

Theorem 2.10 (Jensen's Inequality)**.** *Let f be convex and $x_1, \ldots, x_k \in \text{dom}(f)$ with $\theta_1, \ldots, \theta_k \geq 0$, $\sum \theta_i = 1$. Then:*

$$f\left(\sum_{i=1}^{k} \theta_i x_i\right) \leq \sum_{i=1}^{k} \theta_i f(x_i).$$

Corollary 2.11 (Probabilistic Jensen)**.** *Let X be a random variable taking values in $\text{dom}(f)$ with $\mathbb{E}[|X|] < \infty$. If f is convex:*

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)].$$

2.7 Operations Preserving Convexity of Functions

Theorem 2.12 (Operations on Convex Functions)**.**

1. ***Nonnegative weighted sum**: if $f_1, \dots, f_k$ are convex and $\alpha_1, \dots, \alpha_k \geq 0$, then $\sum \alpha_i f_i$ is convex.*
2. ***Pointwise maximum**: if $f_1, \dots, f_k$ are convex, then $g(x) = \max_i f_i(x)$ is convex.*
3. ***Supremum**: if $(f_\alpha)_{\alpha \in I}$ is a family of convex functions, then $g(x) = \sup_\alpha f_\alpha(x)$ is convex.*
4. ***Affine composition**: if f is convex, then $g(x) = f(Ax + b)$ is convex.*
5. ***Infimal convolution**: if f, g are convex, then $(f \square g)(x) = \inf_y \{f(y) + g(x - y)\}$ is convex.*

2.8 Convexity and Smoothness

Definition 2.13 (L-Smoothness)**.** A differentiable convex function f is *L-smooth* if its gradient is L-Lipschitz:

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|, \quad \forall x, y.$$

Theorem 2.14 (L-Smoothness)**.** *For convex and differentiable f, the following are equivalent:*

1. *∇f is L-Lipschitz.*
2. *$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$ for all x, y.*
3. *$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2L} \|\nabla f(y) - \nabla f(x)\|^2$ for all x, y.*
4. *$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \frac{1}{L} \|\nabla f(x) - \nabla f(y)\|^2$ for all x, y.*

If f is twice differentiable, these are equivalent to $\nabla^2 f(x) \preceq LI$.

Fundamental Sandwich Bound

For f convex, m-strongly convex, and L-smooth:

$$f(x) + \langle \nabla f(x), y - x \rangle + \frac{m}{2} \|y - x\|^2 \leq f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2.$$

The condition number $\kappa = L/m$ measures problem difficulty.

2.9 Important Examples

Example 2.15 (Classical Convex Functions)**.**

1. **Affine functions**: $f(x) = \langle a, x \rangle + b$ (convex and concave).
2. **Norms**: $\|\cdot\|_p$ for $p \geq 1$ (convex).
3. **Quadratic**: $f(x) = \frac{1}{2} x^T Q x + q^T x + c$ with $Q \succeq 0$. Strongly convex iff $Q \succ 0$, with $m = \lambda_{\min}(Q)$, $L = \lambda_{\max}(Q)$.

4. **Exponential**: $f(x) = e^{ax}$ (convex on $\mathbb{R}$).
5. **Log-sum-exp**: $f(x) = \log(\sum_i e^{x_i})$ (convex on $\mathbb{R}^n$).
6. **Negative entropy**: $f(x) = \sum_i x_i \log x_i$ on $\mathbb{R}^n_{++}$.
7. **Logistic loss**: $f(x) = \log(1 + e^{-x})$ (convex, smooth).

2.10 Continuity of Convex Functions

Theorem 2.16 (Continuity)**.** *Every convex function $f : \mathbb{R}^n \to \mathbb{R}$ (finite everywhere) is continuous. More precisely, every proper convex function is continuous on the interior of its effective domain.*

Theorem 2.17 (Almost Everywhere Differentiability)**.** *Every convex function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable almost everywhere (in the Lebesgue sense). In dimension 1, f has left and right derivatives at every point.*

2.11 Convex Conjugates — Preview

Definition 2.18 (Fenchel Conjugate)**.** The *Fenchel conjugate* (or Legendre–Fenchel transform) of $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is:

$$f^*(y) = \sup_{x \in \mathbb{R}^n} \{\langle y, x \rangle - f(x)\}.$$

Remark 2.19. f^* is always convex and lower semicontinuous (as a supremum of affine functions in y), even if f is not. Convex conjugation will be studied in detail in Chapter 4.

2.12 Python Implementation

Convexity verification and visualization

```python
import numpy as np
import matplotlib.pyplot as plt

def check_convexity_numerical(f, x_range, n_tests=1000):
    """Numerical convexity check in 1D."""
    for _ in range(n_tests):
        x = np.random.uniform(*x_range)
        y = np.random.uniform(*x_range)
        t = np.random.uniform(0, 1)
        lhs = f(t * x + (1 - t) * y)
        rhs = t * f(x) + (1 - t) * f(y)
        if lhs > rhs + 1e-10:
            return False
    return True

# Examples
functions = {
```

```
    'x^2': lambda x: x**2,
    'exp(x)': lambda x: np.exp(x),
    '|x|': lambda x: np.abs(x),
    'log(1+exp(x))': lambda x: np.log(1 + np.exp(x)),
}

fig, axes = plt.subplots(2, 2, figsize=(10, 8))
for ax, (name, f) in zip(axes.flat, functions.items()):
    x = np.linspace(-3, 3, 200)
    ax.plot(x, f(x), 'b-', linewidth=2)
    # Tangent at x=1
    x0 = 1.0
    h = 1e-5
    grad = (f(x0 + h) - f(x0 - h)) / (2 * h)
    tangent = f(x0) + grad * (x - x0)
    ax.plot(x, tangent, 'r--', alpha=0.7)
    is_cvx = check_convexity_numerical(f, (-3, 3))
    ax.set_title(f'{name} (convex: {is_cvx})')
    ax.grid(True, alpha=0.3)
plt.tight_layout()
plt.savefig('convex_functions.pdf')
```

2.13 Exercises

Exercise 2.1 ($\star$)**.** Show that $f(x) = \max(x_1, \ldots, x_n)$ is convex on $\mathbb{R}^n$.

Exercise 2.2 ($\star$)**.** Show that if f is convex and g is affine, then $f \circ g$ is convex.

Exercise 2.3 ($\star\star$)**.** Show that a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is m-strongly convex if and only if $\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq m \|x - y\|^2$ for all x, y.

Exercise 2.4 ($\star\star$)**.** Show that $f(x) = \log\left(\sum_{i=1}^n e^{x_i}\right)$ is convex and compute its smoothness constant L with respect to the ℓ_2 norm.

Exercise 2.5 ($\star\star$)**.** Let f be convex, proper, and lower semicontinuous. Show that f is bounded below by an affine function.

Exercise 2.6 ($\star\star\star$)**.** Prove the equivalence of the four characterizations of L-smoothness in Theorem 2.14.

Exercise 2.7 ($\star\star\star$)**.** Let f be convex on $\mathbb{R}^n$ and x^* a minimizer. Show that if f is m-strongly convex:

$$f(x) - f(x^*) \geq \frac{m}{2} \|x - x^*\|^2, \quad \forall x \in \mathbb{R}^n.$$

Chapter 3

Subdifferentials and Subgradients

3.1 Introduction

Here is a concrete problem: you want to minimise $\|x\|_1$ subject to linear constraints—a central problem in sparse learning. But the ℓ_1 norm is not differentiable at the origin: it has a "corner." The gradient does not exist, and yet that is precisely where the minimum lies. How do you write an optimality condition without a gradient?

The answer, developed by Jean-Jacques Moreau and R. Tyrrell Rockafellar in the 1960s, is the *subdifferential*: instead of a single tangent plane, one considers *all* hyperplanes that remain below the graph of the function. At points of differentiability, there is only one (the classical gradient). At corners, there are infinitely many, forming a convex compact set. This generalisation opens the door to a remarkably coherent theory of nonsmooth optimisation.

Intuition

For a differentiable function, the gradient defines the unique tangent hyperplane that minorizes the function. For a nonsmooth convex function, there may exist *multiple* minorizing hyperplanes at a point — their collection forms the subdifferential.

3.2 Definition of Subgradient and Subdifferential

Definition 3.1 (Subgradient)**.** Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function. A vector $g \in \mathbb{R}^n$ is a *subgradient* of f at $x \in \operatorname{dom}(f)$ if:

$$f(y) \geq f(x) + \langle g, y - x \rangle, \quad \forall y \in \mathbb{R}^n.$$

Definition 3.2 (Subdifferential)**.** The *subdifferential* of f at x, denoted $\partial f(x)$, is the set of all subgradients of f at x:

$$\partial f(x) = \{g \in \mathbb{R}^n \mid f(y) \geq f(x) + \langle g, y - x \rangle, \ \forall y \in \mathbb{R}^n\}.$$

If $x \notin \operatorname{dom}(f)$, we set $\partial f(x) = \emptyset$.

Theorem 3.3 (Existence of Subgradients)**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be proper convex. Then $\partial f(x) \neq \emptyset$ for all $x \in \operatorname{ri}(\operatorname{dom}(f))$. In particular, if $\operatorname{dom}(f)$ is open, $\partial f(x) \neq \emptyset$ for all $x \in \operatorname{dom}(f)$.*

Proof. Let $x \in \mathrm{ri}(\mathrm{dom}(f))$. The point $(x, f(x))$ lies on the boundary of $\mathrm{epi}(f)$, which is convex. By the supporting hyperplane theorem, there exists $(g, -\alpha) \neq 0$ such that:

$$\langle g, y - x \rangle - \alpha(t - f(x)) \leq 0, \quad \forall (y, t) \in \mathrm{epi}(f).$$

Taking $y = x$ and $t \to +\infty$, we get $\alpha \geq 0$. If $\alpha = 0$, then $\langle g, y - x \rangle \leq 0$ for all $y \in \mathrm{dom}(f)$, contradicting $x \in \mathrm{ri}(\mathrm{dom}(f))$ (unless $g = 0$). So $\alpha > 0$, and g/α is a subgradient. □

3.3 Properties of the Subdifferential

Theorem 3.4 (Fundamental Properties)**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be proper convex.*

1. *$\partial f(x)$ is a closed convex set (possibly empty).*
2. *If f is differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$.*
3. *$\partial f(x)$ is bounded if $x \in \mathrm{int}(\mathrm{dom}(f))$.*
4. *x^* minimizes f if and only if $0 \in \partial f(x^*)$.*

Proof of (4). ($\Rightarrow$) If x^* minimizes f, then $f(y) \geq f(x^*)$ for all y, i.e., $f(y) \geq f(x^*) + \langle 0, y - x^* \rangle$, so $0 \in \partial f(x^*)$.

($\Leftarrow$) If $0 \in \partial f(x^*)$, then $f(y) \geq f(x^*) + \langle 0, y - x^* \rangle = f(x^*)$ for all y, so x^* is a global minimizer. □

Subdifferential Optimality Condition

$$x^* \in \arg\min_{x \in \mathbb{R}^n} f(x) \quad \Longleftrightarrow \quad 0 \in \partial f(x^*).$$

This is the generalization of $\nabla f(x^*) = 0$ to the nonsmooth case.

3.4 Examples of Subdifferentials

Example 3.5 (Absolute Value)**.** For $f(x) = |x|$ on $\mathbb{R}$:

$$\partial f(x) = \begin{cases} \{-1\} & \text{if } x < 0, \\ [-1, 1] & \text{if } x = 0, \\ \{1\} & \text{if } x > 0. \end{cases}$$

Example 3.6 (ℓ_1 Norm)**.** For $f(x) = \|x\|_1 = \sum_i |x_i|$ on $\mathbb{R}^n$:

$$\partial f(x) = \{g \in \mathbb{R}^n \mid g_i = \mathrm{sign}(x_i) \text{ if } x_i \neq 0,\ g_i \in [-1, 1] \text{ if } x_i = 0\}.$$

Example 3.7 (Maximum of Smooth Functions)**.** For $f(x) = \max(f_1(x), \ldots, f_k(x))$ with f_i differentiable and convex:

$$\partial f(x) = \mathrm{conv}\{\nabla f_i(x) \mid i \in I(x)\},$$

where $I(x) = \{i \mid f_i(x) = f(x)\}$ is the set of active indices.

