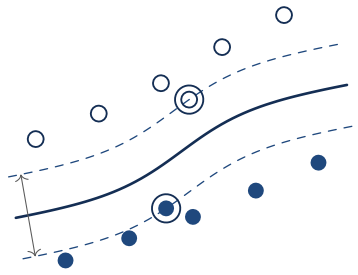
# Mathematical Foundations of Machine Learning

*From Vectors and Probability to Support Vector Machines*



**Taka Khoo**

Thayer School of Engineering
Dartmouth College

*Mathematical Foundations of Machine Learning: From Vectors and Probability to Support Vector Machines.* First edition.



This book is based on the lectures of Professor Peter Chin for ENGS 96, *Mathematical Foundations of Machine Learning*, taught at the Thayer School of Engineering, Dartmouth College, across the summer and fall terms of 2025. It reorganizes and expands the lecture notes, problem sets, coding assignments, and examinations of that course. The mathematical development follows Deisenroth, Faisal, and Ong's *Mathematics for Machine Learning*, the text assigned for the course, with Bishop's *Pattern Recognition and Machine Learning* and Boyd and Vandenberghe's *Convex Optimization*. Worked examples, problem statements, and solutions originate in the course materials; where an original solution contained an error, the corrected value is given and the discrepancy noted in the text.

Self-published by the author in Hanover, New Hampshire.

**How to cite this book.** Khoo, T. (2026). *Mathematical Foundations of Machine Learning: From Vectors and Probability to Support Vector Machines* (1st ed.). Self-published. To cite a single chapter, add its number, for example: Khoo, T. (2026), *Mathematical Foundations of Machine Learning* (1st ed.), Chapter 10.

Written, typeset, and illustrated by Taka Khoo in Latin Modern with LaTeX; the diagrams are drawn with TikZ. Made as a gift for Professor Peter Chin, and as a record of two terms' work.

*Version of June 8, 2026.*

*For Professor Peter Chin,*

*who started us at a single vector,*
*and never once asked us to take a result on faith.*

# Preface

This book grew out of two terms of ENGS 96, *Mathematical Foundations for Machine Learning*, taught by Professor Peter Chin at Dartmouth across its summer and fall offerings. Over those terms the course accumulated a large body of material: lecture notes, problem sets, coding assignments, practice tests, and final reviews. Each piece was written for a single moment in a single term. This book gathers all of it into one place, fills in the steps that a live lecture can leave implicit, and arranges the result the way a textbook should be arranged.

## What this book is

The goal is completeness without compromise. Every topic Professor Chin covered appears here. Every worked example is carried through to its final answer. Every variable is defined where it first appears. When a derivation in the original notes skipped from one line to the next, the intermediate algebra has been restored so that nothing has to be taken on faith. The intent is that you can open to any page, point at any equation, and trace it backward to a definition without leaving the book.

The organization follows the mathematics rather than the calendar. A term runs ten weeks, and a lecture is a unit of time rather than a unit of thought. So the chapters are built around ideas that belong together, and a single chapter often draws on several lectures from both offerings. The six parts mirror the arc Professor Chin laid out at the start: linear algebra and matrix methods, then matrix decompositions, then probability, then concentration and learning theory, then vector calculus and optimization, and finally the machine learning algorithms that all of this machinery was built to explain.

## How to use it

Each chapter opens with a short roadmap and a suggested reading in the official course text, Deisenroth et al. [5]. The body develops the theory with full proofs and runs detailed worked examples in the **Example** environments. Definitions, theorems, and lemmas share a single numbering stream per chapter, so Theorem 3.4 sits right after Definition 3.3 and you never have to guess which counter you are reading.

Every chapter ends with a set of **Exercises**, each followed by a complete solution. These are not decorative. They are the actual problems the course asked, restated in full and solved in full, line by line, together with a few supplementary problems that round out a topic the course only gestured at. Nothing is left as "it can be shown."

The appendices collect the reference material you reach for repeatedly: a linear algebra refresher, a probability and distributions sheet, the matrix calculus identities used in the optimization chapters, the software setup for the coding assignments, and a complete exam bank with solutions.

## A note on notation

Vectors are bold lowercase ($\boldsymbol{x}$, $\boldsymbol{\theta}$), matrices are uppercase italic ($A$, $X$, with bold uppercase greek $\boldsymbol{\Sigma}$ for matrices that are naturally greek), and scalars are lowercase italic. Transpose is written $A^{\top}$, the inverse $A^{-1}$, and the Moore–Penrose pseudoinverse $A^{\dagger}$. The full table of symbols follows the table of contents. The conventions are consistent across every chapter, which is one of the things a pile of separate lecture notes cannot give you and a book can.

## Acknowledgments

The mathematics, the examples, the exam problems, and the way the whole subject hangs together belong to Professor Peter Chin. My part was to write it all down carefully and leave nothing out. Thank you for the class, and for the chance to learn it twice.

*Taka Khoo*
Hanover, New Hampshire

# How to Use This Book

This is a book you can read front to back, or open to any single result and find it self-contained. This short section explains how it is laid out, how the pieces depend on one another, and how to find any statement by its number.

## The shape of the book

The material is organized into six parts.

**Part I, Linear Algebra and Matrix Methods.**
The objects everything is built from: vectors and matrices as linear maps, the three views of matrix multiplication, rank, the determinant, and the least-squares solution of an overdetermined system through projection and the pseudoinverse.

**Part II, Matrix Decompositions.**
The two factorizations the rest of the book leans on: the eigendecomposition and spectral theorem for symmetric matrices, and the singular value decomposition, with its best low-rank approximation theorem.

**Part III, Probability and Distributions.**
Reasoning under uncertainty: random variables, maximum likelihood, Bayes' theorem, the frequentist and Bayesian viewpoints, conjugate priors, and the multivariate Gaussian with its covariance geometry.

**Part IV, Concentration and Learning Theory.**
Why learning from finite data works: Markov, Chebyshev, and Chernoff bounds, empirical risk minimization, the bias–variance tradeoff, and the VC dimension.

**Part V, Vector Calculus and Optimization.**
How models are trained: matrix calculus and backpropagation, convexity and the Hessian, gradient descent and its modern variants, and constrained optimization through linear programming, duality, and the KKT conditions.

**Part VI, Machine Learning Applications.**
The algorithms all of this was built to explain: support vector machines and kernels in supervised learning, and clustering, mixtures, principal component analysis, and independent component analysis in unsupervised learning.

The appendices collect the linear-algebra, probability, and matrix-calculus facts used in the main text, a guide to the software and coding assignments, and a complete bank of examinations worked in full.

## Three kinds of problems

Almost every question in this subject is one of three kinds, and it pays to classify a problem in the first ten seconds.

1. **Compute** something concrete: a pseudoinverse, an eigenvalue, a singular value decomposition, a posterior distribution, a gradient-descent step, a Chernoff bound, the dual of a support vector machine. These reward careful arithmetic. Show every step.
2. **Prove** an identity or inequality: that $A^\top A$ is positive semidefinite, the Cauchy–Schwarz inequality, the spectral theorem, weak duality, that a zero gradient of a convex function is a global minimum. These reward structure. State what you assume, state what you want, then go by construction or contradiction.
3. **Explain** a modeling choice: why the cross-entropy loss rather than squared error, why a Bayesian prior, why the RBF kernel, why the singular value decomposition rather than the eigendecomposition. These reward one clear sentence and one supporting fact.

The worked examples in every chapter and the exercises that close each chapter are written with these three categories in mind.

## Conventions

- **Numbering.** Definitions, theorems, lemmas, propositions, corollaries, remarks, and examples share a single counter within each chapter. So Definition 3.3 and Theorem 3.4 are consecutive, and you can find any result by its number alone. Equations, figures, and exercises are numbered separately, also by chapter.
- **Cross-references** are live links in the digital version. "See Chapter 4" and similar pointers jump to the exact place.
- **Colored boxes are rare and meaningful.** A box with a dark navy border holds the single canonical statement of a major result. A green box is a step-by-step recipe. A red left-rule flags a subtlety or a common mistake. Everything else is ordinary prose and displayed mathematics.
- **Vectors and matrices.** Vectors are bold lowercase ($\boldsymbol{x}$, $\boldsymbol{\theta}$); matrices are uppercase ($A$, $X$), with bold uppercase greek ($\boldsymbol{\Sigma}$) for matrices that are naturally greek; scalars are lowercase italic. Transpose is $A^\top$, inverse $A^{-1}$, pseudoinverse $A^\dagger$. The full symbol table follows this section.

## A map of the book

The chapters follow themes rather than the lecture calendar: a single chapter often gathers several lectures, and the order is the logical one. The table records the core topics of each chapter and the corresponding reading in the course text, Deisenroth et al. [5], so any statement can be placed in the larger arc.

| Chapter | Core topics | Reading |
|---|---|---|
| 1. Vectors, Matrices, Linear Maps | spans, rank, determinant, the linear model | Ch. 2–3 |
| 2. Least Squares and the Pseudoinverse | normal equations, projection, Moore–Penrose | Ch. 3, 4.1 |
| 3. Eigenvalues and the Spectral Theorem | diagonalization, symmetric matrices, PSD | 4.2–4.4 |
| 4. The Singular Value Decomposition | SVD, low rank, Eckart–Young | 4.5 |
| 5. Probability, Bayes, and Estimation | MLE, Bayes, Beta–Bernoulli, Monty Hall | 6.1–6.3 |
| 6. Distributions, Gaussian, Covariance | multinomial, Dirichlet, Gaussian, covariance | 6.4–6.6 |
| 7. Concentration and Learning Theory | Chernoff, ERM, bias–variance, VC dimension | handout |
| 8. Vector Calculus and Gradient Descent | backprop, convexity, GD, Adam | Ch. 5 |
| 9. Constrained Optimization and Duality | LP, simplex, duality, KKT | Ch. 7 |
| 10. Supervised Learning: SVMs and Kernels | perceptron, max margin, the kernel trick | Ch. 12 |
| 11. Unsupervised Learning | $k$-means, mixtures/EM, PCA, ICA | Ch. 10–11 |

## Dependencies and reading paths

The book is cumulative, and the shortest path through its spine is Chapter 1 → Chapter 2 → Chapter 3 → Chapter 4: these four chapters build the linear algebra everything else uses. The probability chapters Chapter 5 and Chapter 6 can be read independently of the matrix decompositions, but the Gaussian's covariance geometry in Chapter 6 draws on the spectral theorem of Chapter 3. The optimization chapters Chapter 8 and Chapter 9 are the prerequisite for the support vector machine of Chapter 10, whose dual is derived with the KKT machinery developed there. Principal component analysis in Chapter 11 returns to the spectral theorem and the SVD, closing the loop with where the book began.

# Contents

# List of Figures

# List of Tables

# Notation

The conventions below are used consistently throughout the book. Vectors are bold lowercase, matrices are uppercase, and scalars are lowercase italic.

## Sets and numbers

| | |
|---|---|
| $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$ | real numbers; real $n$-vectors; real $m \times n$ matrices |
| $\mathbb{C}$, $\mathbb{Z}$, $\mathbb{N}$ | complex numbers, integers, natural numbers |
| $\{\cdot\}$, $\{x \mid P(x)\}$ | a set; the set of $x$ such that $P(x)$ holds |
| $\lvert S\rvert$ | cardinality of a finite set $S$ |
| $[n]$ | the index set $\{1, 2, \dots, n\}$ |

## Linear algebra

| | |
|---|---|
| $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{\theta}$ | vectors (bold lowercase) |
| $A$, $X$, $\boldsymbol{\Sigma}$ | matrices (uppercase; bold greek for greek matrices) |
| $x_i$, $a_{ij}$ | the $i$th entry of $\boldsymbol{x}$; the $(i,j)$ entry of $A$ |
| $\boldsymbol{a}_j$ | the $j$th column of $A$ |
| $A^\top$, $\boldsymbol{x}^\top$ | transpose |
| $A^{-1}$ | inverse (when it exists) |
| $A^\dagger$ | Moore–Penrose pseudoinverse |
| $I$, $I_n$ | identity matrix (of size $n$) |
| $\mathbf{0}$, $\mathbf{1}$ | the all-zeros and all-ones vectors |
| $\boldsymbol{e}_i$ | the $i$th standard basis vector |
| $\operatorname{tr}(A)$ | trace, $\sum_i a_{ii}$ |
| $\operatorname{rank}(A)$ | rank |
| $\det(A)$, $\lvert A\rvert$ | determinant |
| $\operatorname{diag}(\boldsymbol{d})$ | diagonal matrix with diagonal $\boldsymbol{d}$ |
| $\operatorname{Col}(A)$, $\operatorname{Row}(A)$, $\operatorname{Nul}(A)$ | column space, row space, null space |
| $\operatorname{span}\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_k\}$ | span (set of all linear combinations) |
| $\lVert\boldsymbol{x}\rVert$, $\lVert\boldsymbol{x}\rVert_2$, $\lVert\boldsymbol{x}\rVert_1$ | Euclidean ($\ell_2$) norm; $\ell_1$ norm |
| $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \boldsymbol{x}^\top\boldsymbol{y}$ | inner (dot) product |
| $\boldsymbol{u}\boldsymbol{v}^\top$ | outer product (a rank-one matrix) |
| $A \succeq 0$, $A \succ 0$ | $A$ is positive semidefinite; positive definite |

## Probability and statistics

| | |
|---|---|
| $\mathbb{P}(A)$ | probability of event $A$ |
| $p(x),\ p(x \mid y)$ | density/mass of $x$; conditional density of $x$ given $y$ |
| $\mathbb{E}[X],\ \mathrm{Var}(X)$ | expectation; variance |
| $\mathrm{Cov}(X, Y),\ \boldsymbol{\Sigma}$ | covariance; covariance matrix |
| $X \sim \mathcal{N}(\mu, \sigma^2)$ | $X$ is distributed as |
| $X \overset{\text{iid}}{\sim} p$ | independent and identically distributed as $p$ |
| $X \perp\!\!\!\perp Y$ | $X$ and $Y$ are independent |
| $\mathrm{Bernoulli}(\mu),\ \mathrm{Binomial}(n, \mu)$ | Bernoulli, Binomial |
| $\mathrm{Categorical}(\boldsymbol{\pi})$ | Categorical distribution |
| $\mathrm{Multinomial}(n, \boldsymbol{\pi})$ | Multinomial distribution |
| $\mathrm{Beta}(a, b),\ \mathrm{Dirichlet}(\boldsymbol{\alpha})$ | Beta, Dirichlet |
| $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ | (multivariate) Gaussian / normal |
| $\mathrm{H}(p),\ \mathrm{KL}(p \,\|\, q)$ | entropy; Kullback–Leibler divergence |
| $\hat{\theta}_{\mathrm{MLE}},\ \hat{\theta}_{\mathrm{MAP}}$ | maximum likelihood, maximum a posteriori estimates |

## Calculus and optimization

| | |
|---|---|
| $\frac{\partial f}{\partial x}$ | partial derivative |
| $\nabla f,\ \nabla_{\boldsymbol{\theta}} f$ | gradient (with respect to $\boldsymbol{\theta}$) |
| $\nabla^2 f,\ \boldsymbol{J}$ | Hessian; Jacobian |
| $\arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$ | a minimizer of $f$ |
| $f \in \mathcal{O}(g)$ | $f$ grows no faster than $g$ |
| $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ | Lagrangian |
| $\mathrm{dom}\, f,\ \mathrm{epi}\, f$ | domain; epigraph |
| $:=$ | "is defined to equal" |

## Machine learning

| | |
|---|---|
| $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ | training set of $m$ labeled examples |
| $X \in \mathbb{R}^{m \times n}$ | design (data) matrix: one example per row |
| $\boldsymbol{\theta},\ \boldsymbol{w},\ b$ | model parameters; weight vector; bias |
| $\mathcal{H}$ | hypothesis class |
| $\boldsymbol{\phi}(\boldsymbol{x})$ | feature map |
| $k(\boldsymbol{x}, \boldsymbol{x}'),\ K$ | kernel function; Gram (kernel) matrix |
| $\sigma(z) = \frac{1}{1+e^{-z}}$ | logistic sigmoid |
| $\mathbf{1}[\,\cdot\,]$ | indicator (1 if true, 0 otherwise) |

PART I

# Linear Algebra and Matrix Methods

CHAPTER 1

# Vectors, Matrices, and Linear Maps

*Data become matrices, models become linear maps, and learning becomes the solution of a large linear system.*

*a recurring refrain of the course*

**Roadmap**

Machine learning turns data into matrices, models into linear maps, and learning into the solution of linear systems. This chapter builds that vocabulary from the ground up and works every piece on concrete numbers. We start with vectors and how a datum becomes one, add the inner product and norm that let us measure length and angle, and then read a matrix as a *function* rather than a grid. Three complementary views of matrix multiplication follow, then rank and the column space, which decide what a linear system can and cannot do, and finally the determinant, invertibility and its numerical fragility, and the trace. We close by setting up the supervised-learning model $X\boldsymbol{\theta} \approx \boldsymbol{y}$ and seeing precisely why we cannot simply invert $X$, which is the question Chapter 2 answers.

**Suggested reading:** Deisenroth et al. [5], Chapters 2 (linear algebra) and 3 (analytic geometry); Strang [21] for geometric intuition.