Example 3.8 (Indicator Function)**.** For $f = \iota_C$ (indicator of a closed convex set C):

$$\partial \iota_C(x) = N_C(x) = \{g \in \mathbb{R}^n \mid \langle g, y - x \rangle \leq 0,\ \forall y \in C\},$$

which is the *normal cone* to C at x.

3.5 Subdifferential Calculus Rules

Theorem 3.9 (Moreau–Rockafellar Sum Rule)**.** *Let $f_1, f_2 : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be proper convex. If* $\operatorname{ri}(\operatorname{dom}(f_1)) \cap \operatorname{ri}(\operatorname{dom}(f_2)) \neq \emptyset$*, then:*

$$\partial(f_1 + f_2)(x) = \partial f_1(x) + \partial f_2(x).$$

Theorem 3.10 (Chain Rule)**.** *Let f be convex and $g(x) = f(Ax + b)$ with $A \in \mathbb{R}^{m \times n}$, provided $Ax + b \in \operatorname{ri}(\operatorname{dom}(f))$. Then:*

$$\partial g(x) = A^T \partial f(Ax + b).$$

Theorem 3.11 (Subdifferential of the Maximum)**.** *Let $f_1, \ldots, f_k$ be convex. For $f(x) = \max_i f_i(x)$:*

$$\partial f(x) = \operatorname{conv}\left(\bigcup_{i \in I(x)} \partial f_i(x) \right),$$

where $I(x) = \{i \mid f_i(x) = f(x)\}$.

Proposition 3.12 (Scaling)**.** For $\alpha > 0$: $\partial(\alpha f)(x) = \alpha \partial f(x)$.

3.6 Monotonicity of the Subdifferential

Theorem 3.13 (Monotonicity)**.** *Let f be convex. The operator ∂f is* monotone*: for all $x, y \in \operatorname{dom}(\partial f)$, $g_x \in \partial f(x)$, $g_y \in \partial f(y)$:*

$$\langle g_x - g_y, x - y \rangle \geq 0.$$

If f is m-strongly convex, ∂f is strongly monotone*:*

$$\langle g_x - g_y, x - y \rangle \geq m \|x - y\|^2.$$

Proof. By definition of subgradients:

$$f(y) \geq f(x) + \langle g_x, y - x \rangle,$$
$$f(x) \geq f(y) + \langle g_y, x - y \rangle.$$

Summing: $0 \geq \langle g_x, y - x \rangle + \langle g_y, x - y \rangle = -\langle g_x - g_y, x - y \rangle$. □

3.7 Subgradient and Descent Direction

Proposition 3.14. Let f be convex and $x \notin \arg\min f$. Then for any $g \in \partial f(x)$ with $g \neq 0$, the direction $d = -g$ is a descent direction:

$$f(x - tg) < f(x)$$

for $t > 0$ sufficiently small (if f is continuous at x).

Subgradient vs Gradient

Unlike the gradient, the direction $-g$ with $g \in \partial f(x)$ is *not* necessarily the steepest descent direction. Moreover, the subgradient method does not guarantee monotone decrease of f.

3.8 Subgradient Method

Subgradient Method

1. Initialize $x_0 \in \mathrm{dom}(f)$, $f_{\text{best}} = +\infty$.
2. For $k = 0, 1, 2, \ldots$:
   (a) Choose $g_k \in \partial f(x_k)$.
   (b) Update: $x_{k+1} = x_k - \alpha_k g_k$.
   (c) $f_{\text{best}} = \min(f_{\text{best}}, f(x_k))$.

Step size choices:

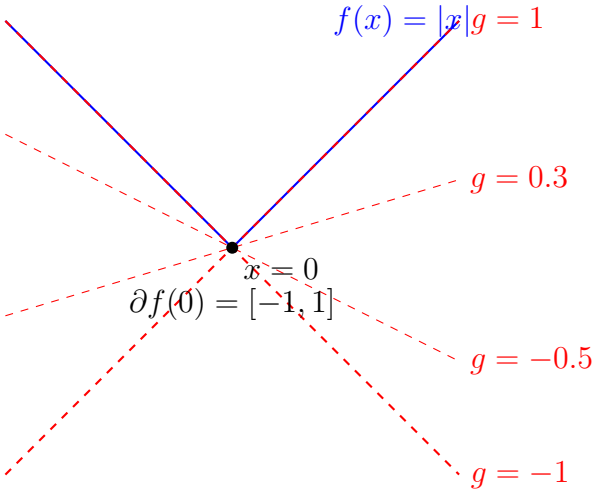
- Constant: $\alpha_k = \alpha$.
- Diminishing: $\alpha_k \to 0$ with $\sum \alpha_k = +\infty$.
- Polyak: $\alpha_k = \frac{f(x_k) - f^*}{\|g_k\|^2}$ (if f^* known).

Theorem 3.15 (Subgradient Convergence)**.** *With step size $\alpha_k = c/\sqrt{k+1}$ and $\|g_k\| \le G$ for all k:*

$$f_{\text{best}}^{(k)} - f^* \le \frac{\|x_0 - x^*\|^2 + c^2 G^2 \sum_{i=0}^{k} 1}{2c\sqrt{k+1}} = O\left(\frac{1}{\sqrt{k}}\right).$$

The optimal rate is $O(1/\sqrt{k})$, much slower than gradient descent for smooth functions.

3.9 Geometric Interpretation



3.10 Python Implementation

Subgradient method for ℓ_1 minimization

```python
import numpy as np
import matplotlib.pyplot as plt

def subgradient_l1(A, b, x0, n_iter=500, c=1.0):
    """Minimize ||Ax - b||_1 via subgradient method."""
    x = x0.copy()
    n = len(x)
    f_best = np.inf
    x_best = x.copy()
    history = []

    for k in range(n_iter):
        r = A @ x - b
        f_val = np.sum(np.abs(r))
        if f_val < f_best:
            f_best = f_val
            x_best = x.copy()
        history.append(f_best)

        # Subgradient of ||Ax - b||_1
        g = A.T @ np.sign(r)
        alpha = c / np.sqrt(k + 1)
        x = x - alpha * g

    return x_best, history

# Test
np.random.seed(42)
m, n = 50, 20
A = np.random.randn(m, n)
x_true = np.zeros(n)
x_true[:5] = np.random.randn(5)
b = A @ x_true + 0.1 * np.random.randn(m)

x0 = np.zeros(n)
x_sol, hist = subgradient_l1(A, b, x0, n_iter=1000)

plt.semilogy(hist)
plt.xlabel('Iteration')
plt.ylabel('Best f value')
plt.title('Subgradient method for L1 minimization')
plt.grid(True, alpha=0.3)
plt.savefig('subgradient_l1.pdf')
```

3.11 Exercises

Exercise 3.1 ($\star$)**.** Compute the subdifferential of $f(x) = \max(0, x)$ at every point $x \in \mathbb{R}$.

Exercise 3.2 ($\star$)**.** Compute $\partial f(x)$ for $f(x) = \|x\|_2$ on $\mathbb{R}^n$.

Exercise 3.3 ($\star\star$)**.** Let $f(x) = \max(x_1, x_2)$ on $\mathbb{R}^2$. Compute $\partial f(x)$ for (x_1, x_2) with $x_1 > x_2$, $x_1 < x_2$, and $x_1 = x_2$.

Exercise 3.4 ($\star\star$)**.** Prove the sum rule: if f and g are convex with $\operatorname{dom}(f) \cap \operatorname{int}(\operatorname{dom}(g)) \neq \emptyset$, then $\partial(f + g)(x) = \partial f(x) + \partial g(x)$.

Exercise 3.5 ($\star\star$)**.** Show that the normal cone $N_C(x) = \partial \iota_C(x)$ is a closed convex cone.

Exercise 3.6 ($\star\star\star$)**.** Show that f is convex if and only if the operator ∂f is monotone.

Exercise 3.7 ($\star\star\star$)**.** Implement the subgradient method with Polyak step size to minimize $f(x) = \|Ax - b\|_1 + \lambda \|x\|_\infty$ and compare with CVXPY.

Chapter 4

Fenchel–Moreau–Rockafellar Duality

The Legendre transform, known to physicists since the eighteenth century for passing from the Lagrangian to the Hamiltonian, was generalised by Werner Fenchel in the 1940s into a tool of remarkable power: *convex conjugation*. The idea is to represent a convex function not by its graph, but by the collection of its supporting hyperplanes—a kind of geometric duality where every convex function has an alter ego. The Fenchel-Moreau biconjugation theorem states that for a lower semicontinuous convex function, applying this transformation twice returns the original function. This involution is the key to duality in convex optimisation.

4.1 Introduction

Convex conjugation, or the Legendre–Fenchel transform, is a fundamental tool of convex analysis. It establishes a correspondence between convex functions and enables the construction of powerful dual problems. The Fenchel–Moreau biconjugation theorem characterizes lower semicontinuous convex functions.

4.2 Fenchel Conjugate

Definition 4.1 (Fenchel Conjugate)**.** Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. The *Fenchel conjugate* (or *Legendre–Fenchel transform*) of f is:

$$f^*(y) = \sup_{x \in \mathbb{R}^n} \{\langle y, x\rangle - f(x)\}.$$

Proposition 4.2 (Immediate Properties)**.**

1. f^* is always convex and lower semicontinuous (as a supremum of affine functions in y).
2. **Fenchel–Young inequality**: $f(x) + f^*(y) \geq \langle x, y\rangle$ for all x, y.
3. If f is proper, then f^* never takes the value $-\infty$.

Proof of the Fenchel–Young inequality. By definition of f^*:

$$f^*(y) = \sup_z \{\langle y, z\rangle - f(z)\} \geq \langle y, x\rangle - f(x),$$

hence $f(x) + f^*(y) \geq \langle x, y\rangle$. □

Fenchel–Young Inequality

$$f(x) + f^*(y) \geq \langle x, y \rangle, \quad \forall x, y \in \mathbb{R}^n,$$

with equality if and only if $y \in \partial f(x)$ (or equivalently $x \in \partial f^*(y)$).

4.3 Examples of Conjugates

Example 4.3 (Classical Conjugates).

1. **Affine function**: $f(x) = \langle a, x \rangle + b$. Then $f^*(y) = \begin{cases} -b & \text{if } y = a, \\ +\infty & \text{otherwise.} \end{cases}$
2. **Quadratic**: $f(x) = \frac{1}{2} x^T Q x$ with $Q \succ 0$. Then $f^*(y) = \frac{1}{2} y^T Q^{-1} y$.
3. **Norm**: $f(x) = \|x\|_p$ for $p \geq 1$. Then $f^* = \iota_{B_q}$ where $B_q = \{y \mid \|y\|_q \leq 1\}$ and $1/p + 1/q = 1$.
4. **Squared norm**: $f(x) = \frac{1}{2} \|x\|^2$. Then $f^*(y) = \frac{1}{2} \|y\|^2$ (self-conjugate).
5. **Indicator**: $f = \iota_C$. Then $f^* = \sigma_C$ (support function).
6. **Negative entropy**: $f(x) = x \log x - x$ for $x > 0$. Then $f^*(y) = e^y$.
7. **Log-sum-exp**: $f(x) = \log(\sum_i e^{x_i})$. Then $f^*(y) = \sum_i y_i \log y_i$ if $y \in \Delta_n$, $+\infty$ otherwise.