## 1.1 Why this mathematics

A modern learning system is, underneath the engineering, a pipeline of linear algebra and calculus. It is worth seeing, before any formalism, why that is true, because every abstraction in this book was invented to make some learning task tractable. Two stories from the opening of the course make the point, and a third idea, that data must first become numbers, sets the agenda for the chapter.

*Intuition.* A neural network looks nothing like a matrix factorization, and a coin flip looks nothing like an eigenvalue. Yet under the hood the network multiplies matrices, the coin's bias is estimated by projecting onto a subspace, and the network's infinite-width limit is governed by the spectrum of a kernel matrix. The objects of this chapter are the common substrate. Learn them once and the rest of the subject becomes variations on a theme.

In the *infinite-width limit*, a neural network with one hidden layer whose width grows without bound stops behaving like a mysterious black box and becomes a *kernel machine*, a predictor of the form

$$f(\boldsymbol{x}) \;=\; \sum_{i=1}^{m} \alpha_i \, k(\boldsymbol{x}, \boldsymbol{x}^{(i)}), \tag{1.1}$$

where $k(\boldsymbol{x}, \boldsymbol{x}')$ measures similarity between inputs and the weights $\alpha_i$ are learned. Every symbol in Eq. (1.1) is a vector, a number, or a function of two vectors. We meet kernels properly in Chapter 10; for now the lesson is that even the most elaborate models rest on the objects of this chapter.

The second story is *ImageNet*. Early computer vision tried to describe objects by hand ("a cat has a round face and pointed ears") and broke on the endless variety of real images. The breakthrough was data: roughly fourteen million labeled images across tens of thousands of categories, enough to train convolutional networks with hundreds

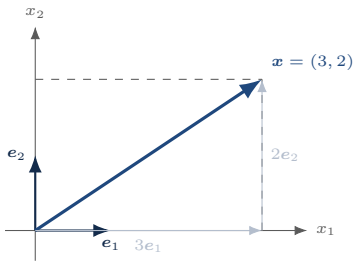



**Figure 1.1.** A vector is a point and an arrow. The coordinates of $\boldsymbol{x} = (3, 2)$ are the amounts of the standard basis vectors needed to build it: $\boldsymbol{x} = 3\boldsymbol{e}_1 + 2\boldsymbol{e}_2$. The entries of a vector are always coordinates *with respect to a basis*.

of millions of parameters. To a computer each of those images is, first of all, a vector of numbers. Before any learning can happen, the data must be put into a form the mathematics can touch. That form is the matrix, and that is where we begin.

### 1.1.1 The supervised learning setup

Throughout the book a *supervised* dataset is a collection of input–output pairs

$$\mathcal{D} \;=\; \left\{ (\boldsymbol{x}^{(i)}, y^{(i)}) \right\}_{i=1}^{m}, \qquad \boldsymbol{x}^{(i)} \in \mathbb{R}^n, \quad y^{(i)} \in \mathbb{R}, \tag{1.2}$$

where $m$ is the number of examples and $n$ is the number of *features* describing each example. For instance, a person might be described by

$$\boldsymbol{x}^{(i)} \;=\; \begin{bmatrix} \text{height} \\ \text{weight} \\ \text{age} \end{bmatrix} \in \mathbb{R}^3.$$

The superscript $(i)$ indexes *which example*; a subscript indexes *which feature*, so $x_j^{(i)}$ is feature $j$ of example $i$. When labels are absent the data is *unsupervised*, $\mathcal{D}_U = \{\boldsymbol{x}^{(i)}\}_{i=1}^{m}$, and the goal shifts from prediction to finding structure (Chapter 11). With this notation in hand, the rest of the chapter assembles the algebra that operates on $\mathcal{D}$.

## 1.2 Vectors

**Definition 1.1 (Vector).** A *(column) vector* $\boldsymbol{x} \in \mathbb{R}^n$ is an ordered list of $n$ real numbers, written

$$\boldsymbol{x} \;=\; \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad x_i \in \mathbb{R}.$$

The number $x_i$ is the $i$th *component* or *entry*, and $n$ is the *dimension*. Vectors are written in bold lowercase and their entries in plain italic.

A vector is two things at once, and fluency means holding both pictures in mind. It is a *point* in $\mathbb{R}^n$, the location with coordinates $(x_1, \dots, x_n)$, and it is the *arrow* from the origin to that point. The arrow picture makes the two operations on vectors intuitive: addition $(\boldsymbol{x} + \boldsymbol{y})_i = x_i + y_i$ places the arrows tip to tail, and scalar multiplication $(\alpha\boldsymbol{x})_i = \alpha x_i$ stretches an arrow by the factor $\alpha$ (reversing it if $\alpha < 0$). Both operations stay inside $\mathbb{R}^n$, and together with the obvious algebraic laws (associativity, distributivity, a zero vector $\mathbf{0}$, negatives) they make $\mathbb{R}^n$ a *vector space* in the sense of Deisenroth et al. [5, Ch. 2].

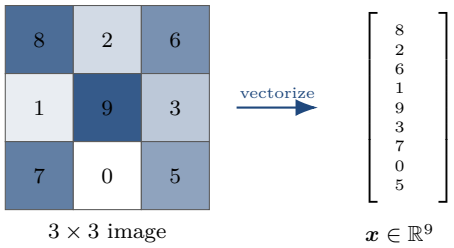



**Figure 1.2.** Vectorization. A $3 \times 3$ image is read out row by row into a single column vector in $\mathbb{R}^9$; the same recipe turns a $28 \times 28$ digit into a vector in $\mathbb{R}^{784}$.

### 1.2.1 Data as vectors: vectorization

Any structured datum must be *vectorized* before a linear model can use it. A grayscale image is a grid of pixel intensities; stacking its rows (or columns) end to end turns the grid into a single tall vector. A $28 \times 28$ image, the size of a handwritten digit in the MNIST dataset, becomes a vector in $\mathbb{R}^{784}$ because $28 \times 28 = 784$ (Fig. 1.2). A color image with three channels becomes a 3-tensor (Section 1.7) and, flattened, a still longer vector. The price of vectorizing is that spatial structure (which pixels are neighbors) is scrambled into coordinate positions; recovering that structure is part of what convolutional networks do. For the linear models of this book, the vectorized form is the input.

### 1.2.2 Basis and coordinates

The entries of a vector are not absolute numbers; they are *coordinates with respect to a basis*. The *standard basis* of $\mathbb{R}^n$ is the set $\{\boldsymbol{e}_1, \dots, \boldsymbol{e}_n\}$, where $\boldsymbol{e}_i$ has a 1 in position $i$ and zeros elsewhere. Every vector expands uniquely as

$$\boldsymbol{x} \;=\; x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2 + \cdots + x_n\boldsymbol{e}_n \;=\; \sum_{i=1}^{n} x_i\,\boldsymbol{e}_i, \tag{1.3}$$

which simply restates that $x_i$ is the amount of $\boldsymbol{e}_i$ in $\boldsymbol{x}$ (Fig. 1.1). Other bases are possible and often far more useful: the right basis can make data look simple. Principal component analysis (Chapter 11) is, at heart, the search for a basis in which the data's variation lines up with the coordinate axes, and the singular value decomposition (Chapter 4) finds, for any matrix, the pair of bases in which it acts most simply. Choosing coordinates well is a recurring theme of the whole subject.

## 1.3 Inner products, norms, and orthogonality

To do geometry, length and angle, we need one more operation. Remarkably, both come from a single one.

**Definition 1.2 (Inner product and norm).** The *inner product* (or dot product) of $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ is the scalar

$$\langle \boldsymbol{x},\, \boldsymbol{y} \rangle \;=\; \boldsymbol{x}^{\top}\boldsymbol{y} \;=\; \sum_{i=1}^{n} x_i\, y_i.$$

The *Euclidean norm* (or $\ell_2$ norm) of $\boldsymbol{x}$ is

$$\|\boldsymbol{x}\| \;=\; \|\boldsymbol{x}\|_2 \;=\; \sqrt{\langle \boldsymbol{x},\, \boldsymbol{x} \rangle} \;=\; \Big(\textstyle\sum_{i=1}^{n} x_i^2\Big)^{1/2},$$

the length of the arrow. More generally the $\ell_p$ norm is $\|\boldsymbol{x}\|_p = \big(\sum_i |x_i|^p\big)^{1/p}$ for $p \geq 1$, with the special cases $p = 1$ (sum of absolute values) and $p = \infty$ (largest absolute entry, $\|\boldsymbol{x}\|_\infty = \max_i |x_i|$).

The inner product measures alignment, and the norm measures size, and the two are tied together by the angle. Writing $\vartheta$ for the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$,

$$\langle \boldsymbol{x},\, \boldsymbol{y} \rangle = \|\boldsymbol{x}\|\,\|\boldsymbol{y}\| \cos\vartheta, \tag{1.4}$$

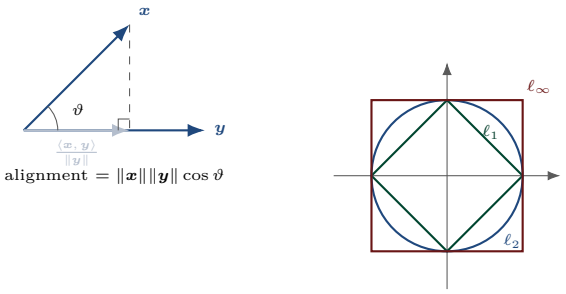



**Figure 1.3.** Left: the inner product as alignment. The signed length of the shadow of $\boldsymbol{x}$ on $\boldsymbol{y}$ is $\langle \boldsymbol{x}, \boldsymbol{y} \rangle / \|\boldsymbol{y}\|$, and $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \|\boldsymbol{x}\|\|\boldsymbol{y}\| \cos\vartheta$. Right: the unit balls $\{\boldsymbol{x} : \|\boldsymbol{x}\|_p \le 1\}$ for $p = 1$ (diamond), $p = 2$ (circle), and $p = \infty$ (square). The shape of the ball is what makes each norm behave differently, a fact that returns when sparsity-seeking $\ell_1$ penalties meet round $\ell_2$ ones.

| Norm | Formula | Unit ball | Role |
|---|---|---|---|
| $\ell_1$ | $\sum_i \lvert x_i \rvert$ | diamond (corners on axes) | promotes sparsity |
| $\ell_2$ | $\left(\sum_i x_i^2\right)^{1/2}$ | circle / sphere | length, least squares |
| $\ell_\infty$ | $\max_i \lvert x_i \rvert$ | square / cube | worst-case error |

**Table 1.1.** The three norms used throughout the book. The $\ell_2$ norm is the default "length"; the $\ell_1$ and $\ell_\infty$ norms appear in regularization and robustness. See Deisenroth et al. [5, Ch. 3].

so the inner product is large and positive when the vectors point the same way, large and negative when opposite, and exactly zero when they are perpendicular. Two vectors with $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$ are *orthogonal*, written $\boldsymbol{x} \perp \boldsymbol{y}$. This single notion, orthogonality, organizes least squares (Chapter 2), the spectral theorem (Chapter 3), and principal components (Chapter 11).

The three norms are compared in Table 1.1; their differently shaped unit balls (Fig. 1.3) are not a curiosity but the reason different norms produce different solutions to the same fitting problem.

The single most-used inequality in the subject bounds the inner product by the norms.

**Theorem 1.3 (Cauchy–Schwarz inequality).** For all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $\lvert \langle \boldsymbol{x}, \boldsymbol{y} \rangle \rvert \le \|\boldsymbol{x}\|\,\|\boldsymbol{y}\|$, with equality if and only if $\boldsymbol{x}$ and $\boldsymbol{y}$ are linearly dependent (one is a scalar multiple of the other).

*Proof.* If $\boldsymbol{y} = \boldsymbol{0}$ both sides are zero. Otherwise, for every $t \in \mathbb{R}$ the squared norm of $\boldsymbol{x} - t\boldsymbol{y}$ is nonnegative:

$$0 \le \|\boldsymbol{x} - t\boldsymbol{y}\|^2 = \langle \boldsymbol{x} - t\boldsymbol{y}, \boldsymbol{x} - t\boldsymbol{y} \rangle = \|\boldsymbol{x}\|^2 - 2t\,\langle \boldsymbol{x}, \boldsymbol{y} \rangle + t^2 \|\boldsymbol{y}\|^2.$$

The right-hand side is a quadratic in $t$ that is never negative, so its discriminant cannot be positive: $\big(2\,\langle \boldsymbol{x}, \boldsymbol{y} \rangle\big)^2 - 4\|\boldsymbol{y}\|^2\|\boldsymbol{x}\|^2 \le 0$, i.e. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle^2 \le \|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2$. Taking square roots gives the inequality. Equality forces the discriminant to vanish, so the quadratic has a real root $t_\star$ with $\|\boldsymbol{x} - t_\star \boldsymbol{y}\|^2 = 0$, that is $\boldsymbol{x} = t_\star \boldsymbol{y}$, which is linear dependence. ∎

A direct consequence is the *triangle inequality* $\|\boldsymbol{x} + \boldsymbol{y}\| \le \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$: expanding, $\|\boldsymbol{x} + \boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + 2\,\langle \boldsymbol{x}, \boldsymbol{y} \rangle + \|\boldsymbol{y}\|^2 \le \|\boldsymbol{x}\|^2 + 2\|\boldsymbol{x}\|\|\boldsymbol{y}\| + \|\boldsymbol{y}\|^2 = (\|\boldsymbol{x}\| + \|\boldsymbol{y}\|)^2$, where the middle step is Cauchy–Schwarz. Geometrically: no side of a triangle is longer than the sum of the other two.

**Example 1.4 (Norms and angle on numbers).** Take $\boldsymbol{x} = (3, -4, 12)$. Its three norms are

$$\|\boldsymbol{x}\|_1 = |3| + |-4| + |12| = 19, \qquad \|\boldsymbol{x}\|_2 = \sqrt{3^2 + 4^2 + 12^2} = \sqrt{169} = 13, \qquad \|\boldsymbol{x}\|_\infty = \max(3, 4, 12) = 12,$$

and they obey $\|\boldsymbol{x}\|_\infty \le \|\boldsymbol{x}\|_2 \le \|\boldsymbol{x}\|_1$, the universal ordering (the more terms a norm adds up, the larger it is). For the angle between $\boldsymbol{u} = (3, 4)$ and $\boldsymbol{v} = (4, 3)$, the inner product is $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 12 + 12 = 24$ and both have length 5, so

$$\cos\vartheta = \frac{\langle \boldsymbol{u}, \boldsymbol{v} \rangle}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|} = \frac{24}{25} = 0.96, \qquad \vartheta = \arccos(0.96) \approx 16.3^\circ.$$

The vectors are nearly aligned, as their near-equal coordinates suggest. Had we taken $\boldsymbol{u} = (1, 2, 2)$ and $\boldsymbol{v} = (2, 0, -1)$, then $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 2 + 0 - 2 = 0$, giving $\vartheta = 90^\circ$: orthogonal.

**Example 1.5 (Gram–Schmidt: building an orthonormal basis).** Orthonormal bases make everything downstream easier, because in such a basis a projection is just a dot product and a coordinate is just a component. Gram–Schmidt is the procedure that turns any independent set into an orthonormal one, peeling off the part of each new vector that the earlier ones already explain. Apply it to

$$\boldsymbol{a}_1 = (1,1,0), \qquad \boldsymbol{a}_2 = (1,0,1), \qquad \boldsymbol{a}_3 = (0,1,1).$$

**Step 1, normalize $\boldsymbol{a}_1$.** Its length is $\|\boldsymbol{a}_1\| = \sqrt{1^2+1^2+0^2} = \sqrt{2}$, so

$$\boldsymbol{q}_1 = \frac{\boldsymbol{a}_1}{\|\boldsymbol{a}_1\|} = \frac{1}{\sqrt{2}}(1,1,0).$$

**Step 2, remove the $\boldsymbol{q}_1$-part of $\boldsymbol{a}_2$.** The overlap is $\langle \boldsymbol{a}_2,\, \boldsymbol{q}_1\rangle = \frac{1}{\sqrt{2}}(1+0+0) = \frac{1}{\sqrt{2}}$, so the leftover is

$$\boldsymbol{v}_2 = \boldsymbol{a}_2 - \langle \boldsymbol{a}_2,\, \boldsymbol{q}_1\rangle\, \boldsymbol{q}_1 = (1,0,1) - \tfrac{1}{\sqrt{2}}\cdot\tfrac{1}{\sqrt{2}}(1,1,0) = (1,0,1) - \tfrac{1}{2}(1,1,0) = \big(\tfrac{1}{2}, -\tfrac{1}{2}, 1\big).$$

Its length is $\|\boldsymbol{v}_2\| = \sqrt{\frac{1}{4}+\frac{1}{4}+1} = \sqrt{\frac{3}{2}} = \frac{\sqrt{6}}{2}$, giving $\boldsymbol{q}_2 = \frac{1}{\sqrt{6}}(1,-1,2)$. **Step 3, remove both parts from $\boldsymbol{a}_3$.** Here $\langle \boldsymbol{a}_3,\, \boldsymbol{q}_1\rangle = \frac{1}{\sqrt{2}}(0+1+0) = \frac{1}{\sqrt{2}}$ and $\langle \boldsymbol{a}_3,\, \boldsymbol{q}_2\rangle = \frac{1}{\sqrt{6}}(0-1+2) = \frac{1}{\sqrt{6}}$, so

$$\boldsymbol{v}_3 = \boldsymbol{a}_3 - \tfrac{1}{\sqrt{2}}\boldsymbol{q}_1 - \tfrac{1}{\sqrt{6}}\boldsymbol{q}_2 = (0,1,1) - \tfrac{1}{2}(1,1,0) - \tfrac{1}{6}(1,-1,2) = \big(-\tfrac{2}{3}, \tfrac{2}{3}, \tfrac{2}{3}\big).$$

With $\|\boldsymbol{v}_3\| = \sqrt{3\cdot(2/3)^2} = \frac{2}{\sqrt{3}}$ we get $\boldsymbol{q}_3 = \frac{1}{\sqrt{3}}(-1,1,1)$. A quick check confirms the payoff: $\langle \boldsymbol{q}_1,\, \boldsymbol{q}_2\rangle = \langle \boldsymbol{q}_1,\, \boldsymbol{q}_3\rangle = \langle \boldsymbol{q}_2,\, \boldsymbol{q}_3\rangle = 0$ and each $\|\boldsymbol{q}_i\| = 1$, so $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3\}$ is an orthonormal basis of $\mathbb{R}^3$. Stacking the $\boldsymbol{q}_i$ as columns of $Q$ and recording the overlaps in an upper-triangular $R$ gives exactly the $QR$ factorization $A = QR$ that powers numerically stable least squares (Chapter 2).