Proof for the quadratic. $f^*(y) = \sup_x \{\langle y, x \rangle - \frac{1}{2} x^T Q x\}$. The optimality condition gives $y - Qx = 0$, i.e., $x = Q^{-1} y$. Thus:

$$f^*(y) = \langle y, Q^{-1} y \rangle - \frac{1}{2} (Q^{-1} y)^T Q (Q^{-1} y) = y^T Q^{-1} y - \frac{1}{2} y^T Q^{-1} y = \frac{1}{2} y^T Q^{-1} y.$$

□

4.4 Fenchel–Moreau Biconjugation Theorem

Theorem 4.4 (Fenchel–Moreau). *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be proper. Then:*

$$f^{**} = f \quad \Longleftrightarrow \quad f \text{ is convex and lower semicontinuous.}$$

In general, f^{**} is the greatest convex l.s.c. function majorized by f (the convex l.s.c. regularization *of f, denoted $\overline{\text{conv}}(f)$).*

Proof sketch. First show that $f^{**} \leq f$ (from the Fenchel–Young inequality). Then use the separation theorem: if f is convex l.s.c. and $(x_0, t_0) \notin \text{epi}(f)$, there exists a hyperplane separating (x_0, t_0) from $\text{epi}(f)$, which shows $f^{**}(x_0) \geq t_0$. Since this holds for all $t_0 < f(x_0)$, we get $f^{**} \geq f$. □

4.5 Link Between Conjugate and Subdifferential

Theorem 4.5 (Conjugate–Subdifferential Equivalence)**.** *Let f be proper convex and l.s.c. The following are equivalent:*

1. $y \in \partial f(x)$,
2. $x \in \partial f^*(y)$,
3. $f(x) + f^*(y) = \langle x, y \rangle$.

Proof. (1) $\Rightarrow$ (3): If $y \in \partial f(x)$, then for all z: $f(z) \geq f(x) + \langle y, z - x \rangle$, i.e., $\langle y, z \rangle - f(z) \leq \langle y, x \rangle - f(x)$. So $f^*(y) = \langle y, x \rangle - f(x)$.

(3) $\Rightarrow$ (1): $f^*(y) = \langle y, x \rangle - f(x)$ means the supremum in the definition of f^* is attained at x, so $y \in \partial f(x)$.

(1) $\Leftrightarrow$ (2): By $f^{**} = f$ and symmetry. □

4.6 Fenchel–Rockafellar Duality

Theorem 4.6 (Fenchel–Rockafellar Duality)**.** *Let $f, g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be proper convex and $A \in \mathbb{R}^{m \times n}$. Consider the primal problem:*

$$(P) \quad p^* = \inf_{x \in \mathbb{R}^n} \{f(x) + g(Ax)\}$$

and the dual problem:

$$(D) \quad d^* = \sup_{y \in \mathbb{R}^m} \{-f^*(-A^T y) - g^*(y)\}.$$

Then:

1. *(Weak duality) $d^* \leq p^*$.*
2. *(Strong duality) If $\mathrm{ri}(\mathrm{dom}(f)) \cap A^{-1}(\mathrm{ri}(\mathrm{dom}(g))) \neq \emptyset$ (constraint qualification), then $d^* = p^*$ and the dual supremum is attained.*

Proof of weak duality. For all x and y, by the Fenchel–Young inequality:

$$\begin{aligned} f(x) &\geq \langle -A^T y, x \rangle - f^*(-A^T y) = -\langle y, Ax \rangle - f^*(-A^T y), \\ g(Ax) &\geq \langle y, Ax \rangle - g^*(y). \end{aligned}$$

Summing: $f(x) + g(Ax) \geq -f^*(-A^T y) - g^*(y)$. Taking the infimum over x and supremum over y: $p^* \geq d^*$. □

4.7 Special Case: $A = I$

Corollary 4.7 (Fenchel Duality)**.** *For proper convex f, g with $\mathrm{ri}(\mathrm{dom}(f)) \cap \mathrm{ri}(\mathrm{dom}(g)) \neq \emptyset$:*

$$\inf_x \{f(x) + g(x)\} = \sup_y \{-f^*(-y) - g^*(y)\} = -\inf_y \{f^*(-y) + g^*(y)\}.$$

4.8 Conjugate Calculus Rules

Proposition 4.8 (Calculus Rules).

1. **Separability**: if $f(x) = \sum_i f_i(x_i)$, then $f^*(y) = \sum_i f_i^*(y_i)$.
2. **Scaling**: $(\alpha f)^*(y) = \alpha f^*(y/\alpha)$ for $\alpha > 0$.
3. **Translation**: if $g(x) = f(x - a)$, then $g^*(y) = f^*(y) + \langle y, a \rangle$.
4. **Adding a linear term**: if $g(x) = f(x) + \langle b, x \rangle + c$, then $g^*(y) = f^*(y - b) - c$.
5. **Linear composition**: if $g(x) = f(Ax)$ with A invertible, then $g^*(y) = f^*(A^{-T}y)$.
6. **Infimal convolution**: $(f \square g)^* = f^* + g^*$.

4.9 Application: Dual Formulation of LASSO

The primal LASSO problem is:

$$\min_x \frac{1}{2} \|Ax - b\|^2 + \lambda \|x\|_1 .$$

Writing $f(x) = \lambda \|x\|_1$ and $g(z) = \frac{1}{2}\|z - b\|^2$ with $z = Ax$:

$$f^*(y) = \iota_{\{y \mid \|y\|_\infty \le \lambda\}}(y),$$
$$g^*(w) = \frac{1}{2}\|w\|^2 + \langle w, b \rangle.$$

Fenchel–Rockafellar duality gives:

$$\max_\nu \left\{ -\frac{1}{2}\|b - \nu\|^2 \right\} \quad \text{subject to} \quad \left\|A^T \nu\right\|_\infty \le \lambda.$$

4.10 Python Implementation

Computing conjugates and verifying Fenchel–Young

```python
import numpy as np
import cvxpy as cp

def fenchel_conjugate_numerical(f, y, n):
    """Numerical computation of f*(y) = sup_x {<y,x> - f(x)}."""
    x = cp.Variable(n)
    objective = cp.Maximize(y @ x - f(x))
    prob = cp.Problem(objective)
    prob.solve()
    return prob.value

# Example: f(x) = ||x||^2 / 2
n = 5
```

```
y_test = np.random.randn(n)

f = lambda x: 0.5 * cp.sum_squares(x)
f_star_val = fenchel_conjugate_numerical(f, y_test, n)
f_star_exact = 0.5 * np.sum(y_test**2)

print(f"f*(y) numerical:  {f_star_val:.6f}")
print(f"f*(y) analytical: {f_star_exact:.6f}")

# Verify Fenchel-Young: f(x) + f*(y) >= <x,y>
x_test = np.random.randn(n)
f_x = 0.5 * np.sum(x_test**2)
lhs = f_x + f_star_exact
rhs = np.dot(x_test, y_test)
print(f"f(x)+f*(y) = {lhs:.4f} >= <x,y> = {rhs:.4f}: {lhs >= rhs-1e-10}")

# Fenchel duality for LASSO
m, nn = 50, 20
A = np.random.randn(m, nn)
b = np.random.randn(m)
lam = 1.0

# Primal
x = cp.Variable(nn)
primal = cp.Problem(cp.Minimize(
    0.5 * cp.sum_squares(A @ x - b) + lam * cp.norm1(x)))
primal.solve()
print(f"\nLASSO primal: {primal.value:.6f}")

# Dual
nu = cp.Variable(m)
dual = cp.Problem(cp.Maximize(
    -0.5 * cp.sum_squares(b - nu)),
    [cp.norm_inf(A.T @ nu) <= lam])
dual.solve()
print(f"LASSO dual:   {dual.value + 0.5*np.sum(b**2):.6f}")
```

4.11 Exercises

Exercise 4.1 ($\star$). Compute the conjugate of $f(x) = \frac{1}{p}\|x\|_p^p$ for $p > 1$. Verify that $f^*(y) = \frac{1}{q}\|y\|_q^q$ with $1/p + 1/q = 1$.

Exercise 4.2 ($\star$). Show that if $f(x) = \iota_C(x)$, then $f^* = \sigma_C$.

Exercise 4.3 ($\star\star$). Show that $f^{**} \leq f$ always, and that $f^{**} = f$ if and only if f is convex and lower semicontinuous.

Exercise 4.4 ($\star\star$). Compute the conjugate of $f(x) = \log(1 + e^x)$ and interpret the result in terms of entropy.

Exercise 4.5 ($\star\star$)**.** Derive the dual problem of the LASSO via Fenchel–Rockafellar duality.

Exercise 4.6 ($\star\star\star$)**.** Prove the Fenchel–Moreau theorem using the hyperplane separation theorem.

Exercise 4.7 ($\star\star\star$)**.** Show the rule $(f \,\square\, g)^* = f^* + g^*$ and deduce that the infimal convolution of two proper convex l.s.c. functions is convex l.s.c.

Chapter 5

Optimality Conditions — KKT

In unconstrained optimisation, the optimality condition is simple: the gradient vanishes. But as soon as constraints are added—inequalities, equalities—this naïve condition no longer suffices. The Karush-Kuhn-Tucker (KKT) conditions, discovered independently by William Karush in 1939 (in a master's thesis that remained long ignored) and by Harold Kuhn and Albert Tucker in 1951, generalise the zero-gradient condition by introducing *Lagrange multipliers* for the constraints. In convex optimisation, under very general constraint qualifications, KKT conditions are not only necessary but also *sufficient*—they completely characterise optimal solutions.

5.1 Introduction

The Karush–Kuhn–Tucker (KKT) conditions generalize the classical condition $\nabla f(x^*) = 0$ to constrained optimization problems. In convex optimization, under very general constraint qualifications, the KKT conditions are both necessary and sufficient for optimality.

5.2 Constrained Convex Optimization Problem

Consider the general problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \le 0, \quad i = 1, \dots, m, \\ & h_j(x) = 0, \quad j = 1, \dots, p, \end{aligned} \tag{P}$$

where f and the g_i are convex and the h_j are affine: $h_j(x) = a_j^T x - b_j$.

Definition 5.1 (Feasible Set)**.** The *feasible set* is:

$$\mathcal{F} = \{x \in \mathbb{R}^n \mid g_i(x) \le 0,\ i = 1, \dots, m,\ h_j(x) = 0,\ j = 1, \dots, p\}.$$

Definition 5.2 (Active Constraint)**.** The constraint $g_i(x) \le 0$ is *active* at x if $g_i(x) = 0$. The active index set is $\mathcal{A}(x) = \{i \mid g_i(x) = 0\}$.

5.3 KKT Conditions

Definition 5.3 (KKT Conditions)**.** A point $x^* \in \mathcal{F}$ satisfies the *KKT conditions* if there exist multipliers $\lambda^* \in \mathbb{R}^m$ and $\nu^* \in \mathbb{R}^p$ such that:

1. **Stationarity**: $0 \in \partial f(x^*) + \sum_{i=1}^{m} \lambda_i^* \partial g_i(x^*) + \sum_{j=1}^{p} \nu_j^* a_j$.

2. **Primal feasibility**: $g_i(x^*) \leq 0$ and $h_j(x^*) = 0$.

3. **Dual feasibility**: $\lambda_i^* \geq 0$ for all i.

4. **Complementary slackness**: $\lambda_i^* g_i(x^*) = 0$ for all i.

If f and the g_i are differentiable, the stationarity condition reads:

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{p} \nu_j^* a_j = 0.$$

KKT Conditions (Differentiable Case)

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + A^T \nu^* = 0, \quad \lambda^* \geq 0, \quad \lambda^* \circ g(x^*) = 0, \quad g(x^*) \leq 0, \quad Ax^* = b.$$

Intuition

The complementary slackness condition $\lambda_i^* g_i(x^*) = 0$ means that either the constraint is inactive ($g_i(x^*) < 0$ and $\lambda_i^* = 0$) or it is active ($g_i(x^*) = 0$ and $\lambda_i^* \geq 0$ potentially nonzero). The multiplier λ_i^* measures the "force" with which the constraint pushes the solution.