## 1.4 Matrices as linear maps

A matrix is usually introduced as a rectangular array of numbers. The far more useful idea, and the one the whole course rests on, is that a matrix *is a function*: it eats a vector and returns a vector, doing so *linearly*.

**Definition 1.6 (Matrix).** A *matrix* $A \in \mathbb{R}^{m\times n}$ is an array of $mn$ real numbers with $m$ rows and $n$ columns, with entry $a_{ij}$ in row $i$, column $j$. We write $A = [\,\boldsymbol{a}_1\ \boldsymbol{a}_2\ \cdots\ \boldsymbol{a}_n\,]$ to name its columns $\boldsymbol{a}_j \in \mathbb{R}^m$, and its rows as $\boldsymbol{r}_1^{\top}, \ldots, \boldsymbol{r}_m^{\top}$.

**Definition 1.7 (Linear map).** A matrix $A \in \mathbb{R}^{m\times n}$ defines a map $A : \mathbb{R}^n \to \mathbb{R}^m$ by $\boldsymbol{x} \mapsto A\boldsymbol{x}$. This map is *linear*: for all $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^n$ and scalars $\alpha, \beta \in \mathbb{R}$,

$$A(\alpha\boldsymbol{x} + \beta\boldsymbol{z}) \;=\; \alpha\, A\boldsymbol{x} + \beta\, A\boldsymbol{z}. \tag{1.5}$$

Conversely, every linear map from $\mathbb{R}^n$ to $\mathbb{R}^m$ is multiplication by a unique matrix, whose $j$th column is the image $A\boldsymbol{e}_j$ of the $j$th basis vector.

**Example 1.8 (A rotation is a linear map).** Rotation of the plane by $90^\circ$ counterclockwise is the linear map with matrix $R = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$, found by rotating the basis vectors: $\boldsymbol{e}_1 = (1,0) \mapsto (0,1)$ and $\boldsymbol{e}_2 = (0,1) \mapsto (-1,0)$ become the columns. Applying it, $R(3,2)^{\top} = (-2,3)^{\top}$, the point $(3,2)$ swung a quarter turn. Linearity is visible: rotating $2\boldsymbol{u} + \boldsymbol{v}$ is the same as rotating each and combining, $R(2\boldsymbol{u}+\boldsymbol{v}) = 2R\boldsymbol{u} + R\boldsymbol{v}$, because rotation preserves the parallelogram of vector addition. Notice $R$ has *no real eigenvectors* (every nonzero vector genuinely turns), and indeed its eigenvalues are the complex pair $e^{\pm i\pi/2} = \pm i$ (Exercise 3.5); a real $2\times 2$ matrix need not have any real eigenvalue, even though a real $3\times 3$ one must.

*Intuition.* A linear map is one that respects the vector-space operations: it does not matter whether you add and scale first and then apply $A$, or apply $A$ first and then add and scale. This is exactly why the standard basis determines everything: once you know where $A$ sends each $\boldsymbol{e}_j$, linearity forces where it sends every other vector, namely the same combination of the images. A matrix is just a table of those images, one per column.

The product $A\boldsymbol{x}$ has two readings we use constantly. The *row reading* computes entry $i$ of the output as the inner product of row $i$ with $\boldsymbol{x}$, namely $(A\boldsymbol{x})_i = \boldsymbol{r}_i^{\top}\boldsymbol{x} = \sum_j a_{ij}x_j$. The *column reading* sees the output as a combination of the columns of $A$:

$$A\boldsymbol{x} \;=\; x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2 + \cdots + x_n\boldsymbol{a}_n \;=\; \sum_{j=1}^{n} x_j\, \boldsymbol{a}_j. \tag{1.6}$$

| View | Formula | What it reveals |
|---|---|---|
| Entrywise (row × column) | $C_{ij} = \sum_k A_{ik} B_{kj}$ | how to compute by hand or code |
| Columnwise | $C = [\, A\boldsymbol{b}_1 \ \cdots \ A\boldsymbol{b}_q \,]$ | each output column is a combination of $A$'s columns |
| Rank-one expansion | $C = \sum_k \boldsymbol{a}_k \boldsymbol{b}_k^\top$ | $C$ is a sum of rank-one tiles (the SVD's view) |

**Table 1.2.** Three equivalent views of the product $AB$. They agree as numbers but emphasize different structure; the rank-one view drives Chapter 4.

The column reading in Eq. (1.6) is the one to keep in mind: applying $A$ to $\boldsymbol{x}$ mixes the columns of $A$ using the entries of $\boldsymbol{x}$ as the recipe. It tells you immediately which outputs are reachable, the point we develop in Section 1.6.

## 1.5 Three views of matrix multiplication

Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times q}$. Their product $C = AB \in \mathbb{R}^{m\times q}$ can be seen three equivalent ways, summarized in Table 1.2. Each computes the same numbers, but each makes a different structure obvious, and fluency means switching between them at will.

### 1.5.1 View 1: entrywise (row times column)

The textbook definition sets entry $(i, j)$ of $C$ to the inner product of row $i$ of $A$ with column $j$ of $B$:

$$C_{ij} \ = \ \sum_{k=1}^{n} A_{ik}\, B_{kj}. \tag{1.7}$$

This is how code computes a product, but it hides the geometry.

### 1.5.2 View 2: columnwise (apply $A$ to each column of $B$)

Writing $B = [\, \boldsymbol{b}_1 \ \cdots \ \boldsymbol{b}_q \,]$ by its columns and applying the column reading Eq. (1.6) one column at a time,

$$C = AB = \big[\, A\boldsymbol{b}_1 \ \ A\boldsymbol{b}_2 \ \ \cdots \ \ A\boldsymbol{b}_q \,\big], \qquad A\boldsymbol{b}_j = \sum_{k=1}^{n} (b_j)_k\, \boldsymbol{a}_k. \tag{1.8}$$

Each column of the product is a linear combination of the columns of $A$. In machine learning, multiplying a feature matrix by a weight vector produces a combination of feature columns, and thinking in column space tells you when a target can be matched.

### 1.5.3 View 3: the rank-one expansion (outer products)

Group the entrywise sum Eq. (1.7) by the summation index $k$ rather than by output position. The $k$th term contributes column $k$ of $A$ times row $k$ of $B$:

$$\boxed{AB \ = \ \sum_{k=1}^{n} \boldsymbol{a}_k\, \boldsymbol{b}_k^\top} \tag{1.9}$$

where each $\boldsymbol{a}_k \boldsymbol{b}_k^\top \in \mathbb{R}^{m\times q}$ is an *outer product*, a rank-one matrix (Section 1.6). To check that Eq. (1.9) agrees with Eq. (1.7), read off entry $(i, j)$ of the right side: $\sum_k (\boldsymbol{a}_k \boldsymbol{b}_k^\top)_{ij} = \sum_k (\boldsymbol{a}_k)_i (\boldsymbol{b}_k)_j = \sum_k A_{ik} B_{kj} = C_{ij}$.

**Example 1.9 (One product, three ways).** Let $A = \left[\begin{smallmatrix} 2 & 3 \\ -1 & 4 \end{smallmatrix}\right]$ and $B = \left[\begin{smallmatrix} 0 & 5 \\ -2 & 3 \end{smallmatrix}\right]$.

**View 1 (entrywise).** $C_{11} = 2(0) + 3(-2) = -6$, $C_{12} = 2(5) + 3(3) = 19$, $C_{21} = (-1)(0) + 4(-2) = -8$, $C_{22} = (-1)(5) + 4(3) = 7$, so $C = \left[\begin{smallmatrix} -6 & 19 \\ -8 & 7 \end{smallmatrix}\right]$.

**View 2 (columnwise).** With $\boldsymbol{b}_1 = (0, -2)^\top$ and $\boldsymbol{b}_2 = (5, 3)^\top$,

$$A\boldsymbol{b}_1 = 0 \begin{bmatrix} 2 \\ -1 \end{bmatrix} - 2 \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} -6 \\ -8 \end{bmatrix}, \quad A\boldsymbol{b}_2 = 5 \begin{bmatrix} 2 \\ -1 \end{bmatrix} + 3 \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 19 \\ 7 \end{bmatrix},$$

and stacking these columns reproduces $C$.

**View 3 (rank-one tiles).** With columns $\boldsymbol{a}_1 = (2,-1)^\top, \boldsymbol{a}_2 = (3,4)^\top$ of $A$ and rows $\boldsymbol{b}_1^\top = [0\ 5], \boldsymbol{b}_2^\top = [-2\ 3]$ of $B$,

$$\boldsymbol{a}_1\boldsymbol{b}_1^\top = \begin{bmatrix} 0 & 10 \\ 0 & -5 \end{bmatrix}, \quad \boldsymbol{a}_2\boldsymbol{b}_2^\top = \begin{bmatrix} -6 & 9 \\ -8 & 12 \end{bmatrix},$$

and their sum is again $\left[\begin{smallmatrix} -6 & 19 \\ -8 & 7 \end{smallmatrix}\right]$. The three views are three bookkeepings of the same arithmetic.

**Example 1.10 (A $3 \times 3$ product, every entry).** With larger matrices the row-by-column rule is the same; only the bookkeeping grows. Let

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 3 \\ 2 & 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 & 1 \\ 2 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Each entry $C_{ij} = \sum_{k=1}^{3} A_{ik}B_{kj}$ is the dot product of row $i$ of $A$ with column $j$ of $B$. Working the first row of $C$:

$$C_{11} = 1{\cdot}1 + 2{\cdot}2 + 0{\cdot}0 = 5, \quad C_{12} = 1{\cdot}0 + 2{\cdot}1 + 0{\cdot}1 = 2, \quad C_{13} = 1{\cdot}1 + 2{\cdot}0 + 0{\cdot}1 = 1.$$

The second row uses row $(0,1,3)$ of $A$:

$$C_{21} = 0{\cdot}1 + 1{\cdot}2 + 3{\cdot}0 = 2, \quad C_{22} = 0{\cdot}0 + 1{\cdot}1 + 3{\cdot}1 = 4, \quad C_{23} = 0{\cdot}1 + 1{\cdot}0 + 3{\cdot}1 = 3.$$

The third uses row $(2,0,1)$:

$$C_{31} = 2{\cdot}1 + 0{\cdot}2 + 1{\cdot}0 = 2, \quad C_{32} = 2{\cdot}0 + 0{\cdot}1 + 1{\cdot}1 = 1, \quad C_{33} = 2{\cdot}1 + 0{\cdot}0 + 1{\cdot}1 = 3.$$

Collecting the nine numbers,

$$AB = \begin{bmatrix} 5 & 2 & 1 \\ 2 & 4 & 3 \\ 2 & 1 & 3 \end{bmatrix}.$$

Note $AB \neq BA$ in general: recomputing $BA$ here gives $\left[\begin{smallmatrix} 3 & 2 & 1 \\ 2 & 5 & 3 \\ 2 & 1 & 4 \end{smallmatrix}\right]$, a different matrix, so the order of multiplication matters even when both products are defined.

The rank-one view is not a curiosity. Modern hardware performs vast numbers of rank-one-like updates, and the search for matrix-multiplication algorithms that use *fewer* multiplications is exactly the search for shorter sums of rank-one tiles; see the aside in Section 1.12.

## 1.6 Rank, span, and column space

**Definition 1.11 (Linear independence and span).** Vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{R}^m$ are *linearly independent* if the only scalars with $\sum_i c_i\boldsymbol{v}_i = \boldsymbol{0}$ are $c_1 = \cdots = c_k = 0$. Their *span* is the set of all linear combinations, $\operatorname{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\} = \left\{\sum_i c_i\boldsymbol{v}_i \,\middle|\, c_i \in \mathbb{R}\right\}$, a subspace of $\mathbb{R}^m$. The *dimension* of a subspace is the number of vectors in any basis (any maximal independent subset).

**Definition 1.12 (Column space, row space, rank).** The *column space* of $A \in \mathbb{R}^{m\times n}$ is $\operatorname{Col}(A) = \operatorname{span}\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\} \subseteq \mathbb{R}^m$, the span of its columns. The *row space* $\operatorname{Row}(A) \subseteq \mathbb{R}^n$ is the span of its rows. The *rank* is $\operatorname{rank}(A) = \dim \operatorname{Col}(A)$, the maximum number of linearly independent columns.

*Intuition.* The column space is the set of places $A$ can reach. By the column reading Eq. (1.6), the outputs $A\boldsymbol{x}$ are exactly the combinations of the columns of $A$; nothing outside their span is attainable, no matter what $\boldsymbol{x}$ you feed in. Rank counts how many genuinely independent directions of output $A$ has. A tall matrix with rank $r$ maps all of $\mathbb{R}^n$ onto an $r$-dimensional sheet inside $\mathbb{R}^m$, and that sheet is where the answer to any equation $A\boldsymbol{x} = \boldsymbol{b}$ must live.

**Proposition 1.13 (Solvability).** The system $A\boldsymbol{x} = \boldsymbol{b}$ has a solution if and only if $\boldsymbol{b} \in \operatorname{Col}(A)$. If $\operatorname{rank}(A) = r < m$, the reachable set is only an $r$-dimensional subspace of $\mathbb{R}^m$, so most targets $\boldsymbol{b}$ are unattainable exactly.

**Property 1.14 (Core facts about rank).** For $A \in \mathbb{R}^{m\times n}$ and conformable $B$:

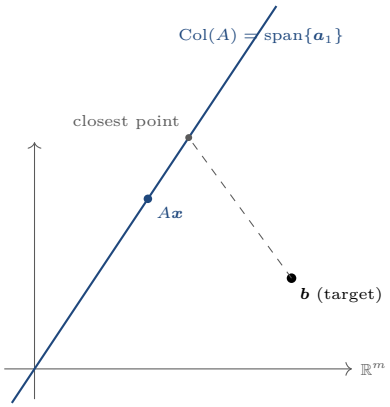



**Figure 1.4.** The column space of $A = \left[\begin{smallmatrix}1&2\\2&4\\3&6\end{smallmatrix}\right]$ is the line $\operatorname{span}\{(1,2,3)\}$ (drawn schematically). Every output $A\boldsymbol{x}$ lies on this line, so a target $\boldsymbol{b}$ off the line cannot be hit exactly. The best we can do is the closest point on the line, the orthogonal projection of $\boldsymbol{b}$ onto $\operatorname{Col}(A)$, the subject of Chapter 2.

(i) $0 \le \operatorname{rank}(A) \le \min\{m, n\}$;
(ii) $\operatorname{rank}(A) = \operatorname{rank}(A^\top)$ (row rank equals column rank, Exercise 1.3);
(iii) $\operatorname{rank}(AB) \le \min\{\operatorname{rank}(A), \operatorname{rank}(B)\}$;
(iv) $A \in \mathbb{R}^{n\times n}$ is invertible if and only if $\operatorname{rank}(A) = n$ (full rank).