5.4 Constraint Qualifications

Definition 5.4 (Slater's Condition)**.** *Slater's condition* is satisfied if there exists a strictly feasible point $\hat{x}$ such that:

$$g_i(\hat{x}) < 0, \quad i = 1, \ldots, m, \quad h_j(\hat{x}) = 0, \quad j = 1, \ldots, p.$$

(For affine constraints g_i, it suffices that $g_i(\hat{x}) \leq 0$.)

Theorem 5.5 (KKT — Necessary and Sufficient Conditions)**.** *Consider problem (P) with f, g_i convex and h_j affine.*

1. ***Sufficiency (always)**: if (x^*, λ^*, ν^*) satisfies the KKT conditions, then x^* is an optimal solution of (P).*

2. ***Necessity (under Slater)**: if Slater's condition holds and x^* is optimal, then there exists (λ^*, ν^*) such that (x^*, λ^*, ν^*) satisfies the KKT conditions.*

Proof of sufficiency. Suppose (x^*, λ^*, ν^*) satisfies KKT. For any $x \in \mathcal{F}$:

$$\begin{aligned}
f(x) &\geq f(x^*) + \langle g_f, x - x^* \rangle \quad \text{where } g_f \in \partial f(x^*) \\
&= f(x^*) - \sum_i \lambda_i^* \langle g_{g_i}, x - x^* \rangle - \sum_j \nu_j^* a_j^T (x - x^*) \quad \text{(stationarity)} \\
&\geq f(x^*) - \sum_i \lambda_i^* (g_i(x) - g_i(x^*)) - \sum_j \nu_j^* (h_j(x) - h_j(x^*)) \quad \text{(subgradient of } g_i) \\
&\geq f(x^*) - \sum_i \lambda_i^* g_i(x) + \sum_i \lambda_i^* g_i(x^*) \quad \text{(since } h_j(x) = h_j(x^*) = 0) \\
&\geq f(x^*) \quad \text{(since } \lambda_i^* \geq 0, g_i(x) \leq 0, \lambda_i^* g_i(x^*) = 0).
\end{aligned}$$

□

5.5 Examples of KKT Application

Example 5.6 (Quadratic Programming)**.** Consider:

$$\min_x \frac{1}{2} x^T Q x + c^T x \quad \text{subject to} \quad Ax \leq b.$$

The KKT conditions are:

$$Qx^* + c + A^T \lambda^* = 0, \quad \lambda^* \geq 0, \quad Ax^* \leq b, \quad \lambda^* \circ (Ax^* - b) = 0.$$

Example 5.7 (Projection onto a Convex Set)**.** The projection $\bar{x} = \text{proj}_C(y)$ solves:

$$\min_x \frac{1}{2} \|x - y\|^2 \quad \text{subject to} \quad x \in C.$$

If $C = \{x \mid g_i(x) \leq 0\}$, the KKT conditions give:

$$\bar{x} - y + \sum_i \lambda_i^* \nabla g_i(\bar{x}) = 0, \quad \lambda_i^* \geq 0, \quad \lambda_i^* g_i(\bar{x}) = 0.$$

Example 5.8 (Hard-Margin SVM)**.** The primal SVM:

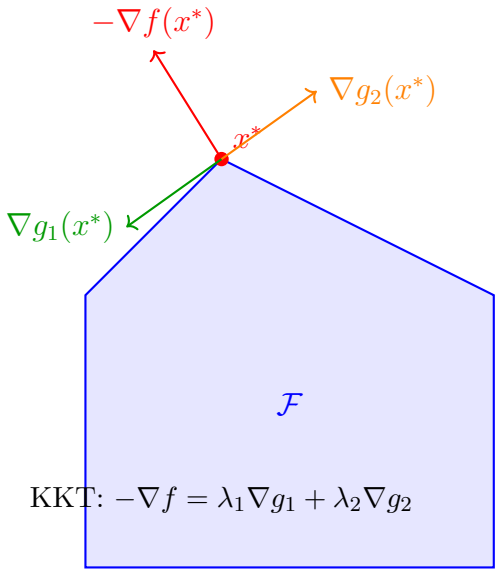
$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad \text{subject to} \quad y_i(\langle w, x_i \rangle + b) \geq 1, \ i = 1, \ldots, N.$$

The KKT conditions:

$$w^* = \sum_i \alpha_i^* y_i x_i, \quad \sum_i \alpha_i^* y_i = 0, \quad \alpha_i^* \geq 0, \quad \alpha_i^* (y_i(\langle w^*, x_i \rangle + b^*) - 1) = 0.$$

Complementary slackness shows that only points with $y_i(\langle w^*, x_i \rangle + b^*) = 1$ (support vectors) have $\alpha_i^* > 0$.

5.6 Geometric Interpretation



5.7 Other Constraint Qualifications

Definition 5.9 (Linear Independence (LICQ))**.** *LICQ* holds at x^* if the gradients of the active constraints $\{\nabla g_i(x^*)\}_{i \in \mathcal{A}(x^*)} \cup \{a_j\}_{j=1}^{p}$ are linearly independent.

Proposition 5.10. Under LICQ, the KKT multipliers are unique.

Definition 5.11 (Mangasarian–Fromovitz (MFCQ))**.** *MFCQ* holds at x^* if the $\{a_j\}$ are linearly independent and there exists $d \in \mathbb{R}^n$ such that:

$$\langle \nabla g_i(x^*), d \rangle < 0, \ \forall i \in \mathcal{A}(x^*), \quad a_j^T d = 0, \ \forall j.$$

5.8 Sensitivity and Economic Interpretation

Theorem 5.12 (Sensitivity)**.** *Let $p^*(u, v)$ be the optimal value of the perturbed problem:*

$$\min f(x) \quad \textit{subject to} \quad g_i(x) \le u_i, \ h_j(x) = v_j.$$

Under regularity conditions, the KKT multipliers satisfy:

$$\lambda_i^* = -\frac{\partial p^*}{\partial u_i}\bigg|_{(0,0)}, \quad \nu_j^* = -\frac{\partial p^*}{\partial v_j}\bigg|_{(0,0)}.$$

Remark 5.13. λ_i^* represents the "shadow price" of the i-th constraint: relaxing the constraint by one unit improves the optimal value by λ_i^*.

5.9 Python Implementation

Verifying KKT conditions with CVXPY

```python
import numpy as np
import cvxpy as cp

# Quadratic programming
n = 5
np.random.seed(42)
Q = np.random.randn(n, n)
Q = Q.T @ Q + 0.1 * np.eye(n)  # PD
c = np.random.randn(n)
A = np.random.randn(3, n)
b = np.ones(3)

x = cp.Variable(n)
constraints = [A @ x <= b]
prob = cp.Problem(cp.Minimize(0.5 * cp.quad_form(x, Q) + c @ x),
                  constraints)
prob.solve()

x_star = x.value
lambda_star = constraints[0].dual_value

print("Solution x*:", x_star)
print("Multipliers lambda*:", lambda_star)

# Verify stationarity
grad_L = Q @ x_star + c + A.T @ lambda_star
print(f"||grad L|| = {np.linalg.norm(grad_L):.2e}")

# Verify complementary slackness
slack = b - A @ x_star
comp = lambda_star * slack
print(f"Complementarity: max|lambda*g| = {np.max(np.abs(comp)):.2e}")

# Verify feasibility
print(f"Primal feasibility: max(Ax-b) = {np.max(A @ x_star - b):.2e}")
print(f"Dual feasibility: min(lambda) = {np.min(lambda_star):.2e}")
```

5.10 Exercises

Exercise 5.1 ($\star$). Solve $\min x_1^2 + x_2^2$ subject to $x_1 + x_2 \geq 1$ using the KKT conditions.

Exercise 5.2 ($\star$). Verify that Slater's condition holds for $\min \|x\|^2$ subject to $\|x\|_\infty \leq 1$.

Exercise 5.3 ($\star\star$). Derive the KKT conditions for the soft-margin SVM problem:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \quad \text{subject to} \quad y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i,\ \xi_i \geq 0.$$

Exercise 5.4 ($\star\star$). Show that for a linear programming problem, the KKT conditions are equivalent to the primal-dual complementarity conditions.

Exercise 5.5 ($\star\star\star$)**.** Prove the sensitivity theorem 5.12 for a linear programming problem.

Exercise 5.6 ($\star\star\star$)**.** Show that under LICQ, the KKT multipliers are unique. Give an example where LICQ fails and multipliers are not unique.

Chapter 6

Lagrangian Duality

Lagrangian duality is one of the most unifying ideas in optimisation. Its principle: relax the constraints by penalising them in the objective via *Lagrange multipliers*, then maximise over these multipliers. The resulting dual problem is always convex (even if the primal is not), provides a guaranteed lower bound, and coincides with the primal under *strong duality* conditions—which are automatically satisfied in convex optimisation as soon as a Slater constraint qualification holds. This theory, heir to the work of Lagrange (1797), Kuhn-Tucker (1951), and Rockafellar (1970), is the bridge between the primal formulation and the KKT conditions.

6.1 Introduction

Lagrangian duality is a fundamental principle that associates to every optimization problem (primal) a dual problem. The dual provides lower bounds on the primal optimal value and, under convexity and qualification conditions, strong duality allows solving the primal via the dual.

6.2 The Lagrangian

Definition 6.1 (Lagrangian)**.** The *Lagrangian* of problem (P) is the function $L : \mathbb{R}^n \times \mathbb{R}^m_+ \times \mathbb{R}^p \to \mathbb{R}$ defined by:

$$L(x, \lambda, \nu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{p} \nu_j h_j(x).$$

The variables $\lambda \in \mathbb{R}^m_+$ and $\nu \in \mathbb{R}^p$ are the *dual variables* or *Lagrange multipliers*.

Proposition 6.2 (The Lagrangian as Relaxation)**.** For all $\lambda \geq 0$ and ν, and all feasible x:

$$L(x, \lambda, \nu) \leq f(x).$$

Indeed, $\lambda_i g_i(x) \leq 0$ and $\nu_j h_j(x) = 0$.

6.3 Lagrange Dual Function

Definition 6.3 (Dual Function)**.** The *Lagrange dual function* is:

$$q(\lambda,\nu) = \inf_{x\in\mathbb{R}^n} L(x,\lambda,\nu) = \inf_x \left\{ f(x) + \sum_i \lambda_i g_i(x) + \sum_j \nu_j h_j(x) \right\}.$$

Theorem 6.4 (Properties of the Dual Function)**.** *1. q is concave (as the infimum of affine functions in (λ,ν)).*

2. ***Weak duality**: $q(\lambda,\nu) \le p^*$ for all $\lambda \ge 0$ and ν, where p^* is the primal optimal value.*

3. *The domain $\operatorname{dom}(q) = \{(\lambda,\nu) \mid q(\lambda,\nu) > -\infty\}$ is convex.*

Proof of weak duality. For any feasible x and any (λ,ν) with $\lambda \ge 0$:

$$q(\lambda,\nu) = \inf_z L(z,\lambda,\nu) \le L(x,\lambda,\nu) = f(x) + \sum_i \underbrace{\lambda_i g_i(x)}_{\le 0} + \sum_j \underbrace{\nu_j h_j(x)}_{=0} \le f(x).$$

Taking the infimum over feasible x: $q(\lambda,\nu) \le p^*$. □

6.4 Dual Problem

Definition 6.5 (Lagrange Dual Problem)**.** The *dual problem* is:

$$(D) \quad d^* = \sup_{\lambda\ge 0,\,\nu} q(\lambda,\nu) = \sup_{\lambda\ge 0,\,\nu} \inf_x L(x,\lambda,\nu).$$

Definition 6.6 (Duality Gap)**.** The *duality gap* is $p^* - d^* \ge 0$. If $p^* = d^*$, we say that *strong duality* holds.