**Example 1.15 (Rank, column space, and null space by elimination).** Let $A = \left[\begin{smallmatrix}1&2&3\\2&4&6\\0&1&1\end{smallmatrix}\right]$. Row-reduce: the operation $R_2 \leftarrow R_2 - 2R_1$ wipes out the second row (it was twice the first), leaving

$$A \xrightarrow{\text{reduce}} \begin{bmatrix}1&2&3\\0&1&1\\0&0&0\end{bmatrix}.$$

There are two pivots (columns 1 and 2), so $\operatorname{rank}(A) = 2$. **Column space.** A basis is the two pivot columns of the *original* $A$, $\boldsymbol{a}_1 = (1,2,0)$ and $\boldsymbol{a}_2 = (2,4,1)$; the third column is dependent, $\boldsymbol{a}_3 = \boldsymbol{a}_1 + \boldsymbol{a}_2 = (3,6,1)$, so $\operatorname{Col}(A)$ is the plane $\operatorname{span}\{\boldsymbol{a}_1, \boldsymbol{a}_2\} \subset \mathbb{R}^3$. **Null space.** Solve $A\boldsymbol{x} = \boldsymbol{0}$ from the reduced rows: $x_2 + x_3 = 0$ gives $x_2 = -x_3$, and $x_1 + 2x_2 + 3x_3 = 0$ gives $x_1 = -2x_2 - 3x_3 = -x_3$. With $x_3$ free, $\boldsymbol{x} = x_3(-1,-1,1)$, so $\operatorname{Nul}(A) = \operatorname{span}\{(1,1,-1)\}$, one-dimensional. One checks $A(1,1,-1)^\top = (1{+}2{-}3,\ 2{+}4{-}6,\ 0{+}1{-}1)^\top = \boldsymbol{0}$. **Rank–nullity.** $\operatorname{rank}(A) + \dim \operatorname{Nul}(A) = 2 + 1 = 3 = n$, as the theorem requires: each of the three columns either adds a new output direction (the two pivots) or records a dependence (the one free variable).

**Example 1.16 (Testing independence with a determinant).** For $n$ vectors in $\mathbb{R}^n$, the fastest independence test is the determinant of the matrix that has them as columns: the vectors are independent exactly when that determinant is nonzero (equivalently, the matrix is invertible, Table 1.3). *Independent set.* Stack $\boldsymbol{v}_1 = (1,1,0)$, $\boldsymbol{v}_2 = (1,0,1)$, $\boldsymbol{v}_3 = (0,1,1)$ as columns and expand along the first row:

$$\det\begin{bmatrix}1&1&0\\1&0&1\\0&1&1\end{bmatrix} = 1\det\begin{bmatrix}0&1\\1&1\end{bmatrix} - 1\det\begin{bmatrix}1&1\\0&1\end{bmatrix} + 0 = 1(0-1) - 1(1-0) = -2 \neq 0,$$

so the three vectors are linearly independent and span all of $\mathbb{R}^3$. *Dependent set.* For $\boldsymbol{w}_1 = (1,2,3)$, $\boldsymbol{w}_2 = (2,4,6)$, $\boldsymbol{w}_3 = (1,1,1)$, the second column is exactly twice the first, $\boldsymbol{w}_2 = 2\boldsymbol{w}_1$, so the columns are dependent and the determinant must be 0 (a matrix with a repeated direction collapses volume to zero, Fig. 1.5). No computation past spotting $\boldsymbol{w}_2 = 2\boldsymbol{w}_1$ is needed. The determinant thus answers "do these vectors point in genuinely different directions?" with a single number: nonzero means yes, zero means at least one is redundant.

### 1.6.1 Rank-one matrices: the atom

A matrix has rank one exactly when it is a single outer product $\boldsymbol{u}\boldsymbol{v}^\top$ with $\boldsymbol{u} \neq \mathbf{0}$, $\boldsymbol{v} \neq \mathbf{0}$. Every column of $\boldsymbol{u}\boldsymbol{v}^\top$ is a multiple of $\boldsymbol{u}$ (the $j$th column is $v_j\boldsymbol{u}$), so its column space is the line $\operatorname{span}\{\boldsymbol{u}\}$ and its rank is 1. View 3 of multiplication (Eq. (1.9)) then says: *every* matrix is a sum of rank-one atoms. The singular value decomposition (Chapter 4) makes that sum canonical, using orthogonal atoms ordered by importance, and that is the single most useful structural fact in the book.

## 1.7 Tensors

A *tensor* generalizes the progression scalar, vector, matrix to higher order. A scalar is a 0-tensor, a vector a 1-tensor, and a matrix a 2-tensor. A color image of height $H$ and width $W$ with three channels is a 3-tensor in $\mathbb{R}^{H\times W\times 3}$, and a batch of such images is a 4-tensor. Formally a tensor is a multilinear map, linear in each argument separately. We will not need the full machinery, but the word recurs in two places: the data fed to vision models is tensor-shaped, and the cost of matrix multiplication is governed by the rank of a particular 3-tensor (Section 1.12).

## 1.8 The determinant

**Definition 1.17 (Determinant).** The *determinant* $\det(A)$ of a square matrix $A \in \mathbb{R}^{n\times n}$ is the unique scalar function of the rows of $A$ that is (i) linear in each row separately, (ii) zero whenever two rows are equal, and (iii) equal to 1 for the identity. For small matrices,

$$\det\begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc, \quad \det\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = a(ei - fh) - b(di - fg) + c(dh - eg),$$

the $3 \times 3$ formula being the cofactor expansion along the first row.

**Example 1.18 (A $3 \times 3$ determinant, expanding along the sparsest line).** Compute $\det\left[\begin{smallmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 0 & 0 \end{smallmatrix}\right]$. Cofactor expansion costs less along a row or column with zeros, and the third row $(1, 0, 0)$ has two, so expand there:

$$\det = 1 \cdot C_{31} + 0 \cdot C_{32} + 0 \cdot C_{33}, \qquad C_{31} = (-1)^{3+1} \det\begin{bmatrix} 1 & 1 \\ 3 & 2 \end{bmatrix} = +(1 \cdot 2 - 1 \cdot 3) = -1,$$

so $\det = -1$. As a check, expanding along the first row instead,

$$2(3 \cdot 0 - 2 \cdot 0) - 1(1 \cdot 0 - 2 \cdot 1) + 1(1 \cdot 0 - 3 \cdot 1) = 0 + 2 - 3 = -1,$$

the same value. Since $\det \neq 0$, the matrix is invertible and its rows (equivalently columns) are linearly independent.

Property (i) is what we mean by the determinant being *multilinear*, verified explicitly for $n = 2$ in Exercise 1.5. The determinant carries two pieces of information we care about.

- **Invertibility.** $A$ is invertible if and only if $\det(A) \neq 0$. For $2 \times 2$ the inverse is

$$A^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}, \tag{1.10}$$

  which exists exactly when $ad - bc \neq 0$.
- **Volume.** $|\det(A)|$ is the factor by which the linear map $A$ scales volume. The unit cube of $\mathbb{R}^n$ maps to a parallelepiped of volume $|\det(A)|$, and the sign of $\det(A)$ records whether orientation is preserved (Fig. 1.5). If $\det(A) = 0$, the map flattens space into a lower-dimensional set, collapsing volume to zero and destroying the information needed to invert.

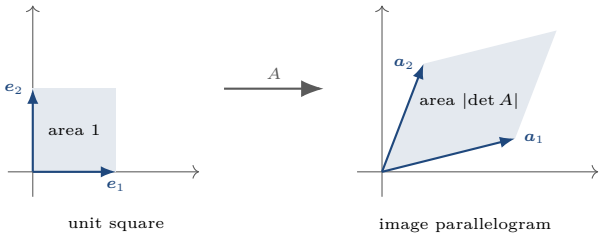



**Figure 1.5.** The determinant as signed area. The map $A$ sends the unit square (area 1) to the parallelogram spanned by the columns $\boldsymbol{a}_1, \boldsymbol{a}_2$ of $A$, whose area is $|\det A|$. When $\det A = 0$ the parallelogram collapses to a segment and $A$ is not invertible.

| **For a square matrix $A \in \mathbb{R}^{n\times n}$, the following are equivalent:** | |
|---|---|
| $A$ is invertible | $\det(A) \neq 0$ |
| $\operatorname{rank}(A) = n$ (full rank) | the columns of $A$ are linearly independent |
| $A\boldsymbol{x} = \boldsymbol{b}$ has a unique solution for every $\boldsymbol{b}$ | the only solution of $A\boldsymbol{x} = \boldsymbol{0}$ is $\boldsymbol{x} = \boldsymbol{0}$ |
| $\operatorname{Col}(A) = \mathbb{R}^n$ | 0 is not an eigenvalue of $A$ (Chapter 3) |

**Table 1.3.** The invertible matrix theorem: a dozen ways of saying the same thing. Any one fails exactly when all fail.

## 1.9 Invertibility and conditioning

**Definition 1.19 (Inverse).** A square matrix $A \in \mathbb{R}^{n\times n}$ is *invertible* (nonsingular) if there is a matrix $A^{-1}$ with $AA^{-1} = A^{-1}A = I_n$. When it exists, the inverse is unique.

Invertibility has many equivalent faces, collected in Table 1.3; this list is sometimes called the *invertible matrix theorem*, and it is worth knowing because a question phrased in terms of one condition is often easiest to answer through another.

*Remark* 1.20 *(Singular matrices are rare but dangerous).* The map $A \mapsto \det(A)$ is continuous in the entries of $A$. The invertible matrices $\{A : \det A \neq 0\}$ therefore form an *open* set: a small perturbation of an invertible matrix stays invertible. The singular matrices $\{A : \det A = 0\}$ form a *closed*, measure-zero set, a thin surface in the $n^2$-dimensional space of matrices, so a random matrix is invertible with probability one. The danger is *near* that surface: matrices close to singular are invertible in principle but *ill-conditioned*, and their inverses are enormous and numerically unstable. The sensitivity is measured by the *condition number* $\kappa(A) = \sigma_{\max}/\sigma_{\min}$ (Chapter 4); a large $\kappa$ means small errors in the data can produce large errors in $A^{-1}\boldsymbol{b}$, the central concern of numerical linear algebra [24].

**Example 1.21 (A near-singular family).** Consider $A_\varepsilon = \begin{bmatrix} 1 & 1 \\ 1 & 1+\varepsilon \end{bmatrix}$, with $\det(A_\varepsilon) = (1)(1+\varepsilon) - (1)(1) = \varepsilon$. For every $\varepsilon \neq 0$ it is invertible, with $A_\varepsilon^{-1} = \frac{1}{\varepsilon}\begin{bmatrix} 1+\varepsilon & -1 \\ -1 & 1 \end{bmatrix}$ by Eq. (1.10). As $\varepsilon \to 0$ the determinant tends to zero and the entries of $A_\varepsilon^{-1}$ blow up like $1/\varepsilon$. The matrix never stops being invertible until $\varepsilon = 0$ exactly, yet the inverse becomes useless long before then. This is the practical reason we avoid forming inverses when we can, preferring the least-squares machinery of Chapter 2.

### 1.9.1 Computing the inverse by Gaussian elimination

When an inverse is genuinely needed, the reliable hand method augments $A$ with the identity and row-reduces until the left block becomes the identity; the right block is then $A^{-1}$:

$$\left[\, A \mid I \,\right] \xrightarrow{\text{row operations}} \left[\, I \mid A^{-1} \,\right].$$

The allowed row operations (scale a row, swap two rows, add a multiple of one row to another) are each invertible linear maps, so the relationship between the two blocks is preserved throughout. (That this elimination, not some cleverer scheme, was long believed optimal is the very assumption Strassen overturned; see Section 1.12.)

**Example 1.22 (A full $3\times 3$ inverse, every step).** Let $A = \left[\begin{smallmatrix} 2 & 1 & 0 \\ -1 & 1 & 1 \\ 0 & 2 & 3 \end{smallmatrix}\right]$. Augment with $I_3$ and reduce.

$$\left[\begin{array}{ccc|ccc} 2 & 1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 2 & 3 & 0 & 0 & 1 \end{array}\right] \xrightarrow{R_1 \to \frac{1}{2}R_1} \left[\begin{array}{ccc|ccc} 1 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ -1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 2 & 3 & 0 & 0 & 1 \end{array}\right]$$

$$\xrightarrow{R_2 \to R_2 + R_1} \left[\begin{array}{ccc|ccc} 1 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{3}{2} & 1 & \frac{1}{2} & 1 & 0 \\ 0 & 2 & 3 & 0 & 0 & 1 \end{array}\right] \xrightarrow{R_3 \to R_3 - \frac{4}{3}R_2} \left[\begin{array}{ccc|ccc} 1 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{3}{2} & 1 & \frac{1}{2} & 1 & 0 \\ 0 & 0 & \frac{5}{3} & -\frac{2}{3} & -\frac{4}{3} & 1 \end{array}\right]$$

$$\xrightarrow{R_3 \to \frac{3}{5}R_3} \left[\begin{array}{ccc|ccc} 1 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{3}{2} & 1 & \frac{1}{2} & 1 & 0 \\ 0 & 0 & 1 & -\frac{2}{5} & -\frac{4}{5} & \frac{3}{5} \end{array}\right] \xrightarrow{R_2 \to R_2 - R_3} \left[\begin{array}{ccc|ccc} 1 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{3}{2} & 0 & \frac{9}{10} & \frac{9}{5} & -\frac{3}{5} \\ 0 & 0 & 1 & -\frac{2}{5} & -\frac{4}{5} & \frac{3}{5} \end{array}\right]$$

$$\xrightarrow{R_2 \to \frac{2}{3}R_2} \left[\begin{array}{ccc|ccc} 1 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 & \frac{3}{5} & \frac{6}{5} & -\frac{2}{5} \\ 0 & 0 & 1 & -\frac{2}{5} & -\frac{4}{5} & \frac{3}{5} \end{array}\right] \xrightarrow{R_1 \to R_1 - \frac{1}{2}R_2} \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{1}{5} & -\frac{3}{5} & \frac{1}{5} \\ 0 & 1 & 0 & \frac{3}{5} & \frac{6}{5} & -\frac{2}{5} \\ 0 & 0 & 1 & -\frac{2}{5} & -\frac{4}{5} & \frac{3}{5} \end{array}\right]$$

The left block is now $I_3$, so $A^{-1} = \frac{1}{5}\left[\begin{smallmatrix} 1 & -3 & 1 \\ 3 & 6 & -2 \\ -2 & -4 & 3 \end{smallmatrix}\right]$. One verifies $AA^{-1} = I_3$ directly.

## 1.10 The trace

**Definition 1.23 (Trace).** The *trace* of a square matrix $A \in \mathbb{R}^{n\times n}$ is the sum of its diagonal entries, $\operatorname{tr}(A) = \sum_{i=1}^{n} a_{ii}$.

The trace is linear, $\operatorname{tr}(\alpha A + \beta B) = \alpha \operatorname{tr}(A) + \beta \operatorname{tr}(B)$, and it satisfies the *cyclic property*

$$\operatorname{tr}(AB) = \operatorname{tr}(BA) \tag{1.11}$$

whenever both products are square (Exercise 1.8). Two facts we will use repeatedly, both proved once we have eigenvalues in Chapter 3: the trace equals the sum of the eigenvalues, and the determinant equals their product. The trace also turns the squared Frobenius norm of a matrix into $\operatorname{tr}(A^\top A) = \sum_{ij} a_{ij}^2$, which is how least-squares costs get differentiated in Chapter 8.

## 1.11 The linear model for supervised learning

We can now state the central computational problem of the first part of the course. Stack the $m$ training inputs from Eq. (1.2) as the rows of a *design matrix*, and the targets as a vector:

$$X = \begin{bmatrix} (\boldsymbol{x}^{(1)})^\top \\ (\boldsymbol{x}^{(2)})^\top \\ \vdots \\ (\boldsymbol{x}^{(m)})^\top \end{bmatrix} \in \mathbb{R}^{m\times n}, \qquad \boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \in \mathbb{R}^m. \tag{1.12}$$

A *linear model* posits that the target is approximately a linear function of the features, $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x}$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^n$, so that for all examples at once

$$\hat{\boldsymbol{y}} = X\boldsymbol{\theta} \approx \boldsymbol{y}. \tag{1.13}$$

Learning means choosing $\boldsymbol{\theta}$. The tempting move is to solve $X\boldsymbol{\theta} = \boldsymbol{y}$ by inverting $X$, but this fails for structural reasons, each an instance of something in this chapter:

- $X$ is usually *tall* ($m \gg n$, far more examples than features), so it is not square and has no inverse;
- even when $m = n$, $X$ may be singular or, by Remark 1.20, so close to singular that its inverse is meaningless;
- real data is noisy, so exact equality $X\boldsymbol{\theta} = \boldsymbol{y}$ generally has no solution at all, because $\boldsymbol{y} \notin \operatorname{Col}(X)$ (Proposition 1.13).

The way out, developed in Chapter 2, is to stop demanding equality and instead make $X\boldsymbol{\theta}$ *as close as possible* to $\boldsymbol{y}$ in the Euclidean sense,

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^n} \|X\boldsymbol{\theta}-\boldsymbol{y}\|_2^2, \tag{1.14}$$

which by Fig. 1.4 amounts to projecting $\boldsymbol{y}$ onto the column space of $X$. That projection, the resulting *normal equations*, and the *pseudoinverse* that solves them are the next chapter.

## 1.12 Aside: AlphaTensor and the rank-one view

**Enrichment: how fast can we multiply matrices?**

The number of scalar multiplications needed to multiply two matrices is a rank problem in disguise, which is why the rank-one expansion Eq. (1.9) is worth internalizing.
Multiplying two $2\times 2$ matrices the obvious way uses 8 scalar multiplications. In 1969 Strassen found a scheme using only 7, which, applied recursively to large matrices, gives a genuinely faster algorithm and overturned the belief that ordinary Gaussian elimination was optimal [22]; Winograd later showed 7 is optimal for $2\times 2$ over the reals. The task of multiplying $n\times n$ matrices is encoded by a fixed three-dimensional tensor $\mathcal{T}$, and an algorithm corresponds to writing $\mathcal{T}$ as a sum of rank-one tensor terms, one term per scalar multiplication. Fewer terms means a faster algorithm. In 2022 DeepMind's *AlphaTensor* framed this as a single-player game and trained a reinforcement-learning agent to peel off rank-one terms until the tensor vanished, rediscovering Strassen's algorithm and, for $4\times 4$ matrices over $\mathbb{Z}_2$, beating the previous record with 47 multiplications instead of 49 [7]. The lesson for us is the lens: complicated computation, decomposed into rank-one atoms, the same lens the SVD will turn on every matrix in Chapter 4.