Primal–Dual Relationship

$$d^* = \sup_{\lambda\ge 0,\nu} \inf_x L(x,\lambda,\nu) \le \inf_x \sup_{\lambda\ge 0,\nu} L(x,\lambda,\nu) = p^*.$$

The inequality is the max-min inequality.

6.5 Strong Duality

Theorem 6.7 (Strong Duality — Slater)**.** *If the primal problem is convex (f, g_i convex, h_j affine), p^* is finite, and Slater's condition is satisfied, then:*

1. *Strong duality holds: $p^* = d^*$.*

2. *The dual supremum is attained: there exist optimal (λ^*,ν^*).*

Proof sketch. Consider the set:

$$\mathcal{G} = \{(u,v,t) \in \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R} \mid \exists x : g_i(x) \le u_i, h_j(x) = v_j, f(x) \le t\}.$$

$\mathcal{G}$ is convex (by convexity of f, g_i and affinity of h_j). The point $(0,0,p^*)$ lies on the boundary of $\mathcal{G}$. By the supporting hyperplane theorem, there exists a separating hyperplane, which yields the optimal Lagrange multipliers. □

6.6 Examples of Dual Computation

Example 6.8 (Linear Programming)**.** Primal: $\min c^T x$ subject to $Ax \le b$, $x \ge 0$.
$L(x, \lambda, \mu) = c^T x + \lambda^T (Ax - b) - \mu^T x = (c + A^T \lambda - \mu)^T x - \lambda^T b$.
$q(\lambda, \mu) = \inf_x L = \begin{cases} -\lambda^T b & \text{if } c + A^T \lambda - \mu = 0, \\ -\infty & \text{otherwise.} \end{cases}$
Dual: $\max -b^T \lambda$ subject to $A^T \lambda \le c$, $\lambda \ge 0$.
After standard reformulation: the dual of an LP is an LP.

Example 6.9 (Minimum Norm)**.** Primal: $\min \|x\|^2$ subject to $Ax = b$.
$L(x, \nu) = \|x\|^2 + \nu^T (Ax - b)$.
Optimality condition $2x + A^T \nu = 0$ gives $x = -\frac{1}{2} A^T \nu$.
$q(\nu) = -\frac{1}{4} \nu^T A A^T \nu - \nu^T b$.
Dual: $\max -\frac{1}{4} \nu^T A A^T \nu - \nu^T b$.

Example 6.10 (Maximum Entropy)**.** $\min \sum_i x_i \log x_i$ subject to $Ax = b$, $\mathbf{1}^T x = 1$, $x \ge 0$.
Dual: $\max -\log \left(\sum_i \exp(-a_i^T \nu - \mu) \right)$ subject to dual constraints, where a_i is the i-th column of A.

6.7 Min-Max Interpretation

Theorem 6.11 (Von Neumann–Sion Minimax Theorem)**.** *If $\mathcal{X}$ is compact convex, $\mathcal{Y}$ is convex, and $\Phi(x, y)$ is convex-concave (convex in x, concave in y), then:*

$$\min_{x \in \mathcal{X}} \sup_{y \in \mathcal{Y}} \Phi(x, y) = \sup_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} \Phi(x, y).$$

Remark 6.12. The Lagrangian $L(x, \lambda, \nu)$ is convex in x (sum of convex functions) and affine (hence concave) in (λ, ν). Under Slater's conditions, the minimax theorem applies, yielding strong duality.

6.8 Saddle Point of the Lagrangian

Definition 6.13 (Saddle Point)**.** (x^*, λ^*, ν^*) is a *saddle point* of the Lagrangian if:

$$L(x^*, \lambda, \nu) \le L(x^*, \lambda^*, \nu^*) \le L(x, \lambda^*, \nu^*), \quad \forall x, \ \forall \lambda \ge 0, \nu.$$

Theorem 6.14 (Saddle Point and KKT Equivalence)**.** *Under Slater's conditions, (x^*, λ^*, ν^*) is a saddle point of the Lagrangian if and only if x^* is primal optimal, (λ^*, ν^*) is dual optimal, and strong duality holds.*

6.9 Economic Interpretation

Intuition

Consider the profit maximization problem of a firm under resource constraints:

$$\max_x \ \text{profit}(x) \quad \text{subject to} \quad \text{resource}_i(x) \le b_i.$$

The multiplier λ_i^* represents the *shadow price* of resource i: it is the marginal gain from increasing the availability of resource i by one unit. A competitive market would naturally set this price.

6.10 Python Implementation

Lagrangian duality and duality gap

```
import numpy as np
import cvxpy as cp

# Example: min ||x||^2 s.t. Ax = b
np.random.seed(42)
m, n = 3, 10
A = np.random.randn(m, n)
b = np.random.randn(m)

# Primal
x = cp.Variable(n)
primal = cp.Problem(cp.Minimize(cp.sum_squares(x)), [A @ x == b])
primal.solve()
p_star = primal.value
print(f"Primal value p* = {p_star:.6f}")
print(f"Multipliers: {primal.constraints[0].dual_value}")

# Analytical dual: max -1/4 nu^T A A^T nu - nu^T b
nu = cp.Variable(m)
M = A @ A.T
dual = cp.Problem(cp.Maximize(-0.25 * cp.quad_form(nu, M) - b @ nu))
dual.solve()
d_star = dual.value
print(f"Dual value d*   = {d_star:.6f}")
print(f"Duality gap     = {p_star - d_star:.2e}")

# LP: verify strong duality
c = np.random.randn(n)
A_ineq = np.random.randn(m, n)
b_ineq = np.abs(np.random.randn(m)) + 1

x_lp = cp.Variable(n)
primal_lp = cp.Problem(cp.Minimize(c @ x_lp),
                       [A_ineq @ x_lp <= b_ineq, x_lp >= 0])
primal_lp.solve()

y_lp = cp.Variable(m)
dual_lp = cp.Problem(cp.Maximize(b_ineq @ y_lp),
                     [A_ineq.T @ y_lp + c >= 0, y_lp <= 0])
dual_lp.solve()
```

```
print(f"\nLP primal: {primal_lp.value:.6f}")
print(f"LP dual:   {dual_lp.value:.6f}")
print(f"Gap:       {abs(primal_lp.value - dual_lp.value):.2e}")
```

6.11 Exercises

Exercise 6.1 ($\star$)**.** Write the Lagrange dual of $\min c^T x$ subject to $Ax = b$, $x \geq 0$.

Exercise 6.2 ($\star$)**.** Compute the dual function for $\min \frac{1}{2} x^T Q x + c^T x$ subject to $Ax = b$ with $Q \succ 0$.

Exercise 6.3 ($\star\star$)**.** Show that strong duality holds for linear programming without needing Slater's condition.

Exercise 6.4 ($\star\star$)**.** Derive the dual of the soft-margin SVM problem and show it is a bounded QP.

Exercise 6.5 ($\star\star$)**.** Show that (x^*, λ^*, ν^*) is a saddle point of the Lagrangian if and only if the KKT conditions are satisfied.

Exercise 6.6 ($\star\star\star$)**.** Prove Slater's theorem using the supporting hyperplane theorem applied to the set $\mathcal{G}$ defined in the proof.

Exercise 6.7 ($\star\star\star$)**.** Let $p^*(u)$ be the optimal value of the perturbed problem $\min f(x)$ subject to $g_i(x) \leq u_i$. Show that p^* is convex in u. Deduce the interpretation of multipliers as subgradients of p^*.

Chapter 7

Gradient Descent and Variants

7.1 Introduction

In 1847, Augustin-Louis Cauchy proposed an idea of disarming simplicity: to minimize a function, simply follow the direction opposite to the gradient. Like a ball rolling on a surface, one descends the steepest slope, step by step, until reaching the bottom of the valley. This method, *gradient descent*, long remained a theoretical tool. But the explosion of machine learning in the twenty-first century propelled it to the rank of most-executed algorithm in the world: every time a neural network learns, it is a variant of gradient descent that adjusts its billions of parameters. The fundamental questions — which step size to choose? how to accelerate convergence? what to do when the gradient is too expensive to compute exactly? — have led to Nesterov's method, Adam, and stochastic gradient descent, which together form the indispensable toolkit of modern optimization.

7.2 Fixed Step-Size Gradient Descent

Gradient Descent

1. Choose $x_0 \in \mathbb{R}^n$, step size $\alpha > 0$.
2. For $k = 0, 1, 2, \ldots$:
$$x_{k+1} = x_k - \alpha \nabla f(x_k).$$
3. Stop when $\|\nabla f(x_k)\| < \varepsilon$.

Intuition

At each iteration, we move in the direction opposite to the gradient (the direction of steepest local descent) with step size α. The gradient points in the direction of steepest ascent, so $-\nabla f(x_k)$ is the steepest descent direction.

7.3 Convergence Analysis — L-Smooth Functions

Theorem 7.1 (Convergence for Convex L-Smooth Functions)**.** *Let f be convex and L-smooth. With step size $\alpha = 1/L$:*

$$f(x_k) - f^* \leq \frac{L \|x_0 - x^*\|^2}{2k}.$$

The convergence rate is $O(1/k)$.

Proof. By L-smoothness, with $\alpha = 1/L$:

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2} \|x_{k+1} - x_k\|^2 .$$

With $x_{k+1} - x_k = -\frac{1}{L}\nabla f(x_k)$:

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L} \|\nabla f(x_k)\|^2 .$$

By the first-order condition: $f(x_k) - f^* \leq \langle \nabla f(x_k), x_k - x^* \rangle \leq \|\nabla f(x_k)\| \, \|x_k - x^*\|$, so:

$$\|\nabla f(x_k)\|^2 \geq \frac{(f(x_k) - f^*)^2}{\|x_k - x^*\|^2}.$$

Moreover, $\|x_{k+1} - x^*\|^2 = \|x_k - x^*\|^2 - \frac{2}{L}\langle \nabla f(x_k), x_k - x^* \rangle + \frac{1}{L^2} \|\nabla f(x_k)\|^2$.

Setting $\delta_k = f(x_k) - f^*$ and $R_k = \|x_k - x^*\|^2$:

$$\delta_{k+1} \leq \delta_k - \frac{1}{2L} \|\nabla f(x_k)\|^2 , \quad R_{k+1} \leq R_k - \frac{2}{L}\delta_k + \frac{1}{L^2} \|\nabla f(x_k)\|^2 .$$

Summing the sufficient decrease: $\sum_{i=0}^{k-1} \frac{1}{2L} \|\nabla f(x_i)\|^2 \leq \delta_0$, and by a telescoping argument on R_k:

$$\sum_{i=0}^{k-1} \delta_i \leq \frac{L}{2} R_0.$$

Since δ_k is decreasing: $k\delta_k \leq \sum_{i=0}^{k-1} \delta_i \leq \frac{L}{2} R_0$. □

7.4 Convergence for Strongly Convex Functions

Theorem 7.2 (Linear Convergence)**.** *Let f be m-strongly convex and L-smooth. With $\alpha = 1/L$:*

$$f(x_k) - f^* \leq \left(1 - \frac{m}{L}\right)^k (f(x_0) - f^*).$$

The rate is linear (geometric convergence) with factor $1 - 1/\kappa$ where $\kappa = L/m$ is the condition number.

Proof. By L-smoothness and step $\alpha = 1/L$: $f(x_{k+1}) \leq f(x_k) - \frac{1}{2L} \|\nabla f(x_k)\|^2$.

By strong convexity: $\|\nabla f(x_k)\|^2 \geq 2m(f(x_k) - f^*)$ (Polyak–Łojasiewicz inequality). Thus:

$$f(x_{k+1}) - f^* \leq (f(x_k) - f^*) - \frac{m}{L}(f(x_k) - f^*) = \left(1 - \frac{m}{L}\right)(f(x_k) - f^*).$$

□

Convergence Rate Summary

Assumption	Rate	Complexity for ε
Convex, L-smooth	$O(1/k)$	$O(L/\varepsilon)$
m-str. convex, L-smooth	$O((1-m/L)^k)$	$O(\kappa \log(1/\varepsilon))$
Convex, nonsmooth	$O(1/\sqrt{k})$	$O(1/\varepsilon^2)$

7.5 Line Search

Definition 7.3 (Armijo Condition (Backtracking))**.** Choose α_k as the largest $\alpha = \beta^j \hat{\alpha}$ $(j = 0, 1, \ldots)$ satisfying:

$$f(x_k - \alpha \nabla f(x_k)) \leq f(x_k) - c\alpha \left\|\nabla f(x_k)\right\|^2,$$

with $c \in (0, 1/2)$ and $\beta \in (0, 1)$.

Proposition 7.4. Gradient descent with Armijo backtracking preserves the same convergence rates $O(1/k)$ and $O((1-m/L)^k)$ (with different constants).

7.6 Nesterov's Accelerated Gradient

Nesterov's Accelerated Gradient (NAG)

1. Initialize $x_0 = y_0 \in \mathbb{R}^n$.
2. For $k = 0, 1, 2, \ldots$:

$$x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k),$$
$$y_{k+1} = x_{k+1} + \frac{k}{k+3}(x_{k+1} - x_k).$$

Theorem 7.5 (Nesterov's Convergence)**.** *Let f be convex and L-smooth. Nesterov's algorithm satisfies:*

$$f(x_k) - f^* \leq \frac{2L\left\|x_0 - x^*\right\|^2}{(k+1)^2}.$$

The rate $O(1/k^2)$ is optimal among first-order methods (Nemirovski–Yudin lower bound).

Optimality of the $O(1/k^2)$ Rate

Nemirovski and Yudin (1983) showed that no first-order method (using only values of f and ∇f) can converge faster than $O(1/k^2)$ for the class of convex L-smooth functions. Nesterov's acceleration achieves this bound: it is *optimal*.

7.7 Conjugate Gradient

Nonlinear Conjugate Gradient (Fletcher–Reeves)

1. $d_0 = -\nabla f(x_0)$.
2. For $k = 0, 1, 2, \ldots$:
   (a) $\alpha_k = \arg\min_{\alpha > 0} f(x_k + \alpha d_k)$.
   (b) $x_{k+1} = x_k + \alpha_k d_k$.
   (c) $\beta_{k+1} = \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}$.
   (d) $d_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1} d_k$.

Remark 7.6. For a quadratic function $f(x) = \frac{1}{2} x^T A x - b^T x$ with $A \succ 0$, conjugate gradient converges in at most n iterations (finite convergence).

7.8 Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent

For $f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$:

1. For $k = 0, 1, 2, \ldots$:
   (a) Sample i_k uniformly from $\{1, \ldots, N\}$.
   (b) $x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k)$.

Theorem 7.7 (SGD Convergence)**.** *For convex, L-smooth f, with $\mathbb{E}[\|\nabla f_i(x) - \nabla f(x)\|^2] \leq \sigma^2$ and $\alpha_k = c/\sqrt{k}$:*

$$\mathbb{E}[f(\bar{x}_k)] - f^* = O\left(\frac{1}{\sqrt{k}}\right),$$

where $\bar{x}_k = \frac{1}{k} \sum_{i=1}^{k} x_i$. For m-strongly convex f with $\alpha_k = 1/(mk)$:

$$\mathbb{E}[f(\bar{x}_k)] - f^* = O\left(\frac{1}{k}\right).$$

7.9 Modern SGD Variants

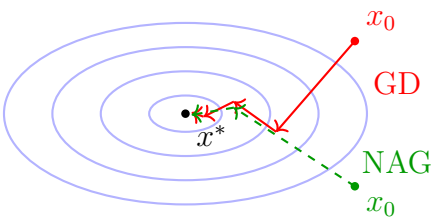
Adam (Kingma & Ba, 2015)

1. Parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, α.
2. $m_0 = 0$, $v_0 = 0$.

3. For $k = 1, 2, \ldots$:

$$
\begin{aligned}
g_k &= \nabla f_{i_k}(x_k), \\
m_k &= \beta_1 m_{k-1} + (1-\beta_1) g_k, \\
v_k &= \beta_2 v_{k-1} + (1-\beta_2) g_k^2, \\
\hat{m}_k &= m_k / (1-\beta_1^k), \\
\hat{v}_k &= v_k / (1-\beta_2^k), \\
x_{k+1} &= x_k - \alpha \hat{m}_k / (\sqrt{\hat{v}_k} + \varepsilon).
\end{aligned}
$$

7.10 Geometric Interpretation



7.11 Python Implementation

Comparison of GD, Nesterov, and SGD

```python
import numpy as np
import matplotlib.pyplot as plt

def gradient_descent(f, grad_f, x0, L, n_iter):
    x = x0.copy()
    history = [f(x)]
    for k in range(n_iter):
        x = x - (1/L) * grad_f(x)
        history.append(f(x))
    return x, history

def nesterov_accelerated(f, grad_f, x0, L, n_iter):
    x = x0.copy()
    y = x0.copy()
    history = [f(x)]
    for k in range(n_iter):
        x_new = y - (1/L) * grad_f(y)
        y = x_new + (k / (k + 3)) * (x_new - x)
        x = x_new
        history.append(f(x))
    return x, history

# Ill-conditioned quadratic
```

```python
np.random.seed(42)
n = 50
eigvals = np.logspace(-2, 2, n)  # kappa = 10000
Q_diag = eigvals
b = np.random.randn(n)

f = lambda x: 0.5 * np.sum(Q_diag * x**2) - np.sum(b * x)
grad_f = lambda x: Q_diag * x - b
L = np.max(Q_diag)
f_star = -0.5 * np.sum(b**2 / Q_diag)

x0 = np.zeros(n)
_, hist_gd = gradient_descent(f, grad_f, x0, L, 500)
_, hist_nag = nesterov_accelerated(f, grad_f, x0, L, 500)

plt.semilogy(np.array(hist_gd) - f_star, label='GD')
plt.semilogy(np.array(hist_nag) - f_star, label='Nesterov')
plt.xlabel('Iteration k')
plt.ylabel('$f(x_k) - f^*$')
plt.legend()
plt.title('GD vs Nesterov (kappa=10000)')
plt.grid(True, alpha=0.3)
plt.savefig('gd_vs_nesterov.pdf')
```

7.12 Exercises

Exercise 7.1 ($\star$)**.** Show that for $f(x) = \frac{1}{2}\|Ax - b\|^2$, the gradient is $\nabla f(x) = A^T(Ax - b)$ and the smoothness constant is $L = \|A^T A\|$.

Exercise 7.2 ($\star\star$)**.** Prove the Polyak–Łojasiewicz inequality: if f is m-strongly convex, then $\|\nabla f(x)\|^2 \geq 2m(f(x) - f^*)$.

Exercise 7.3 ($\star\star$)**.** Implement Armijo backtracking and compare with fixed step $1/L$ on a logistic regression problem.

Exercise 7.4 ($\star\star$)**.** Show that conjugate gradient converges in at most n iterations for a quadratic in dimension n.

Exercise 7.5 ($\star\star\star$)**.** Prove the $O(1/k^2)$ convergence bound of Nesterov's accelerated gradient using a Lyapunov function.

Exercise 7.6 ($\star\star\star$)**.** Show the Nemirovski–Yudin lower bound: there exists a convex L-smooth function such that any first-order method satisfies $f(x_k) - f^* \geq \frac{cL\|x_0 - x^*\|^2}{k^2}$ for $k \leq (n-1)/2$.

Chapter 8

Proximal Methods

8.1 Introduction

In the 1960s, Jean-Jacques Moreau, a mathematician at Montpellier, introduced the *proximal operator*: for a convex function f, the proximal of x is the point that minimizes f plus a quadratic penalty around x. The idea seems innocuous, but it reveals considerable power: where gradient descent requires differentiability, the proximal operator works for any convex lower semicontinuous function, even nonsmooth ones. This observation opened the way to *proximal methods*, which dominate applied optimization today: ISTA and FISTA for the LASSO in statistics, ADMM for distributed optimization, the Douglas–Rachford algorithm for feasibility. Each exploits the decomposition of a complex problem into simple proximal subproblems.

Intuition

The proximal operator generalizes projection onto a convex set. Where gradient descent takes a step and then projects, the proximal operator combines both into a single step: it finds a point close to the argument that also minimizes the function. It is a trade-off between staying near the current point and decreasing the objective.

8.2 Proximal operator

Definition 8.1 (Proximal operator)**.** Let $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be convex, lower semicontinuous, and proper. The **proximal operator** of g is:

$$\operatorname{prox}_g(v) = \arg\min_{x \in \mathbb{R}^n} \left\{ g(x) + \frac{1}{2} \|x - v\|^2 \right\}.$$

Proposition 8.2 (Existence and uniqueness)**.** For any convex, l.s.c., proper function g, $\operatorname{prox}_g(v)$ exists and is unique for every $v \in \mathbb{R}^n$.

Theorem 8.3 (Characterization)**.** $x^* = \operatorname{prox}_g(v)$ *if and only if:*

$$v - x^* \in \partial g(x^*),$$

where ∂g denotes the subdifferential of g.

Proof. The optimality condition for $\min_x g(x) + \frac{1}{2}\|x - v\|^2$ is: $0 \in \partial g(x^*) + (x^* - v)$, i.e., $v - x^* \in \partial g(x^*)$. □

Common proximal operators

- $g = 0$: $\text{prox}_g(v) = v$.
- $g = \iota_C$ (indicator of convex set C): $\text{prox}_g(v) = \Pi_C(v)$ (projection).
- $g = \lambda \|\cdot\|_1$ (LASSO): $[\text{prox}_g(v)]_i = \text{sign}(v_i)\max(|v_i| - \lambda, 0)$ (soft thresholding).
- $g = \frac{\lambda}{2}\|\cdot\|^2$: $\text{prox}_g(v) = \frac{v}{1+\lambda}$.
- $g = \lambda \|\cdot\|_2$ (group LASSO): $\text{prox}_g(v) = v \max\left(1 - \frac{\lambda}{\|v\|}, 0\right)$.

Definition 8.4 (Moreau decomposition)**. Moreau's identity** relates the proximal operator of g to that of its conjugate g^*:

$$\text{prox}_g(v) + \text{prox}_{g^*}(v) = v.$$

8.3 Proximal gradient method

Definition 8.5 (Composite problem)**.** Consider the problem:

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + g(x),$$

where f is convex differentiable (∇f is L-Lipschitz) and g is convex, l.s.c., proper (possibly nondifferentiable).

Proximal Gradient (ISTA)

1. Choose x_0, step size $\gamma \in (0, 1/L]$.
2. For $k = 0, 1, 2, \ldots$:
$$x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k)).$$

Theorem 8.6 (ISTA convergence)**.** *With $\gamma = 1/L$, ISTA satisfies:*

$$F(x_k) - F(x^*) \leq \frac{L\|x_0 - x^*\|^2}{2k}.$$

The convergence rate is $\mathcal{O}(1/k)$.

8.4 Nesterov acceleration: FISTA

FISTA (Fast ISTA)

1. Choose $x_0 = y_0$, $t_0 = 1$, $\gamma = 1/L$.

2. For $k = 0, 1, 2, \ldots$:

$$\begin{aligned} x_{k+1} &= \operatorname{prox}_{\gamma g}(y_k - \gamma \nabla f(y_k)), \\ t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \\ y_{k+1} &= x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k). \end{aligned}$$

Theorem 8.7 (FISTA convergence)**.** *FISTA satisfies:*

$$F(x_k) - F(x^*) \leq \frac{2L \|x_0 - x^*\|^2}{(k+1)^2}.$$

*The $\mathcal{O}(1/k^2)$ rate is **optimal** among first-order methods.*

FISTA is not monotone

Unlike ISTA, the sequence $F(x_k)$ generated by FISTA is not necessarily nonincreasing. The extrapolation step may temporarily increase the objective.