## 1.13 Summary

- Data becomes a vector by *vectorization*; a dataset becomes the design matrix $X$ of Eq. (1.12). Vectors carry coordinates with respect to a basis, and the inner product supplies length, angle, and orthogonality.
- A matrix *is* a linear map. Reading $A\boldsymbol{x}$ as a combination of columns (Eq. (1.6)) shows the reachable outputs form the column space $\mathrm{Col}(A)$.
- Matrix multiplication has three faces: entrywise, columnwise, and the rank-one expansion $AB=\sum_k \boldsymbol{a}_k\boldsymbol{b}_k^\top$ (Table 1.2). Rank-one matrices are the atoms from which every matrix is built.
- Rank counts independent columns and equals the rank of the transpose. The determinant tests invertibility (Table 1.3) and measures volume; near-singular matrices are invertible but ill-conditioned.
- The supervised linear model $X\boldsymbol{\theta}\approx\boldsymbol{y}$ usually cannot be solved by inversion, which motivates least squares (Chapter 2).

## Exercises

**Exercise 1.1.** Let $A=\left[\begin{smallmatrix}2&-1&-2\\-4&2&-3\end{smallmatrix}\right]$ and $B=\left[\begin{smallmatrix}2&-1\\-1&2\\7&-3\end{smallmatrix}\right]$. Compute $AB$ three ways: (a) entrywise, (b) as linear combinations of the columns of $A$, and (c) as a sum of rank-one outer products. Verify all three agree.

*Solution.* **(a) Entrywise**, using Eq. (1.7): $(AB)_{11}=2(2)+(-1)(-1)+(-2)(7)=-9$, $(AB)_{12}=2(-1)+(-1)(2)+(-2)(-3)=2$, $(AB)_{21}=-4(2)+2(-1)+(-3)(7)=-31$, $(AB)_{22}=-4(-1)+2(2)+(-3)(-3)=17$, so $AB=\left[\begin{smallmatrix}-9&2\\-31&17\end{smallmatrix}\right]$.

**(b) Columnwise.** With $\boldsymbol{b}_1=(2,-1,7)^\top$, $\boldsymbol{b}_2=(-1,2,-3)^\top$ and columns $\boldsymbol{a}_1=(2,-4)^\top, \boldsymbol{a}_2=(-1,2)^\top, \boldsymbol{a}_3=(-2,-3)^\top$,

$$A\boldsymbol{b}_1=2\boldsymbol{a}_1-\boldsymbol{a}_2+7\boldsymbol{a}_3=\begin{bmatrix}-9\\-31\end{bmatrix},\qquad A\boldsymbol{b}_2=-\boldsymbol{a}_1+2\boldsymbol{a}_2-3\boldsymbol{a}_3=\begin{bmatrix}2\\17\end{bmatrix}.$$

**(c) Rank-one expansion**, with rows $\boldsymbol{b}_1^\top=[2\ -1], \boldsymbol{b}_2^\top=[-1\ 2], \boldsymbol{b}_3^\top=[7\ -3]$: $\boldsymbol{a}_1\boldsymbol{b}_1^\top+\boldsymbol{a}_2\boldsymbol{b}_2^\top+\boldsymbol{a}_3\boldsymbol{b}_3^\top=\left[\begin{smallmatrix}4&-2\\-8&4\end{smallmatrix}\right]+\left[\begin{smallmatrix}1&-2\\-2&4\end{smallmatrix}\right]+\left[\begin{smallmatrix}-14&6\\-21&9\end{smallmatrix}\right]=\left[\begin{smallmatrix}-9&2\\-31&17\end{smallmatrix}\right]$, matching (a) and (b). □

**Exercise 1.2.** Let $b, c \in \mathbb{R}$ and define $T : \mathbb{R}^3 \to \mathbb{R}^2$ by $T(x, y, z) = (2x - 4y + 3z + b,\ 6x + c\,xyz)$. Show that $T$ is linear if and only if $b = c = 0$.

*Solution.* ($\Leftarrow$) If $b = c = 0$, $T(x, y, z) = (2x - 4y + 3z,\ 6x)$. Each output coordinate is a linear function of $(x, y, z)$, so $T(\alpha \boldsymbol{u}_1 + \beta \boldsymbol{u}_2) = \alpha T(\boldsymbol{u}_1) + \beta T(\boldsymbol{u}_2)$ holds coordinatewise, i.e. Eq. (1.5).

($\Rightarrow$) If $T$ is linear, additivity holds. Comparing first coordinates of $T(\boldsymbol{u}_1 + \boldsymbol{u}_2) = T(\boldsymbol{u}_1) + T(\boldsymbol{u}_2)$ and cancelling the common linear terms leaves $b = 2b$, so $b = 0$. Comparing second coordinates and cancelling $6(x_1 + x_2)$ leaves $c(x_1+x_2)(y_1+y_2)(z_1+z_2) = c(x_1y_1z_1 + x_2y_2z_2)$; taking $\boldsymbol{u}_1 = \boldsymbol{u}_2 = (1, 1, 1)$ gives $8c = 2c$, so $c = 0$. Hence linearity forces $b = c = 0$. □

**Exercise 1.3.** Show that for any $A \in \mathbb{R}^{m\times n}$, the number of linearly independent rows equals the number of linearly independent columns (row rank equals column rank).

*Solution.* Let $r$ be the column rank, with $\boldsymbol{c}_1, \dots, \boldsymbol{c}_r$ a basis of $\mathrm{Col}(A)$ collected as the columns of $C \in \mathbb{R}^{m\times r}$. Each column $\boldsymbol{a}_j$ of $A$ lies in $\mathrm{Col}(A)$, so $\boldsymbol{a}_j = C\boldsymbol{f}_j$ for a coordinate vector $\boldsymbol{f}_j$; assembling the $\boldsymbol{f}_j$ as the columns of $F \in \mathbb{R}^{r\times n}$ gives the factorization $A = CF$. Reading this rowwise, row $i$ of $A$ is a combination of the $r$ rows of $F$, so the row rank is at most $r$. Applying the same argument to $A^\top$ gives column rank at most row rank. The two inequalities force equality, so $\mathrm{rank}(A) = \mathrm{rank}(A^\top)$. □

**Exercise 1.4.** Show that for any $A \in \mathbb{R}^{m\times n}$, $\mathrm{rank}(A^\top A) = \mathrm{rank}(A)$.

*Solution.* Both $A$ and $A^\top A$ have $n$ columns, so by rank–nullity it suffices to show $\mathrm{Nul}(A^\top A) = \mathrm{Nul}(A)$. If $A\boldsymbol{x} = \boldsymbol{0}$ then $A^\top A\boldsymbol{x} = \boldsymbol{0}$, giving $\mathrm{Nul}(A) \subseteq \mathrm{Nul}(A^\top A)$. Conversely, if $A^\top A\boldsymbol{x} = \boldsymbol{0}$, left-multiply by $\boldsymbol{x}^\top$: $0 = \boldsymbol{x}^\top A^\top A\boldsymbol{x} = \|A\boldsymbol{x}\|_2^2$, so $A\boldsymbol{x} = \boldsymbol{0}$, giving the reverse inclusion. Equal null spaces give equal dimension, and subtracting from $n$ yields $\mathrm{rank}(A^\top A) = \mathrm{rank}(A)$. This is exactly what guarantees the matrix $X^\top X$ in the normal equations of Chapter 2 is invertible whenever $X$ has full column rank. □

**Exercise 1.5.** Show that for a $2 \times 2$ matrix the determinant is a multilinear function of the rows: it is linear in each row when the other is held fixed.

*Solution.* Write the rows as $\boldsymbol{r}_1 = (a, b)$, $\boldsymbol{r}_2 = (c, d)$, so $\det = ad - bc$. Fixing $\boldsymbol{r}_2$ and varying the first row, $\det\left[\begin{smallmatrix} \alpha\boldsymbol{r}_1+\beta\boldsymbol{r}_1' \\ \boldsymbol{r}_2 \end{smallmatrix}\right] = (\alpha a + \beta a')d - (\alpha b + \beta b')c = \alpha(ad - bc) + \beta(a'd - b'c)$, which is $\alpha \det[\boldsymbol{r}_1; \boldsymbol{r}_2] + \beta \det[\boldsymbol{r}_1'; \boldsymbol{r}_2]$. By the symmetric computation it is also linear in the second row. Hence the determinant is multilinear in the rows. □

**Exercise 1.6.** Given $\det\left[\begin{smallmatrix} a & -d \\ p & -q \end{smallmatrix}\right] = 17$, compute $\det\left[\begin{smallmatrix} 2a+5p & 5q+2d \\ p & q \end{smallmatrix}\right]$.

*Solution.* The hypothesis is $\det\left[\begin{smallmatrix} a & -d \\ p & -q \end{smallmatrix}\right] = a(-q) - (-d)(p) = dp - aq = 17$. Expanding the target, $\det\left[\begin{smallmatrix} 2a+5p & 5q+2d \\ p & q \end{smallmatrix}\right] = (2a + 5p)q - (5q + 2d)p = 2aq + 5pq - 5pq - 2dp = 2aq - 2dp = -2(dp - aq) = -2(17) = \boxed{-34}$. The cross terms cancel because the determinant is multilinear (Exercise 1.5): the new matrix differs from a simpler one by row operations that scale the determinant predictably. □

**Exercise 1.7.** Let $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{v} \in \mathbb{R}^n$ be nonzero. Show the outer product $\boldsymbol{u}\boldsymbol{v}^\top$ has rank exactly 1, and identify its column space.

*Solution.* The $j$th column of $\boldsymbol{u}\boldsymbol{v}^\top$ is $v_j\boldsymbol{u}$, a scalar multiple of $\boldsymbol{u}$, so every column lies in $\mathrm{span}\{\boldsymbol{u}\}$ and $\mathrm{Col}(\boldsymbol{u}\boldsymbol{v}^\top) \subseteq \mathrm{span}\{\boldsymbol{u}\}$. Since $\boldsymbol{v} \neq \boldsymbol{0}$, some $v_j \neq 0$, making that column $v_j\boldsymbol{u} \neq \boldsymbol{0}$; hence the column space is exactly the line $\mathrm{span}\{\boldsymbol{u}\}$, which is one-dimensional, so $\mathrm{rank}(\boldsymbol{u}\boldsymbol{v}^\top) = 1$. This is why each term $\boldsymbol{a}_k\boldsymbol{b}_k^\top$ in Eq. (1.9) has rank one. □

**Exercise 1.8.** Prove the cyclic property Eq. (1.11): for $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times m}$, $\mathrm{tr}(AB) = \mathrm{tr}(BA)$.

*Solution.* Both $AB$ and $BA$ are square. Swapping the order of two finite sums,

$$\mathrm{tr}(AB) = \sum_{i=1}^{m} (AB)_{ii} = \sum_{i=1}^{m}\sum_{k=1}^{n} A_{ik}B_{ki} = \sum_{k=1}^{n}\sum_{i=1}^{m} B_{ki}A_{ik} = \sum_{k=1}^{n} (BA)_{kk} = \mathrm{tr}(BA).$$

□

**Exercise 1.9.** Let $X = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ and $\boldsymbol{y} = (1, 2, 0)^\top$. (a) Is there a $\boldsymbol{\theta}$ with $X\boldsymbol{\theta} = \boldsymbol{y}$ exactly? (b) If not, describe geometrically what least squares Eq. (1.14) will return.

*Solution.* (a) A solution exists iff $\boldsymbol{y} \in \mathrm{Col}(X)$ (Proposition 1.13). The columns $(1, 1, 1)^\top$ and $(0, 1, 2)^\top$ span a plane in $\mathbb{R}^3$. Matching $X\boldsymbol{\theta} = \boldsymbol{y}$ forces $\theta_1 = 1$ (row 1) and $\theta_2 = 1$ (row 2), but then row 3 needs $1 + 2 = 0$, false. So $\boldsymbol{y} \notin \mathrm{Col}(X)$ and there is no exact solution. (b) Least squares returns the $\boldsymbol{\theta}$ whose prediction $X\boldsymbol{\theta}$ is the orthogonal projection of $\boldsymbol{y}$ onto the plane $\mathrm{Col}(X)$, the closest reachable point. Solving the normal equations (Chapter 2) gives $\boldsymbol{\theta} = (\frac{3}{2}, -\frac{1}{2})^\top$, prediction $X\boldsymbol{\theta} = (\frac{3}{2}, 1, \frac{1}{2})^\top$, and residual $\boldsymbol{y} - X\boldsymbol{\theta} = (-\frac{1}{2}, 1, -\frac{1}{2})^\top$ orthogonal to both columns of $X$. □

**Exercise 1.10.** Let $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2\}$ with $\boldsymbol{b}_1 = (1, 1)$ and $\boldsymbol{b}_2 = (1, -1)$. (a) Show $B$ is a basis of $\mathbb{R}^2$. (b) Find the coordinates of $\boldsymbol{v} = (3, 1)$ in $B$, the scalars $(c_1, c_2)$ with $\boldsymbol{v} = c_1\boldsymbol{b}_1 + c_2\boldsymbol{b}_2$.

*Solution.* (a) Stack the vectors as columns: $\det\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right] = (1)(-1) - (1)(1) = -2 \neq 0$, so the two vectors are linearly independent, and two independent vectors span $\mathbb{R}^2$, hence $B$ is a basis. (b) The equation $c_1\boldsymbol{b}_1 + c_2\boldsymbol{b}_2 = \boldsymbol{v}$ is the linear system

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \implies \begin{cases} c_1 + c_2 = 3 \\ c_1 - c_2 = 1. \end{cases}$$

Adding the equations gives $2c_1 = 4$, so $c_1 = 2$ and then $c_2 = 1$. The coordinate vector is $[\boldsymbol{v}]_B = (2, 1)$, and indeed $2(1, 1) + 1(1, -1) = (3, 1)$. Because $B$ is orthogonal ($\boldsymbol{b}_1^\top \boldsymbol{b}_2 = 1 - 1 = 0$) the coordinates can also be read off by projection, $c_i = \dfrac{\boldsymbol{v}^\top \boldsymbol{b}_i}{\boldsymbol{b}_i^\top \boldsymbol{b}_i}$, giving $c_1 = \frac{3+1}{2} = 2$ and $c_2 = \frac{3-1}{2} = 1$ with no system to solve at all. That shortcut is exactly why orthogonal bases, and the Gram–Schmidt process of Example 1.5 that builds them, are worth the trouble. □

**Exercise 1.11.** Find the inverse of $A = \left[\begin{smallmatrix} 4 & 3 \\ 2 & 2 \end{smallmatrix}\right]$ using the closed-form $2 \times 2$ rule, and verify $AA^{-1} = I$.

*Solution.* For a $2 \times 2$ matrix $\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ the inverse is $\dfrac{1}{ad - bc}\left[\begin{smallmatrix} d & -b \\ -c & a \end{smallmatrix}\right]$: swap the diagonal, negate the off-diagonal, and divide by the determinant. Here $\det A = 4(2) - 3(2) = 2 \neq 0$, so $A$ is invertible and

$$A^{-1} = \frac{1}{2}\begin{bmatrix} 2 & -3 \\ -2 & 4 \end{bmatrix} = \begin{bmatrix} 1 & -\frac{3}{2} \\ -1 & 2 \end{bmatrix}.$$

Checking, $AA^{-1} = \left[\begin{smallmatrix} 4 & 3 \\ 2 & 2 \end{smallmatrix}\right]\left[\begin{smallmatrix} 1 & -\frac{3}{2} \\ -1 & 2 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 4-3 & -6+6 \\ 2-2 & -3+4 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. The determinant in the denominator is why a singular matrix ($\det = 0$) has no inverse: the formula would divide by zero, the algebraic shadow of the geometric fact that a zero-determinant map collapses area and cannot be undone. □

CHAPTER 2

# Least Squares, Projections, and the Pseudoinverse

*When you cannot reach the target, aim for the closest point you can reach.*

*the geometry of least squares*

**Roadmap**

Chapter 1 ended with a problem it could not solve: the regression system $X\boldsymbol{\theta} = \boldsymbol{y}$ usually has no exact solution, because the target $\boldsymbol{y}$ rarely lies in the column space of $X$. This chapter resolves it three ways at once. We minimize the squared error instead of demanding equality (the *algebra*), read the solution as an *orthogonal projection* of $\boldsymbol{y}$ onto $\mathrm{Col}(X)$ (the *geometry*), and recognize it as the maximum-likelihood estimate under Gaussian noise (the *statistics*). The normal equations, the projection matrix $P = AA^{\dagger}$, the Moore–Penrose pseudoinverse, ridge regularization for when the problem is ill-posed, and the practical $QR$ route all fall out of these three views. We close with why nobody forms these inverses directly at scale.