8.5 ADMM

Definition 8.8 (ADMM)**.** The **Alternating Direction Method of Multipliers** solves:

$$\min_{x,z} f(x) + g(z) \quad \text{s.t.} \quad Ax + Bz = c.$$

The iterations are:

$$\begin{aligned} x_{k+1} &= \arg\min_x \left\{ f(x) + \frac{\rho}{2} \|Ax + Bz_k - c + u_k\|^2 \right\}, \\ z_{k+1} &= \arg\min_z \left\{ g(z) + \frac{\rho}{2} \|Ax_{k+1} + Bz - c + u_k\|^2 \right\}, \\ u_{k+1} &= u_k + Ax_{k+1} + Bz_{k+1} - c, \end{aligned}$$

where u is the scaled dual variable and $\rho > 0$.

Theorem 8.9 (ADMM convergence)**.** *Under convexity assumptions on f and g (proper, closed, convex) and feasibility:*

- ***Primal residual**: $Ax_k + Bz_k - c \to 0$.*
- ***Dual residual**: $\rho A^\top B(z_k - z_{k-1}) \to 0$.*
- ***Objective**: $f(x_k) + g(z_k) \to p^*$.*

Remark 8.10. ADMM is particularly effective when the x- and z-subproblems admit closed-form solutions (e.g., least squares + soft thresholding for LASSO).

8.6 Douglas-Rachford splitting

Definition 8.11 (Douglas-Rachford)**.** To solve $\min_x f(x) + g(x)$, the Douglas-Rachford algorithm iterates:

$$\begin{aligned} \hat{x}_k &= \operatorname{prox}_f(z_k), \\ z_{k+1} &= z_k + \operatorname{prox}_g(2\hat{x}_k - z_k) - \hat{x}_k. \end{aligned}$$

Proposition 8.12 (Connection to ADMM)**.** ADMM applied to $\min f(x)+g(z)$ s.t. $x = z$ is equivalent to Douglas-Rachford applied to the dual formulation.

8.7 Applications in signal processing

Example 8.13 (LASSO – sparse regression)**.** The LASSO problem $\min_x \frac{1}{2}\|Ax-b\|^2 + \lambda\|x\|_1$ is composite with $f(x) = \frac{1}{2}\|Ax-b\|^2$ (smooth, $L = \|A^\top A\|$) and $g(x) = \lambda\|x\|_1$. ISTA gives:

$$x_{k+1} = S_{\lambda/L}\left(x_k - \frac{1}{L}A^\top(Ax_k - b)\right),$$

where S_τ is component-wise soft thresholding.

Example 8.14 (Total variation denoising)**.** For signal denoising of $y \in \mathbb{R}^n$:

$$\min_x \frac{1}{2}\|x-y\|^2 + \lambda\|Dx\|_1,$$

where D is the finite difference matrix. ADMM decomposes this into a quadratic subproblem and soft thresholding.

8.8 Exercises

Exercise 8.1 ($\star$ – Proximal operators)**.** Compute the proximal operator of $g(x) = \lambda\|x\|_1$ and of $g(x) = \iota_{[0,+\infty)^n}(x)$ (indicator of the nonnegative orthant).

Exercise 8.2 ($\star\star$ – ISTA for LASSO)**.** Implement ISTA for LASSO with $A \in \mathbb{R}^{50\times 200}$, $x^* \in \mathbb{R}^{200}$ sparse (10 nonzero entries), and $b = Ax^* + \epsilon$. Plot convergence and compare with FISTA.

Exercise 8.3 ($\star\star$ – Moreau identity)**.** Prove Moreau's identity: $\operatorname{prox}_g(v) + \operatorname{prox}_{g^*}(v) = v$. Deduce $\operatorname{prox}_{\lambda\|\cdot\|_\infty}$.

Exercise 8.4 ($\star\star\star$ – ADMM for LASSO)**.** Derive the ADMM iterations for LASSO. Show that the x-subproblem is a linear system and the z-subproblem is soft thresholding. Implement and compare with FISTA.

Exercise 8.5 ($\star\star\star$ – Accelerated convergence)**.** Show that the FISTA sequence t_k satisfies $t_k \geq (k+1)/2$ and deduce the $\mathcal{O}(1/k^2)$ convergence rate.

Chapter 9

Interior Point Methods

9.1 Introduction

In 1984, Narendra Karmarkar, an engineer at AT&T Bell Labs, published an algorithm that shook the optimization world: a polynomial-time linear programming method that, unlike Dantzig's simplex, does not traverse the vertices of the polyhedron but passes through its *interior*. The community was electrified — and divided. Some saw a revolution; others dismissed it as hype (the simplex, though exponential in the worst case, is fast in practice). But history would decide: interior point methods, refined by Nesterov and Nemirovski in the 1990s, proved decisive for semidefinite optimization, conic optimization, and large-scale problems. The idea is elegant: replace inequality constraints with a logarithmic barrier, then follow the *central path* toward the optimum.

Intuition

Imagine you are inside a room (the feasible polyhedron) searching for the lowest point (the optimum). The simplex walks along the walls and corners. Interior point methods walk through the middle of the room, gradually approaching the optimal solution (often near a wall). An invisible barrier prevents you from touching the walls, but this barrier weakens over iterations.

9.2 Barrier method

9.2.1 Formulation

Definition 9.1 (Inequality-constrained problem)**.** Consider the convex problem:

$$\min_{x\in\mathbb{R}^n} f_0(x) \quad \text{s.t.} \quad f_i(x) \le 0,\ i = 1,\dots,m, \qquad Ax = b,$$

where $f_0, f_1, \dots, f_m$ are convex and twice differentiable.

Definition 9.2 (Logarithmic barrier function)**.** The **logarithmic barrier function** is:

$$\phi(x) = -\sum_{i=1}^{m} \ln(-f_i(x)),$$

defined on the strict interior $\{x : f_i(x) < 0,\ \forall i\}$.

Definition 9.3 (Barrier problem)**.** The **barrier problem** for parameter $t > 0$ is:

$$\min_x \; t\, f_0(x) + \phi(x) \quad \text{s.t.} \quad Ax = b.$$

Its solution $x^*(t)$ is called the **central point** for parameter t.

9.2.2 Central path

Definition 9.4 (Central path)**.** The **central path** is the set $\{x^*(t) : t > 0\}$. As $t \to +\infty$, $x^*(t) \to x^*$, the solution of the original problem.

Theorem 9.5 (Suboptimality bound)**.** *For any $t > 0$, the central point $x^*(t)$ satisfies:*

$$f_0(x^*(t)) - p^* \leq \frac{m}{t},$$

where p^* is the optimal value and m is the number of inequality constraints.

Proof. The KKT conditions of the barrier problem show that $x^*(t)$ is feasible for the original problem, with multipliers $\lambda_i^*(t) = -1/(t\, f_i(x^*(t)))$. By weak duality:

$$f_0(x^*(t)) - p^* \leq \sum_{i=1}^{m} \lambda_i^*(t)(-f_i(x^*(t))) = \sum_{i=1}^{m} \frac{1}{t} = \frac{m}{t}.$$

□

Barrier Method

1. Choose strictly feasible x_0, $t_0 > 0$, factor $\mu > 1$, tolerance $\varepsilon > 0$.
2. For $k = 0, 1, 2, \ldots$:
   (a) **Centering**: solve (via Newton) $\min_x t_k f_0(x) + \phi(x)$ s.t. $Ax = b$, starting from x_k, to obtain $x_{k+1} = x^*(t_k)$.
   (b) **Stopping**: if $m/t_k < \varepsilon$, **stop**.
   (c) **Update**: $t_{k+1} = \mu \cdot t_k$.

Barrier method complexity

- Number of outer iterations (barrier steps): $\lceil \frac{\ln(m/(t_0\varepsilon))}{\ln \mu} \rceil$.
- Typical choice: $\mu = 10$ to 20, yielding few outer iterations.
- Each centering step requires a few Newton steps (6–40 in practice).

9.3 Short-step and long-step methods

Definition 9.6 (Short-step method)**.** The **short-step** method chooses μ close to 1 (e.g., $\mu = 1 + 1/\sqrt{m}$) and performs a single Newton step per centering phase. The iteration count is $\mathcal{O}(\sqrt{m} \ln(m/\varepsilon))$.

Definition 9.7 (Long-step method)**.** The **long-step** method chooses a large μ and performs multiple Newton steps to re-center. The total Newton step count is also $\mathcal{O}(\sqrt{m}\ln(m/\varepsilon))$ but with a larger constant.

Theorem 9.8 (Polynomial complexity)**.** *Interior point methods solve a linear program with n variables and m constraints in $\mathcal{O}(\sqrt{m}\ln(1/\varepsilon))$ Newton steps, each costing $\mathcal{O}(n^3)$ (or less when exploiting sparsity). The total complexity is polynomial.*

9.4 Primal-dual interior point method

Definition 9.9 (Modified KKT system)**.** The **primal-dual** method simultaneously solves the perturbed KKT conditions:

$$\begin{cases} \nabla f_0(x) + \sum_i \lambda_i \nabla f_i(x) + A^\top \nu = 0, \\ -\lambda_i f_i(x) = 1/t, \quad i = 1, \dots, m, \\ Ax = b. \end{cases}$$

Newton's method is applied to this system.

Remark 9.10. The primal-dual method is more efficient in practice than the pure barrier method because it does not require solving the centering problem exactly at each stage. Modern solvers (MOSEK, CVXOPT) use this approach.

Theorem 9.11 (Primal-dual convergence)**.** *The primal-dual method achieves ε-accuracy in $\mathcal{O}(\sqrt{m}\ln(1/\varepsilon))$ Newton iterations under standard regularity assumptions.*

9.5 Application to linear programming

Example 9.12 (LP in standard form)**.** For the LP $\min\ c^\top x$ s.t. $Ax = b$, $x \geq 0$, the barrier function is:

$$\phi(x) = -\sum_{j=1}^{n} \ln(x_j).$$

The Newton system for centering is:

$$\begin{pmatrix} X^{-2} & A^\top \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \nu \end{pmatrix} = - \begin{pmatrix} t\,c - X^{-1}e \\ 0 \end{pmatrix},$$

where $X = \operatorname{diag}(x)$ and $e = (1, \dots, 1)^\top$.

Example 9.13 (Semidefinite programming)**.** Interior point methods extend naturally to **semidefinite programs** (SDPs), where the constraint $x \geq 0$ is replaced by $X \succeq 0$ (positive semidefinite matrix). The barrier function becomes $\phi(X) = -\ln\det(X)$.

9.6 Exercises

Exercise 9.1 ($\star$ – Logarithmic barrier)**.** For the LP $\min -x_1 - x_2$ s.t. $x_1 + x_2 \leq 1$, $x_1, x_2 \geq 0$, write the barrier function and compute $x^*(t)$ for $t = 1, 10, 100$. Verify that $x^*(t) \to x^*$.

Exercise 9.2 ($\star\star$ – Central path)**.** Plot the central path for $\min x_1$ s.t. $x_1^2 + x_2^2 \leq 1$ for $t \in \{0.1, 1, 10, 100\}$. Show it converges to $(-1, 0)$.

Exercise 9.3 ($\star\star$ – Newton step)**.** Derive the Newton system for the barrier method applied to the constrained QP: $\min \frac{1}{2} x^\top Q x + c^\top x$ s.t. $Ax \leq b$.

Exercise 9.4 ($\star\star\star$ – Suboptimality bound)**.** Prove in detail that $f_0(x^*(t)) - p^* \leq m/t$. Deduce the number of outer iterations needed to achieve accuracy ε with factor μ.

Exercise 9.5 ($\star\star\star$ – Primal-dual implementation)**.** Implement the primal-dual interior point method for LP in standard form. Test on a problem with 50 variables and 20 constraints. Compare iteration counts with the pure barrier method.