**Suggested reading:** Deisenroth et al. [5], Chapter 3 and Section 9.2; Strang [21] for the projection picture; Hastie et al. [11, Ch. 3] for ridge regression.

## 2.1 The least-squares problem

We study the general problem of solving $A\boldsymbol{x} = \boldsymbol{b}$ when no exact solution exists. To fix notation, $A \in \mathbb{R}^{m\times n}$ is a known matrix, $\boldsymbol{b} \in \mathbb{R}^m$ a known target, and $\boldsymbol{x} \in \mathbb{R}^n$ the unknown we solve for. This is exactly the regression problem of Section 1.11 under the dictionary

$$A = X \text{ (design matrix)}, \qquad \boldsymbol{x} = \boldsymbol{\theta} \text{ (parameters)}, \qquad \boldsymbol{b} = \boldsymbol{y} \text{ (targets)}, \tag{2.1}$$

so everything below applies verbatim to fitting a linear model. The reason an exact solution typically fails to exist bears repeating from Proposition 1.13: a solution to $A\boldsymbol{x} = \boldsymbol{b}$ exists if and only if $\boldsymbol{b} \in \mathrm{Col}(A)$, and when $m \gg n$ the column space is a thin $n$-dimensional (at most) sliver of $\mathbb{R}^m$ that a noisy target almost never lands in. The system is *overdetermined* and generally *inconsistent*.

*Intuition.* Picture the columns of $A$ as a small set of directions in a high-dimensional room. Every prediction $A\boldsymbol{x}$ is some combination of those directions, so the reachable predictions fill out a low-dimensional floor, the column space. The target $\boldsymbol{b}$ floats somewhere above the floor. We cannot reach it, but we can stand on the floor directly beneath it. That spot, the foot of the perpendicular, is the least-squares answer. The whole chapter is this one picture made precise.

The fix is to ask for the best approximation rather than equality. We choose $\boldsymbol{x}$ to minimize the squared Euclidean distance between the prediction $A\boldsymbol{x}$ and the target $\boldsymbol{b}$:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2. \tag{2.2}$$

The quantity $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{x}$ is the *residual*, and Eq. (2.2) minimizes $\|\boldsymbol{r}\|_2^2 = \sum_{i=1}^m r_i^2$, the sum of squared errors. The same solution will turn out to wear three hats, summarized in Table 2.1 and proved over the next several sections.

| View | Statement | Where |
|---|---|---|
| Algebra | $\boldsymbol{x}^\star$ solves the normal equations $A^\top A\boldsymbol{x} = A^\top \boldsymbol{b}$ | Section 2.2 |
| Geometry | $A\boldsymbol{x}^\star$ is the orthogonal projection of $\boldsymbol{b}$ onto $\mathrm{Col}(A)$ | Section 2.3 |
| Statistics | $\boldsymbol{x}^\star$ is the maximum-likelihood estimate under Gaussian noise | Section 2.6 |

**Table 2.1.** One solution, three readings. Each view makes a different property obvious: the algebra computes it, the geometry explains why it is best, and the statistics says when it is the *right* thing to compute.

## 2.2 The normal equations

The objective in Eq. (2.2) is a smooth function of $\boldsymbol{x}$, so we minimize it by setting its gradient to zero. Expand

$$\begin{aligned} f(\boldsymbol{x}) &= \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 = (A\boldsymbol{x} - \boldsymbol{b})^\top (A\boldsymbol{x} - \boldsymbol{b}) \\ &= \boldsymbol{x}^\top A^\top A\boldsymbol{x} - 2\boldsymbol{b}^\top A\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{b}, \end{aligned} \tag{2.3}$$

where we used $(A\boldsymbol{x})^\top \boldsymbol{b} = \boldsymbol{b}^\top A\boldsymbol{x}$ (a scalar equals its own transpose). The gradient of each term follows from the matrix-calculus identities developed in Chapter 8; for the symmetric matrix $S = A^\top A$ we have $\nabla_{\boldsymbol{x}}(\boldsymbol{x}^\top S\boldsymbol{x}) = 2S\boldsymbol{x}$ and $\nabla_{\boldsymbol{x}}(\boldsymbol{b}^\top A\boldsymbol{x}) = A^\top \boldsymbol{b}$, so

$$\nabla f(\boldsymbol{x}) = 2A^\top A\boldsymbol{x} - 2A^\top \boldsymbol{b}. \tag{2.4}$$

Setting $\nabla f(\boldsymbol{x}) = \boldsymbol{0}$ and dividing by 2 gives the *normal equations*.

**Theorem 2.1 (Normal equations).** Any minimizer $\boldsymbol{x}^\star$ of $\|A\boldsymbol{x} - \boldsymbol{b}\|_2^2$ satisfies

$$A^\top A\,\boldsymbol{x}^\star = A^\top \boldsymbol{b}. \tag{2.5}$$

If $A$ has full column rank ($\mathrm{rank}(A) = n$), then $A^\top A \in \mathbb{R}^{n\times n}$ is invertible and the minimizer is unique:

$$\boxed{\boldsymbol{x}^\star = (A^\top A)^{-1} A^\top \boldsymbol{b}.} \tag{2.6}$$

*Proof.* Stationarity of the convex quadratic $f$ gives Eq. (2.5) directly from Eq. (2.4). The function $f$ is convex because its Hessian $\nabla^2 f = 2A^\top A$ is positive semidefinite ($\boldsymbol{x}^\top A^\top A\boldsymbol{x} = \|A\boldsymbol{x}\|_2^2 \geq 0$), so a stationary point is a global minimum (Chapter 8 makes this precise). For uniqueness, recall from Exercise 1.4 that $\mathrm{rank}(A^\top A) = \mathrm{rank}(A)$; when $\mathrm{rank}(A) = n$ the $n \times n$ matrix $A^\top A$ has full rank and is therefore invertible, and Eq. (2.5) has the unique solution Eq. (2.6). ∎

The matrix appearing in Eq. (2.6) has a name.

**Definition 2.2 (Pseudoinverse, full-column-rank case).** For $A \in \mathbb{R}^{m\times n}$ with full column rank, the *Moore–Penrose pseudoinverse* is

$$A^\dagger = (A^\top A)^{-1} A^\top \in \mathbb{R}^{n\times m}, \tag{2.7}$$

so that the least-squares solution is $\boldsymbol{x}^\star = A^\dagger \boldsymbol{b}$. When $A$ is square and invertible, $A^\dagger = A^{-1}$, because $(A^\top A)^{-1} A^\top = A^{-1}(A^\top)^{-1} A^\top = A^{-1}$.

Section 2.10 gives the general definition that drops the full-rank assumption.

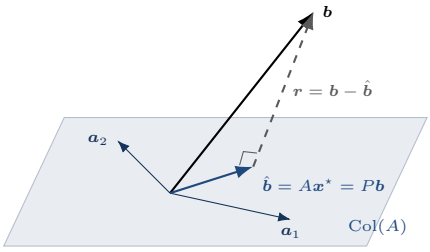



**Figure 2.1.** Least squares as orthogonal projection. The prediction $\hat{\boldsymbol{b}} = A\boldsymbol{x}^\star$ is the point of the plane $\mathrm{Col}(A)$ closest to the target $\boldsymbol{b}$. The residual $\boldsymbol{r} = \boldsymbol{b} - \hat{\boldsymbol{b}}$ meets the plane at a right angle, which is precisely the normal-equation condition $A^\top \boldsymbol{r} = \boldsymbol{0}$.

## 2.3 The geometry: orthogonal projection

The algebra of Eq. (2.5) has a clean picture. The reachable predictions $A\boldsymbol{x}$ sweep out the column space $\mathrm{Col}(A)$ as $\boldsymbol{x}$ varies. Minimizing $\|A\boldsymbol{x} - \boldsymbol{b}\|$ means finding the point of $\mathrm{Col}(A)$ closest to $\boldsymbol{b}$. In Euclidean geometry the closest point of a subspace to an external point is the foot of the perpendicular, the *orthogonal projection*.

The orthogonality is exactly the content of the normal equations. Rewriting Eq. (2.5) as $A^\top(\boldsymbol{b} - A\boldsymbol{x}^\star) = \boldsymbol{0}$, that is

$$A^\top \boldsymbol{r} = \boldsymbol{0}, \qquad \boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{x}^\star, \tag{2.8}$$

says that the residual $\boldsymbol{r}$ is orthogonal to every column of $A$, hence to all of $\mathrm{Col}(A)$. So the least-squares prediction $\hat{\boldsymbol{b}} = A\boldsymbol{x}^\star$ is the orthogonal projection of $\boldsymbol{b}$ onto $\mathrm{Col}(A)$, and the residual is what is left over in the orthogonal complement $\mathrm{Col}(A)^\perp$. This is the content of Fig. 2.1: the dashed residual is perpendicular to the shaded plane, and any other point of the plane would be reached by a longer (slanted) arrow from $\boldsymbol{b}$, by the Pythagorean theorem.

**Proposition 2.3 (Projection form of the solution).** When $A$ has full column rank, the least-squares prediction is

$$\hat{\boldsymbol{b}} = A\boldsymbol{x}^\star = A(A^\top A)^{-1}A^\top \boldsymbol{b} = AA^\dagger \boldsymbol{b}, \tag{2.9}$$

and $P = AA^\dagger = A(A^\top A)^{-1}A^\top$ is the matrix that projects any vector orthogonally onto $\mathrm{Col}(A)$.

## 2.4 Projection matrices

We pause to study projection matrices in their own right, since they reappear in principal component analysis (Chapter 11) and throughout.

**Definition 2.4 (Projection and orthogonal projection).** A matrix $P \in \mathbb{R}^{m\times m}$ is a *projection* (or *idempotent*) if

$$P^2 = P.$$

If in addition $P$ is symmetric, $P^\top = P$, then $P$ is an *orthogonal projection*.

Idempotence is the algebraic statement that projecting twice does nothing new: once a vector lands in the target subspace it stays put. Symmetry is what makes the projection drop *perpendicularly* rather than at an angle. The distinction matters: a generalized inverse (Section 2.9) gives an idempotent $AG$ that may be *oblique*, casting shadows at a slant, while only the symmetric case casts them straight down.

**Example 2.5 (Projection onto a line).** Let $\boldsymbol{u} \in \mathbb{R}^m$ be a unit vector, $\|\boldsymbol{u}\| = 1$, and set $P = \boldsymbol{u}\boldsymbol{u}^\top$. For any $\boldsymbol{v}$,

$$P\boldsymbol{v} = \boldsymbol{u}\boldsymbol{u}^\top \boldsymbol{v} = (\boldsymbol{u}^\top \boldsymbol{v})\,\boldsymbol{u},$$

which is the component of $\boldsymbol{v}$ along $\boldsymbol{u}$ (Fig. 2.2). It is a projection since $P^2 = \boldsymbol{u}(\boldsymbol{u}^\top\boldsymbol{u})\boldsymbol{u}^\top = \boldsymbol{u}(1)\boldsymbol{u}^\top = P$, and orthogonal since $P^\top = (\boldsymbol{u}\boldsymbol{u}^\top)^\top = \boldsymbol{u}\boldsymbol{u}^\top = P$. Its trace is $\mathrm{tr}(\boldsymbol{u}\boldsymbol{u}^\top) = \mathrm{tr}(\boldsymbol{u}^\top\boldsymbol{u}) = \|\boldsymbol{u}\|^2 = 1$, matching the one dimension it projects onto. Think of a flashlight shining straight down: $P\boldsymbol{v}$ is the shadow of $\boldsymbol{v}$ on the line through $\boldsymbol{u}$.

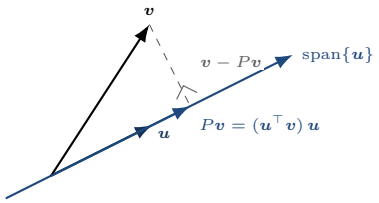



**Figure 2.2.** Projection onto a line. With $\boldsymbol{u}$ a unit vector, $P = \boldsymbol{u}\boldsymbol{u}^\top$ sends $\boldsymbol{v}$ to its shadow $(\boldsymbol{u}^\top \boldsymbol{v})\,\boldsymbol{u}$ on the line $\operatorname{span}\{\boldsymbol{u}\}$. The error $\boldsymbol{v} - P\boldsymbol{v}$ is perpendicular to the line. This rank-one $P$ is the simplest orthogonal projection and the building block of every other (Chapter 4).

**Proposition 2.6 (Properties of $P = AA^\dagger$).** Let $A$ have full column rank and $P = A(A^\top A)^{-1}A^\top$. Then:

(i) $P^2 = P$ and $P^\top = P$ (orthogonal projection);
(ii) $\operatorname{Range}(P) = \operatorname{Col}(A)$ and $\operatorname{Nul}(P) = \operatorname{Col}(A)^\perp$;
(iii) every eigenvalue of $P$ is 0 or 1, and $\operatorname{tr}(P) = \operatorname{rank}(A)$;
(iv) $I - P$ is the orthogonal projection onto $\operatorname{Col}(A)^\perp$, sending $\boldsymbol{b}$ to the residual $\boldsymbol{r}$.

*Proof.* (i) Idempotence: $P^2 = A(A^\top A)^{-1}A^\top A(A^\top A)^{-1}A^\top = A(A^\top A)^{-1}A^\top = P$, since the middle $A^\top A$ cancels its inverse. Symmetry: $(A^\top A)^{-1}$ is symmetric (the inverse of a symmetric matrix), so $P^\top = A\,(A^\top A)^{-\top}A^\top = A(A^\top A)^{-1}A^\top = P$. (ii) For any $\boldsymbol{b}$, $P\boldsymbol{b} = A\big[(A^\top A)^{-1}A^\top \boldsymbol{b}\big] \in \operatorname{Col}(A)$, and if $\boldsymbol{b} \in \operatorname{Col}(A)$, say $\boldsymbol{b} = A\boldsymbol{c}$, then $P\boldsymbol{b} = A(A^\top A)^{-1}A^\top A\boldsymbol{c} = A\boldsymbol{c} = \boldsymbol{b}$, so $P$ fixes $\operatorname{Col}(A)$ and $\operatorname{Range}(P) = \operatorname{Col}(A)$. If $\boldsymbol{b} \in \operatorname{Col}(A)^\perp$ then $A^\top \boldsymbol{b} = \boldsymbol{0}$, so $P\boldsymbol{b} = \boldsymbol{0}$; thus $\operatorname{Col}(A)^\perp \subseteq \operatorname{Nul}(P)$, and dimensions match by rank–nullity. (iii) If $P\boldsymbol{v} = \lambda\boldsymbol{v}$ with $\boldsymbol{v} \neq \boldsymbol{0}$, then $\lambda\boldsymbol{v} = P\boldsymbol{v} = P^2\boldsymbol{v} = \lambda^2\boldsymbol{v}$, forcing $\lambda^2 = \lambda$, so $\lambda \in \{0, 1\}$. The trace equals the sum of eigenvalues (Chapter 3), which counts the dimension of the eigenspace for $\lambda = 1$, namely $\dim \operatorname{Col}(A) = \operatorname{rank}(A)$. (iv) $(I-P)^2 = I - 2P + P^2 = I - P$ and $(I-P)^\top = I - P$, and $(I-P)\boldsymbol{b} = \boldsymbol{b} - \hat{\boldsymbol{b}} = \boldsymbol{r}$. ∎

**Example 2.7 (Projecting a vector onto a plane).** Take the same $A = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, whose two columns span a plane in $\mathbb{R}^3$. Its projection matrix (computed in Example 2.8) is

$$P = AA^\dagger = \tfrac{1}{6}\begin{bmatrix} 5 & 2 & -1 \\ 2 & 2 & 2 \\ -1 & 2 & 5 \end{bmatrix}.$$

Project the target $\boldsymbol{b} = (6, 0, 0)$ onto the plane:

$$\hat{\boldsymbol{b}} = P\boldsymbol{b} = \tfrac{1}{6}\begin{bmatrix} 5 & 2 & -1 \\ 2 & 2 & 2 \\ -1 & 2 & 5 \end{bmatrix}\begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix} = \tfrac{1}{6}\begin{bmatrix} 30 \\ 12 \\ -6 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \\ -1 \end{bmatrix}.$$

The residual is $\boldsymbol{r} = \boldsymbol{b} - \hat{\boldsymbol{b}} = (1, -2, 1)$, and it is orthogonal to the plane, as it must be: $A^\top \boldsymbol{r} = (1{-}2{+}1,\ 0{-}2{+}2)^\top = \boldsymbol{0}$. Projecting again changes nothing, $P\hat{\boldsymbol{b}} = \hat{\boldsymbol{b}}$ (the shadow of a point already on the floor is itself), confirming $P^2 = P$ on this vector. So the nearest point of the plane to $(6, 0, 0)$ is $(5, 2, -1)$, at distance $\|\boldsymbol{r}\| = \sqrt{6}$.