Chapter 10

Applications

10.1 Introduction

This chapter illustrates the power of convex optimization on concrete problems from signal processing, finance, and machine learning. Each application is formulated as a convex optimization problem and solved with the tools from previous chapters.

Intuition

Convex optimization is not an abstract theory: it is a universal language for formulating and efficiently solving problems across many domains. The key is to *recognize* the hidden convex structure in an applied problem, then apply the appropriate algorithm.

10.2 Compressed sensing

Definition 10.1 (Compressed sensing)**. Compressed sensing** aims to recover a sparse signal $x^* \in \mathbb{R}^n$ from $m \ll n$ linear measurements $b = Ax^* + \epsilon$, where $A \in \mathbb{R}^{m \times n}$.

Theorem 10.2 (ℓ_1 recovery)**.** *If x^* is s-sparse and A satisfies the **restricted isometry property** (RIP) of order $2s$ with constant $\delta_{2s} < \sqrt{2} - 1$, then the solution of:*

$$\min_x \|x\|_1 \quad s.t. \quad \|Ax - b\|_2 \leq \epsilon$$

satisfies $\|\hat{x} - x^*\|_2 \leq C\epsilon$ for a constant C.

Remark 10.3. In practice, the equivalent LASSO formulation is often preferred:

$$\min_x \frac{1}{2} \|Ax - b\|^2 + \lambda \|x\|_1 ,$$

which is a composite problem solvable by ISTA/FISTA (Chapter 8).

Example 10.4 (Sparse signal recovery)**.** Let $x^* \in \mathbb{R}^{500}$ with 20 nonzero entries, $A \in \mathbb{R}^{100 \times 500}$ Gaussian, and $b = Ax^*$. LASSO with $\lambda = 0.1$ recovers x^* exactly, whereas (underdetermined) least squares fails.

10.3 LASSO and ℓ_1 regularization

Definition 10.5 (LASSO)**.** The **LASSO** (Least Absolute Shrinkage and Selection Operator):

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y - X\beta\|^2 + \lambda \|\beta\|_1$$

simultaneously performs estimation and variable selection in linear regression.

Proposition 10.6 (LASSO properties)**.**

1. The ℓ_1 penalty produces **sparse** solutions (many coefficients exactly zero).
2. The parameter λ controls sparsity: $\lambda \to +\infty$ gives $\hat{\beta} = 0$; $\lambda \to 0$ gives ordinary least squares.
3. The regularization path $\lambda \mapsto \hat{\beta}(\lambda)$ is piecewise linear.

Theorem 10.7 (LASSO optimality conditions)**.** *$\hat{\beta}$ solves LASSO if and only if:*

$$\frac{1}{n} X^\top (X\hat{\beta} - y) + \lambda v = 0,$$

where $v \in \partial \left\| \hat{\beta} \right\|_1$, i.e., $v_j = \operatorname{sign}(\hat{\beta}_j)$ if $\hat{\beta}_j \neq 0$ and $|v_j| \leq 1$ otherwise.

10.4 Portfolio optimization

Definition 10.8 (Markowitz model)**.** The **Markowitz model** seeks the minimum-variance portfolio for a given expected return:

$$\min_{w} w^\top \Sigma w \quad \text{s.t.} \quad \mu^\top w \geq r_{\min}, \quad \mathbf{1}^\top w = 1, \quad w \geq 0,$$

where $w \in \mathbb{R}^n$ is the weight vector, Σ the covariance matrix, and μ the expected returns vector.

Remark 10.9. This is a convex **quadratic program** (QP) since Σ is positive semidefinite. It can be solved by interior point methods or QP-specialized simplex.

Proposition 10.10 (Efficient frontier)**.** The set of optimal portfolios (as $r_{\min}$ varies) forms the **efficient frontier** in the (standard deviation, return) plane. It is a convex curve.

Example 10.11 (3-asset portfolio)**.** With $\mu = (0.12, 0.10, 0.07)^\top$ and

$$\Sigma = \begin{pmatrix} 0.04 & 0.006 & 0.002 \\ 0.006 & 0.025 & 0.004 \\ 0.002 & 0.004 & 0.01 \end{pmatrix},$$

the minimum variance portfolio (no return constraint) is $w^* \approx (0.14, 0.28, 0.58)^\top$ with $\sigma^* \approx 8.1\%$.

10.5 Optimal transport

Definition 10.12 (Kantorovich problem)**.** Given two discrete probability measures $\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$, the **Kantorovich optimal transport** problem is:

$$\min_{\Pi \geq 0} \sum_{i,j} C_{ij} \Pi_{ij} \quad \text{s.t.} \quad \Pi \mathbf{1} = a, \quad \Pi^\top \mathbf{1} = b,$$

where $C_{ij} = c(x_i, y_j)$ is the transport cost.

Remark 10.13. This is a linear program. The associated Wasserstein distance defines a metric on the space of probability measures, with applications in generative models, computer vision, and statistics.

Definition 10.14 (Entropic regularization)**. Sinkhorn's** entropic regularization adds an entropy term:

$$\min_{\Pi \geq 0} \sum_{i,j} C_{ij} \Pi_{ij} + \varepsilon \sum_{i,j} \Pi_{ij} \ln \Pi_{ij} \quad \text{s.t.} \quad \Pi \mathbf{1} = a,\ \Pi^\top \mathbf{1} = b.$$

Sinkhorn's algorithm solves this via alternating matrix-vector products. For discrete measures with n points, the complexity per iteration is $\mathcal{O}(n^2)$, and the number of iterations to reach precision ε is $\mathcal{O}(1/\varepsilon)$ (Altschuler et al., 2017).

10.6 Machine learning applications

10.6.1 SVM dual

Definition 10.15 (SVM – dual formulation)**.** The linear **Support Vector Machine** (SVM) dual is:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^\top x_j \quad \text{s.t.} \quad 0 \leq \alpha_i \leq C, \quad \sum_i \alpha_i y_i = 0,$$

where $(x_i, y_i) \in \mathbb{R}^d \times \{-1, +1\}$ are the data and $C > 0$. This is a convex QP.

Proposition 10.16 (Support vectors)**.** The **support vectors** are the data points x_i for which $\alpha_i^* > 0$. The separating hyperplane is $w^* = \sum_i \alpha_i^* y_i x_i$.

10.6.2 Logistic regression

Definition 10.17 (Regularized logistic regression)**.** Logistic regression with ℓ_2 regularization:

$$\min_{\beta} \sum_{i=1}^{n} \ln(1 + e^{-y_i x_i^\top \beta}) + \frac{\lambda}{2} \|\beta\|^2 .$$

The objective is strongly convex ($\lambda > 0$) and smooth. It is efficiently solved by Newton, L-BFGS, or proximal gradient (with ℓ_1 regularization for sparsity).

Theorem 10.18 (Convexity of the logistic loss)**.** *The logistic loss $\ell(\beta) = \ln(1 + e^{-yx^\top \beta})$ is convex in β for every (x, y). With ℓ_2 regularization, the problem is λ-strongly convex, guaranteeing uniqueness of the solution.*

10.7 Python implementation

LASSO and portfolio with CVXPY

```python
import numpy as np
import cvxpy as cp

# --- LASSO ---
n, p = 100, 200
X = np.random.randn(n, p)
beta_true = np.zeros(p)
beta_true[:10] = np.random.randn(10)
y = X @ beta_true + 0.1 * np.random.randn(n)

beta = cp.Variable(p)
lam = 0.1
prob = cp.Problem(cp.Minimize(
    0.5 * cp.sum_squares(X @ beta - y)
    + lam * cp.norm1(beta)))
prob.solve()
print("LASSO: nnz =", np.sum(np.abs(beta.value) > 1e-4))

# --- Markowitz Portfolio ---
mu = np.array([0.12, 0.10, 0.07])
Sigma = np.array([[0.04, 0.006, 0.002],
                  [0.006, 0.025, 0.004],
                  [0.002, 0.004, 0.01]])
w = cp.Variable(3)
ret = mu @ w
risk = cp.quad_form(w, Sigma)
prob2 = cp.Problem(cp.Minimize(risk),
    [ret >= 0.09, cp.sum(w) == 1, w >= 0])
prob2.solve()
print("Portfolio:", np.round(w.value, 3))
```

10.8 Exercises

Exercise 10.1 ($\star$ – LASSO)**.** For $X \in \mathbb{R}^{20\times 50}$ and $y = X\beta^* + \epsilon$ with β^* 5-sparse, solve LASSO for $\lambda \in \{0.01, 0.1, 1\}$. Plot the number of nonzero components as a function of λ.

Exercise 10.2 ($\star\star$ – Portfolio)**.** Solve the Markowitz problem for 5 assets with historical data. Plot the efficient frontier for $r_{\min}$ ranging from 5% to 15%.

Exercise 10.3 ($\star\star$ – SVM)**.** Formulate and solve the SVM dual for a linearly separable 2D dataset. Identify the support vectors.

Exercise 10.4 ($\star\star\star$ – Optimal transport)**.** Implement Sinkhorn's algorithm for two discrete distributions on $\mathbb{R}^2$ (50 points each). Plot the optimal transport plan and study the influence of ε.

Exercise 10.5 ($\star\star\star$ – Compressed sensing)**.** Generate $A \in \mathbb{R}^{m\times n}$ Gaussian, x^* s-sparse, and $b = Ax^*$. Study the phase transition experimentally: for $n = 200$, vary m and s and determine when ℓ_1 recovery succeeds.

Exercise 10.6 ($\star\star$ – RIP constant properties)**.** Let $A \in \mathbb{R}^{m\times n}$ and δ_s be the restricted isometry property (RIP) constant of order s, defined as the smallest $\delta \geq 0$ such that:

$$(1-\delta)\ \|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta)\ \|x\|_2^2$$

for all s-sparse vectors x.

1. Show that $\delta_s \leq \delta_{s'}$ for $s \leq s'$ (monotonicity).
2. Show that if $\delta_{2s} < 1$, then A is injective on the set of s-sparse vectors.
3. Let A be an $m \times n$ Gaussian matrix with i.i.d. $\mathcal{N}(0, 1/m)$ entries. Using a concentration bound on the norm over column subsets, show that $\delta_s < \delta$ with high probability whenever $m \geq C\,\delta^{-2}\,s\,\ln(n/s)$.

Exercise 10.7 ($\star\star\star$ – LASSO regularization path)**.** Consider the LASSO problem: $\hat{\beta}(\lambda) = \arg\min_\beta \frac{1}{2n}\|y - X\beta\|^2 + \lambda\|\beta\|_1$.

1. Show that for $\lambda \geq \lambda_{\max} = \frac{1}{n}\left\|X^\top y\right\|_\infty$, the solution is $\hat{\beta}(\lambda) = 0$.
2. Using the KKT conditions, show that the active set $\mathcal{A}(\lambda) = \{j : \hat{\beta}_j(\lambda) \neq 0\}$ is piecewise constant in λ and that changes occur at finitely many values $\lambda_1 > \lambda_2 > \cdots > 0$.
3. On each interval $(\lambda_{k+1}, \lambda_k)$, show that $\hat{\beta}_{\mathcal{A}}(\lambda)$ is an affine function of λ:

   $$\hat{\beta}_{\mathcal{A}}(\lambda) = (X_{\mathcal{A}}^\top X_{\mathcal{A}})^{-1}(X_{\mathcal{A}}^\top y - n\lambda\, s_{\mathcal{A}})$$

   where $s_{\mathcal{A}} = \operatorname{sign}(\hat{\beta}_{\mathcal{A}})$. Conclude that the path $\lambda \mapsto \hat{\beta}(\lambda)$ is piecewise linear.

Bibliography

[1] Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge, 2004.

[2] Rockafellar, R.T., *Convex Analysis*, Princeton, 1970.

[3] Bauschke, H.H. and Combettes, P.L., *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd ed., Springer, 2017.

[4] Nesterov, Y., *Introductory Lectures on Convex Optimization*, Springer, 2004.