## 2.5 A worked least-squares fit

**Example 2.8 (Fitting a line through three points).** This completes Exercise 1.9. Take

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix},$$

which fits $y = \theta_1 + \theta_2 t$ to the data $(t, y) \in \{(0, 1), (1, 2), (2, 0)\}$ (the first column is the intercept, the second the input $t$). Form the pieces of the normal equations:

$$A^\top A = \begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix}, \qquad A^\top \boldsymbol{b} = \begin{bmatrix} 1+2+0 \\ 0+2+0 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}.$$

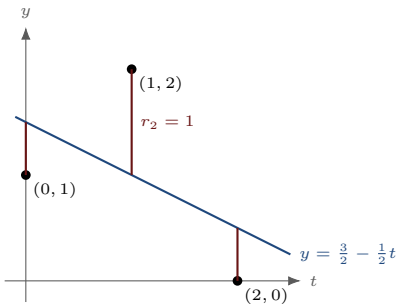



**Figure 2.3.** The same fit, the statistician's picture. The fitted line minimizes the total squared length of the vertical residual bars (red). This is the same $\boldsymbol{x}^\star$ as the projection in Fig. 2.1, seen in data space rather than in $\mathbb{R}^m$: minimizing vertical errors here is minimizing $\|A\boldsymbol{x} - \boldsymbol{b}\|_2$ there.

Since $\det(A^\top A) = 15 - 9 = 6 \neq 0$, we invert:

$$(A^\top A)^{-1} = \frac{1}{6}\begin{bmatrix} 5 & -3 \\ -3 & 3 \end{bmatrix}, \qquad A^\dagger = (A^\top A)^{-1}A^\top = \frac{1}{6}\begin{bmatrix} 5 & 2 & -1 \\ -3 & 0 & 3 \end{bmatrix}.$$

The least-squares parameters are

$$\boldsymbol{x}^\star = A^\dagger \boldsymbol{b} = \frac{1}{6}\begin{bmatrix} 5 & 2 & -1 \\ -3 & 0 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 9 \\ -3 \end{bmatrix} = \begin{bmatrix} 3/2 \\ -1/2 \end{bmatrix},$$

the fitted line $y = \frac{3}{2} - \frac{1}{2}t$ (Fig. 2.3). The prediction and residual are

$$\hat{\boldsymbol{b}} = A\boldsymbol{x}^\star = \begin{bmatrix} 3/2 \\ 1 \\ 1/2 \end{bmatrix}, \qquad \boldsymbol{r} = \boldsymbol{b} - \hat{\boldsymbol{b}} = \begin{bmatrix} -1/2 \\ 1 \\ -1/2 \end{bmatrix},$$

with sum of squared errors $\|\boldsymbol{r}\|_2^2 = \frac{1}{4} + 1 + \frac{1}{4} = \frac{3}{2}$, and one checks $A^\top \boldsymbol{r} = (-\frac{1}{2} + 1 - \frac{1}{2},\ 0 + 1 - 1)^\top = \boldsymbol{0}$, confirming Eq. (2.8): the residual is orthogonal to both columns of $A$. The projection matrix is

$$P = AA^\dagger = \frac{1}{6}\begin{bmatrix} 5 & 2 & -1 \\ 2 & 2 & 2 \\ -1 & 2 & 5 \end{bmatrix},$$

which is symmetric, satisfies $P^2 = P$, and has $\operatorname{tr}(P) = \frac{5+2+5}{6} = 2 = \operatorname{rank}(A)$, exactly as Proposition 2.6 predicts.

**Example 2.9 (Fitting a parabola: least squares is still linear).** Least squares fits any model that is linear *in the parameters*, even a curved one. To fit $y = \theta_1 + \theta_2 t + \theta_3 t^2$ to the four points $(-1, 1), (0, 0), (1, 1), (2, 3)$, put the powers of $t$ in the columns of the design matrix:

$$A = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}, \qquad \boldsymbol{y} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 3 \end{bmatrix}.$$

The normal equations $A^\top A\boldsymbol{\theta} = A^\top \boldsymbol{y}$ use

$$A^\top A = \begin{bmatrix} 4 & 2 & 6 \\ 2 & 6 & 8 \\ 6 & 8 & 18 \end{bmatrix}, \qquad A^\top \boldsymbol{y} = \begin{bmatrix} 5 \\ 6 \\ 14 \end{bmatrix},$$

where, for instance, $(A^\top A)_{33} = \sum_i t_i^4 = 1 + 0 + 1 + 16 = 18$ and $(A^\top \boldsymbol{y})_3 = \sum_i t_i^2 y_i = 1 + 0 + 1 + 12 = 14$. Solving the $3 \times 3$ system gives $\boldsymbol{\theta} = (0.15, -0.05, 0.75)$, the parabola $y = 0.15 - 0.05t + 0.75t^2$. Nothing about the method changed: the model is nonlinear in $t$ but linear in $\boldsymbol{\theta}$, so the same projection onto $\operatorname{Col}(A)$ applies. Adding more powers of $t$ just adds columns, the machinery behind polynomial regression and, with other basis functions, behind every linear-in-parameters model.

**Example 2.10 (Goodness of fit: the $R^2$ statistic).** A fit returns numbers, but how much of the data did the line actually explain? Fit the line $y = a + bx$ to the four points $(1,1),(2,2),(3,2),(4,3)$ and then measure quality. The design matrix and normal equations are

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 3 \end{bmatrix}, \qquad A^\top A = \begin{bmatrix} 4 & 10 \\ 10 & 30 \end{bmatrix}, \quad A^\top \boldsymbol{y} = \begin{bmatrix} 8 \\ 23 \end{bmatrix}.$$

The determinant is $4 \cdot 30 - 10 \cdot 10 = 20$, so Cramer's rule gives

$$a = \frac{8 \cdot 30 - 10 \cdot 23}{20} = \frac{10}{20} = 0.5, \qquad b = \frac{4 \cdot 23 - 10 \cdot 8}{20} = \frac{12}{20} = 0.6,$$

the line $\hat{y} = 0.5 + 0.6x$. Its fitted values are $(1.1, 1.7, 2.3, 2.9)$, leaving residuals $\boldsymbol{r} = (-0.1,\ 0.3,\ -0.3,\ 0.1)$ and a residual sum of squares $\mathrm{SS}_{\mathrm{res}} = \sum r_i^2 = 0.01 + 0.09 + 0.09 + 0.01 = 0.2$. Compare that against the spread of the data around its own mean $\bar{y} = 2$, the *total* sum of squares $\mathrm{SS}_{\mathrm{tot}} = \sum (y_i - \bar{y})^2 = 1 + 0 + 0 + 1 = 2$. The coefficient of determination is

$$R^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}} = 1 - \frac{0.2}{2} = 0.9.$$

The line explains 90% of the variation in $y$; the remaining 10% is scatter no line through these points can remove. The two anchors are worth memorizing: $R^2 = 1$ is a perfect fit with every residual zero, and $R^2 = 0$ means the line does no better than the flat guess $\hat{y} = \bar{y}$. Think of $\mathrm{SS}_{\mathrm{tot}}$ as the error you would make knowing only the average, $\mathrm{SS}_{\mathrm{res}}$ as the error left after using the line, and $R^2$ as the fraction of that ignorance the predictor clawed back.

## 2.6 Least squares as maximum likelihood

Why squared error, rather than absolute error or any other penalty? The squared norm is convenient (it has a linear gradient), but there is a deeper reason: least squares is the *maximum-likelihood* estimate when the noise is Gaussian. This is the statistical leg of Table 2.1, and it connects this chapter to the probability of Chapters 5 and 6.

Model each target as a linear function of its features corrupted by independent Gaussian noise,

$$y_i = \boldsymbol{a}_i^\top \boldsymbol{x} + \varepsilon_i, \qquad \varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2), \tag{2.10}$$

where $\boldsymbol{a}_i^\top$ is row $i$ of $A$. Then each $y_i$, given the parameters $\boldsymbol{x}$, is Gaussian with mean $\boldsymbol{a}_i^\top \boldsymbol{x}$ and variance $\sigma^2$, and because the noise terms are independent the likelihood of the whole dataset factorizes:

$$L(\boldsymbol{x}) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left( -\frac{(y_i - \boldsymbol{a}_i^\top \boldsymbol{x})^2}{2\sigma^2} \right). \tag{2.11}$$

Maximizing a product of exponentials is awkward, so we maximize its logarithm instead (the logarithm is increasing, so the maximizer is unchanged):

$$\log L(\boldsymbol{x}) = -\frac{m}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - \boldsymbol{a}_i^\top \boldsymbol{x})^2. \tag{2.12}$$

The first term does not involve $\boldsymbol{x}$, and the second is a negative constant times $\sum_i (y_i - \boldsymbol{a}_i^\top \boldsymbol{x})^2 = \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2$. Maximizing Eq. (2.12) over $\boldsymbol{x}$ is therefore *identical* to minimizing the sum of squared errors Eq. (2.2):

$$\arg\max_{\boldsymbol{x}} \ \log L(\boldsymbol{x}) \ = \ \arg\min_{\boldsymbol{x}} \ \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 \ = \ \boldsymbol{x}^\star. \tag{2.13}$$

*Intuition.* The choice of squared error is a choice of noise model. Squared error trusts that errors are Gaussian, which is the assumption that large deviations are rare and symmetric. Different assumptions give different fits: absolute error ($\ell_1$) corresponds to heavier-tailed Laplace noise and is more robust to outliers, while a Gaussian *prior* on $\boldsymbol{x}$ adds a penalty and gives ridge regression (Section 2.8). Least squares is the special, central case.

This reading also explains the success of least squares on real data: whenever the errors in a measurement are the sum of many small independent effects, the central limit theorem (Chapter 6) makes them approximately Gaussian, and least squares becomes the principled estimator. See Deisenroth et al. [5, §9.2] and Bishop [2, §3.1].

**Example 2.11 (One fit, three derivations that agree).** Nothing ties the course together like watching the same answer fall out of geometry, algebra, and probability. Fit a line $y = a+bx$ to the three points $(-1,1),(0,1),(1,4)$. The design matrix and its normal-equation pieces are

$$X = \begin{bmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 1 \\ 1 \\ 4 \end{bmatrix}, \qquad X^\top X = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}, \quad X^\top \boldsymbol{y} = \begin{bmatrix} 6 \\ 3 \end{bmatrix},$$

where $X^\top X$ is diagonal because the $x$-values are centered ($\sum x_i = 0$).

*View 1, the normal equations.* Solving $X^\top X\boldsymbol{\theta} = X^\top \boldsymbol{y}$ is immediate: $a = \frac{6}{3} = 2$ and $b = \frac{3}{2} = 1.5$, so $\hat{y} = 2 + 1.5x$.

*View 2, orthogonal projection.* The fitted values $\hat{\boldsymbol{y}} = X\boldsymbol{\theta} = (0.5, 2, 3.5)$ are the point of the plane $\mathrm{Col}(X)$ closest to $\boldsymbol{y}$. The residual $\boldsymbol{r} = \boldsymbol{y} - \hat{\boldsymbol{y}} = (0.5, -1, 0.5)$ is orthogonal to both columns of $X$: it sums to zero (orthogonal to the all-ones column) and $\sum x_i r_i = -0.5 + 0 + 0.5 = 0$ (orthogonal to the $x$ column), i.e. $X^\top \boldsymbol{r} = \mathbf{0}$. That orthogonality *is* the normal equations, read geometrically.

*View 3, Gaussian maximum likelihood.* Model $y_i = a + bx_i + \varepsilon_i$ with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ independent. The log-likelihood is $-\frac{1}{2\sigma^2}\sum_i (y_i - a - bx_i)^2$ plus a constant, so *maximizing* it is exactly *minimizing* the sum of squared residuals, the same objective whose gradient gives the same normal equations and hence the same $(2, 1.5)$. Least squares is Gaussian MLE in disguise.

*Coda, the Bayesian shrink.* Put a prior $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \tau^2 I)$ on the coefficients. The MAP estimate solves $(X^\top X + \lambda I)\boldsymbol{\theta} = X^\top \boldsymbol{y}$ with $\lambda = \sigma^2/\tau^2$, which is ridge regression (Section 2.8); at $\lambda = 1$ it returns $(1.5, 1.0)$, the same fit pulled gently toward the origin. Projection, the normal equations, maximum likelihood, and the Bayesian penalty are four windows onto one computation, which is why this single fit reappears in nearly every chapter that follows.

## 2.7 When the inverse fails: rank deficiency

Equation (2.6) assumed full column rank. Real feature matrices often violate this, because features can be linearly dependent (one column a combination of others), and then $A^\top A$ is singular.

**Example 2.12 (A rank-deficient design).** Let

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 5 \\ 10 \end{bmatrix}.$$

The second row is twice the first, so $\mathrm{rank}(A) = 1 < 2$. Then

$$A^\top A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 5 & 10 \\ 10 & 20 \end{bmatrix}, \qquad \det(A^\top A) = 100 - 100 = 0,$$

so $A^\top A$ is singular and Eq. (2.6) does not apply: the parameters are not unique (here $\boldsymbol{b} = 5\,\boldsymbol{a}_1$ exactly, so infinitely many $\boldsymbol{x}$ give zero error). What survives is the *prediction*: the projection $\hat{\boldsymbol{b}}$ onto $\mathrm{Col}(A)$ is still well defined and unique. Recovering a canonical $\boldsymbol{x}^\star$ in this case needs either the general pseudoinverse, built from the SVD in Chapter 4, or the regularization of the next section.

Once features are correlated, rank deficiency is common rather than exceptional. Two remedies follow. The first, regularization, changes the problem slightly so that a unique answer always exists. The second, the general pseudoinverse, keeps the problem and picks the most natural answer among the infinitely many.

## 2.8 Regularized least squares (ridge)

The cleanest fix for rank deficiency, and for the near-singular conditioning of Remark 1.20, is to add a penalty on the size of $\boldsymbol{x}$. *Ridge regression* (Tikhonov regularization) minimizes

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda \|\boldsymbol{x}\|_2^2, \qquad \lambda > 0, \tag{2.14}$$

trading a little fit for a smaller, more stable parameter vector. The scalar $\lambda$ sets the exchange rate.

**Proposition 2.13 (Ridge solution).** The objective Eq. (2.14) has the unique minimizer

$$\boxed{\boldsymbol{x}_\lambda = (A^\top A + \lambda I)^{-1} A^\top \boldsymbol{b},} \tag{2.15}$$

and $A^\top A + \lambda I$ is invertible for every $\lambda > 0$, regardless of the rank of $A$.

*Proof.* The gradient of Eq. (2.14) is $2A^\top(A\boldsymbol{x} - \boldsymbol{b}) + 2\lambda\boldsymbol{x}$; setting it to zero gives $(A^\top A + \lambda I)\boldsymbol{x} = A^\top \boldsymbol{b}$, which is Eq. (2.15) once we show the matrix is invertible. For any $\boldsymbol{v} \neq \boldsymbol{0}$,

$$\boldsymbol{v}^\top (A^\top A + \lambda I)\boldsymbol{v} = \|A\boldsymbol{v}\|_2^2 + \lambda\|\boldsymbol{v}\|_2^2 \;\geq\; \lambda\|\boldsymbol{v}\|_2^2 \;>\; 0,$$

so $A^\top A + \lambda I$ is positive definite, hence invertible. (Its eigenvalues are $\sigma_i^2 + \lambda \geq \lambda > 0$, where $\sigma_i$ are the singular values of $A$; adding $\lambda I$ lifts every eigenvalue away from zero, which is exactly what cures the ill-conditioning of Remark 1.20.) ∎

**Example 2.14 (Ridge restores a unique answer).** Apply ridge to the rank-deficient $A = \left[\begin{smallmatrix} 1 & 2 \\ 2 & 4 \end{smallmatrix}\right]$, $\boldsymbol{b} = (5, 10)^\top$ of Example 2.12, where ordinary least squares failed. With $A^\top A = \left[\begin{smallmatrix} 5 & 10 \\ 10 & 20 \end{smallmatrix}\right]$ and $A^\top \boldsymbol{b} = (25, 50)^\top$, take $\lambda = 5$:

$$A^\top A + 5I = \begin{bmatrix} 10 & 10 \\ 10 & 25 \end{bmatrix}, \quad (A^\top A + 5I)^{-1} = \frac{1}{150}\begin{bmatrix} 25 & -10 \\ -10 & 10 \end{bmatrix},$$

$$\boldsymbol{x}_5 = (A^\top A + 5I)^{-1} A^\top \boldsymbol{b} = \frac{1}{150}\begin{bmatrix} 25 & -10 \\ -10 & 10 \end{bmatrix}\begin{bmatrix} 25 \\ 50 \end{bmatrix} = \frac{1}{150}\begin{bmatrix} 125 \\ 250 \end{bmatrix} = \begin{bmatrix} 5/6 \\ 5/3 \end{bmatrix}.$$

A unique answer now exists. As $\lambda \to 0^+$ the ridge solution tends to $(1, 2)^\top$, which is precisely the minimum-norm solution the general pseudoinverse would return (Example 2.12); as $\lambda \to \infty$ it shrinks toward $\boldsymbol{0}$. Regularization interpolates between fitting the data and keeping the parameters small.

Ridge has a Bayesian reading we develop in Chapter 5: it is the *maximum a posteriori* estimate under a Gaussian prior $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \tau^2 I)$, with $\lambda = \sigma^2/\tau^2$. The penalty $\lambda\|\boldsymbol{x}\|^2$ is the prior's distrust of large parameters. Replacing the $\ell_2$ penalty with an $\ell_1$ penalty $\lambda\|\boldsymbol{x}\|_1$ gives the *lasso*, whose diamond-shaped constraint (recall Fig. 1.3) drives some parameters exactly to zero, producing sparse models [11, Ch. 3].

## 2.9 Generalized inverses

The other route through rank deficiency keeps the original problem and generalizes the inverse so that something always exists.

**Definition 2.15 (Generalized inverse).** A matrix $G \in \mathbb{R}^{n\times m}$ is a *generalized inverse* of $A \in \mathbb{R}^{m\times n}$ if

$$AGA = A. \tag{2.16}$$

The defining identity says that $G$ undoes $A$ wherever $A$ acts nontrivially: placed between two copies of $A$, it leaves $A$ unchanged. If $A$ is square and invertible then $G = A^{-1}$ works, since $AA^{-1}A = A$. The point of the definition is that a generalized inverse exists even when $A^{-1}$ does not.

**Proposition 2.16 (Existence by a full-rank block).** Every matrix has a generalized inverse. If, after row and column permutations, $A = \left[\begin{smallmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{smallmatrix}\right]$ with $A_{11} \in \mathbb{R}^{r\times r}$ invertible and $r = \operatorname{rank}(A)$, then the matrix built by inverting that block and padding with zeros, $G = \left[\begin{smallmatrix} A_{11}^{-1} & 0 \\ 0 & 0 \end{smallmatrix}\right]$ (placed at the right shape), satisfies $AGA = A$.

**Example 2.17 (Generalized inverses are not unique).** Let $A = [\,1\ 2\,] \in \mathbb{R}^{1\times 2}$ and seek $G = \left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] \in \mathbb{R}^{2\times 1}$. Then $AG = x + 2y$ (a scalar), and

$$AGA = (x + 2y)\begin{bmatrix} 1 & 2 \end{bmatrix} = \begin{bmatrix} x + 2y & 2x + 4y \end{bmatrix}.$$

Requiring $AGA = A = [\,1\ 2\,]$ gives $x + 2y = 1$ and $2x + 4y = 2$, the same equation twice. So *any* $(x, y)$ on the line $x + 2y = 1$ works:

$$G = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad G = \begin{bmatrix} 3 \\ -1 \end{bmatrix}, \quad G = \begin{bmatrix} 5 \\ -2 \end{bmatrix}, \quad \dots$$

A generalized inverse is generally not unique.

| | Condition | Name | What it forces |
|---|---|---|---|
| (i) | $AA^\dagger A = A$ | weak inverse | $A^\dagger$ is a generalized inverse |
| (ii) | $A^\dagger AA^\dagger = A^\dagger$ | reflexive | the relation is symmetric in $A, A^\dagger$ |
| (iii) | $(AA^\dagger)^\top = AA^\dagger$ | left projection orthogonal | $AA^\dagger$ projects onto $\mathrm{Col}(A)$ |
| (iv) | $(A^\dagger A)^\top = A^\dagger A$ | right projection orthogonal | $A^\dagger A$ projects onto $\mathrm{Row}(A)$ |

**Table 2.2.** The four Penrose conditions. Any generalized inverse satisfies (i); (ii)–(iv) single out the unique one whose two associated projections are both orthogonal, which is what makes $A^\dagger \boldsymbol{b}$ the minimum-norm least-squares solution.

A generalized inverse already gives a projection, just not necessarily an orthogonal one: if $AGA = A$ then $P = AG$ satisfies $P^2 = AGAG = (AGA)G = AG = P$, so $AG$ is idempotent and projects onto $\mathrm{Col}(A)$ (Exercise 2.1), but without symmetry the projection can be oblique. To pin down a unique, orthogonal projection we impose more conditions.

## 2.10 The Moore–Penrose pseudoinverse

**Definition 2.18 (Moore–Penrose pseudoinverse).** The *Moore–Penrose pseudoinverse* of $A \in \mathbb{R}^{m\times n}$ is the unique matrix $A^\dagger \in \mathbb{R}^{n\times m}$ satisfying the four *Penrose conditions*:

$$\text{(i) } AA^\dagger A = A, \quad \text{(ii) } A^\dagger AA^\dagger = A^\dagger, \quad \text{(iii) } (AA^\dagger)^\top = AA^\dagger, \quad \text{(iv) } (A^\dagger A)^\top = A^\dagger A. \tag{2.17}$$

The four conditions are read in Table 2.2. Such a matrix exists for *every* $A$ and is unique, which is why we speak of *the* pseudoinverse. The general construction comes from the SVD (Chapter 4); here we verify that in the full-column-rank case the formula of Definition 2.2 is the pseudoinverse.

**Proposition 2.19 (The formula satisfies the Penrose conditions).** If $A$ has full column rank, then $A^\dagger = (A^\top A)^{-1}A^\top$ satisfies all four conditions in Eq. (2.17).

*Proof.* Write $A^\dagger = (A^\top A)^{-1}A^\top$ and note $A^\dagger A = (A^\top A)^{-1}A^\top A = I_n$.

- (i) $AA^\dagger A = A(A^\dagger A) = AI_n = A$.
- (ii) $A^\dagger AA^\dagger = (A^\dagger A)A^\dagger = I_n A^\dagger = A^\dagger$.
- (iii) $(AA^\dagger)^\top = \big(A(A^\top A)^{-1}A^\top\big)^\top = A(A^\top A)^{-\top}A^\top = A(A^\top A)^{-1}A^\top = AA^\dagger$, using that $A^\top A$ is symmetric.
- (iv) $(A^\dagger A)^\top = I_n^\top = I_n = A^\dagger A$.

All four hold, so $(A^\top A)^{-1}A^\top$ is the Moore–Penrose pseudoinverse. ∎

Combining Proposition 2.19 with Proposition 2.3: in the full-rank case $P = AA^\dagger$ is the orthogonal projection onto $\mathrm{Col}(A)$, the least-squares prediction is $\hat{\boldsymbol{b}} = P\boldsymbol{b}$, and the parameters are $\boldsymbol{x}^\star = A^\dagger \boldsymbol{b}$. The pseudoinverse is the single object that ties the algebra (normal equations), the geometry (orthogonal projection), and the statistics (minimum squared error) together. When $A$ is rank-deficient the same $A^\dagger$ still exists (built from the SVD) and returns the *minimum-norm* solution among all minimizers, which is the answer ridge approached as $\lambda \to 0^+$.

**Example 2.20 (Weighted least squares: trusting some points more).** Sometimes the data points are not equally reliable. Fit the line $y = \theta_1 + \theta_2 t$ to $(0,0), (1,0), (2,6)$, but suppose the last point was measured far more carefully and deserves four times the weight. With $X = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, $\boldsymbol{y} = (0,0,6)$, and weights $W = \mathrm{diag}(1,1,4)$, the *weighted* normal equations replace $X^\top X$ and $X^\top \boldsymbol{y}$ by their weighted versions, $X^\top W X\,\boldsymbol{\theta} = X^\top W \boldsymbol{y}$, which minimizes $\sum_i w_i(y_i - \theta_1 - \theta_2 t_i)^2$.

*Unweighted first.* With $W = I$, $X^\top X = \left[\begin{smallmatrix} 3 & 3 \\ 3 & 5 \end{smallmatrix}\right]$, $X^\top \boldsymbol{y} = (6, 12)$, and solving gives $\boldsymbol{\theta}_{\mathrm{OLS}} = (-1, 3)$, the line $y = -1 + 3t$. Its prediction at $t = 2$ is 5, missing the measured 6 by 1.

*Now weighted.* Each sum picks up the weight $w_i$:

$$X^\top W X = \begin{bmatrix} \sum w_i & \sum w_i t_i \\ \sum w_i t_i & \sum w_i t_i^2 \end{bmatrix} = \begin{bmatrix} 6 & 9 \\ 9 & 17 \end{bmatrix}, \qquad X^\top W \boldsymbol{y} = \begin{bmatrix} \sum w_i y_i \\ \sum w_i t_i y_i \end{bmatrix} = \begin{bmatrix} 24 \\ 48 \end{bmatrix}.$$

The determinant is $6 \cdot 17 - 9^2 = 21$, so

$$\boldsymbol{\theta} = \frac{1}{21}\begin{bmatrix} 17 & -9 \\ -9 & 6 \end{bmatrix}\begin{bmatrix} 24 \\ 48 \end{bmatrix} = \frac{1}{21}\begin{bmatrix} -24 \\ 72 \end{bmatrix} = \left(-\tfrac{8}{7},\ \tfrac{24}{7}\right) \approx (-1.14,\ 3.43).$$

The weighted line $y = -\frac{8}{7} + \frac{24}{7}t$ predicts $\frac{40}{7} \approx 5.71$ at $t = 2$, much closer to the trusted value 6 (the miss shrinks from 1 to $\frac{2}{7}$). Weighting is like listening more to your most reliable instrument; it is exactly the maximum-likelihood fit when the noise variances differ across points (Exercise 2.6), with $w_i = 1/\sigma_i^2$.

## 2.11 Computing it in practice

The formula $A^\dagger = (A^\top A)^{-1}A^\top$ is the right thing to *understand* and the wrong thing to *compute*. Forming $A^\top A$ and inverting it is avoided in real systems for three reasons.

- **Conditioning.** Forming $A^\top A$ squares the condition number of $A$, so it amplifies any near-dependence among the columns (Remark 1.20) and makes the inverse numerically unstable.
- **Memory and time.** The matrix $A^\top A$ is $n \times n$, and a direct inverse costs $\mathcal{O}(n^3)$ time. For $n$ in the millions, as in modern models, the matrix does not even fit in memory.
- **Better tools exist.** The $QR$ factorization and the SVD (Chapter 4) solve least squares without ever forming $A^\top A$, and iterative methods such as conjugate gradients and stochastic gradient descent (Chapter 8) need only matrix–vector products $A\boldsymbol{x}$ and $A^\top \boldsymbol{y}$.

### 2.11.1 Solving by $QR$

The standard direct method factors $A$ (full column rank) into an orthonormal part and a triangular part,

$$A = QR, \qquad Q^\top Q = I_n, \quad R \text{ upper triangular}, \tag{2.18}$$

which the Gram–Schmidt process produces by orthonormalizing the columns of $A$ in turn. Substituting into the normal equations and using $Q^\top Q = I$ collapses them:

$$A^\top A\boldsymbol{x} = A^\top \boldsymbol{b} \implies R^\top Q^\top QR\,\boldsymbol{x} = R^\top Q^\top \boldsymbol{b} \implies R\boldsymbol{x} = Q^\top \boldsymbol{b},$$

a triangular system solved by back-substitution. No $A^\top A$ is ever formed, so the condition number is not squared.

**Example 2.21 ($QR$ on the worked fit).** For $A = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, Gram–Schmidt on the columns gives $\boldsymbol{q}_1 = \frac{1}{\sqrt{3}}(1,1,1)^\top$ and, after removing the $\boldsymbol{q}_1$-component from the second column, $\boldsymbol{q}_2 = \frac{1}{\sqrt{2}}(-1,0,1)^\top$, so

$$Q = \begin{bmatrix} 1/\sqrt{3} & -1/\sqrt{2} \\ 1/\sqrt{3} & 0 \\ 1/\sqrt{3} & 1/\sqrt{2} \end{bmatrix}, \qquad R = \begin{bmatrix} \sqrt{3} & \sqrt{3} \\ 0 & \sqrt{2} \end{bmatrix}.$$

With $\boldsymbol{b} = (1,2,0)^\top$ we get $Q^\top \boldsymbol{b} = (3/\sqrt{3},\ -1/\sqrt{2})^\top = (\sqrt{3}, -1/\sqrt{2})^\top$, and back-substitution in $R\boldsymbol{x} = Q^\top \boldsymbol{b}$ gives $\sqrt{2}\,x_2 = -1/\sqrt{2} \Rightarrow x_2 = -\frac{1}{2}$, then $\sqrt{3}\,x_1 + \sqrt{3}(-\frac{1}{2}) = \sqrt{3} \Rightarrow x_1 = \frac{3}{2}$. The answer $\boldsymbol{x}^\star = (\frac{3}{2}, -\frac{1}{2})^\top$ matches Example 2.8 exactly, obtained without inverting anything.

So the pseudoinverse is, in practice, a way of thinking. The clean expression $\boldsymbol{x}^\star = A^\dagger \boldsymbol{b}$ tells us what is being computed, namely the orthogonal projection of the target onto the span of the features, while the actual arithmetic is done by more stable means.

| Method | Forms $A^\top A$? | Cost | Use when |
|---|---|---|---|
| Normal equations | yes | $\mathcal{O}(mn^2+n^3)$ | small, well-conditioned $n$ |
| $QR$ | no | $\mathcal{O}(mn^2)$ | the dense default; stable |
| SVD | no | $\mathcal{O}(mn^2)$ | rank-deficient or ill-conditioned |
| SGD / iterative | no | $\mathcal{O}(mn)$ per step | $m, n$ huge; streaming data |

**Table 2.3.** Ways to solve least squares. The normal equations are for understanding; $QR$ is the workhorse for dense problems, the SVD handles the hard cases, and iterative methods scale to the sizes of modern machine learning [8, 24].

## 2.12 Summary

- When $A\boldsymbol{x} = \boldsymbol{b}$ has no solution, minimize $\|A\boldsymbol{x}-\boldsymbol{b}\|_2^2$ instead. Setting the gradient to zero gives the normal equations $A^\top A\boldsymbol{x}^\star = A^\top \boldsymbol{b}$.
- With full column rank, $A^\top A$ is invertible and $\boldsymbol{x}^\star = (A^\top A)^{-1}A^\top \boldsymbol{b} = A^\dagger \boldsymbol{b}$.
- Three views agree (Table 2.1): $\hat{\boldsymbol{b}} = A\boldsymbol{x}^\star = P\boldsymbol{b}$ is the orthogonal projection of $\boldsymbol{b}$ onto $\mathrm{Col}(A)$ (geometry), with residual $A^\top \boldsymbol{r} = \boldsymbol{0}$; and $\boldsymbol{x}^\star$ is the maximum-likelihood estimate under Gaussian noise (statistics).
- Orthogonal projections are exactly the symmetric idempotents, with eigenvalues in $\{0, 1\}$ and $\mathrm{tr}(P) = \mathrm{rank}(A)$.
- When $A^\top A$ is singular or ill-conditioned, ridge regression $(A^\top A + \lambda I)^{-1}A^\top \boldsymbol{b}$ always has a unique answer and limits to the minimum-norm solution as $\lambda \to 0^+$.
- A generalized inverse ($AGA = A$) always exists but is not unique; the Moore–Penrose pseudoinverse is the unique generalized inverse whose two projections are also orthogonal. The general construction waits for the SVD.
- Nobody forms $(A^\top A)^{-1}$ at scale; $QR$, SVD, and iterative solvers do the real work (Table 2.3).

## Exercises

**Exercise 2.1.** Let $A \in \mathbb{R}^{m\times n}$ and let $G \in \mathbb{R}^{n\times m}$ be a generalized inverse, so $AGA = A$. Show that $\mathrm{Col}(AG) = \mathrm{Col}(A)$, where $\mathrm{Col}(M)$ denotes the column space of $M$.

*Solution.* We show the two column spaces contain each other.

$\mathrm{Col}(AG) \subseteq \mathrm{Col}(A)$: any vector in $\mathrm{Col}(AG)$ has the form $AG\boldsymbol{w} = A(G\boldsymbol{w})$ for some $\boldsymbol{w}$, which is $A$ times a vector, hence lies in $\mathrm{Col}(A)$.

$\mathrm{Col}(A) \subseteq \mathrm{Col}(AG)$: any vector in $\mathrm{Col}(A)$ has the form $A\boldsymbol{z}$ for some $\boldsymbol{z}$. Using $AGA = A$,

$$A\boldsymbol{z} = (AGA)\boldsymbol{z} = (AG)(A\boldsymbol{z}),$$

which is $AG$ applied to the vector $A\boldsymbol{z}$, hence lies in $\mathrm{Col}(AG)$.

Both inclusions hold, so $\mathrm{Col}(AG) = \mathrm{Col}(A)$. Geometrically, $P = AG$ is a (possibly oblique) projection onto $\mathrm{Col}(A)$: it is idempotent, $P^2 = AGAG = (AGA)G = AG = P$, and its range is all of $\mathrm{Col}(A)$. □

**Exercise 2.2.** Let $A = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$. Compute $A^\top A$, the pseudoinverse $A^\dagger = (A^\top A)^{-1}A^\top$, and the projection matrix $P = AA^\dagger$. Verify that $P$ is symmetric, idempotent, and has trace equal to $\mathrm{rank}(A)$.

*Solution.* From Example 2.8, $A^\top A = \left[\begin{smallmatrix} 3 & 3 \\ 3 & 5 \end{smallmatrix}\right]$ with $\det = 6$, so $(A^\top A)^{-1} = \frac{1}{6}\left[\begin{smallmatrix} 5 & -3 \\ -3 & 3 \end{smallmatrix}\right]$ and

$$A^\dagger = (A^\top A)^{-1}A^\top = \frac{1}{6}\begin{bmatrix} 5 & 2 & -1 \\ -3 & 0 & 3 \end{bmatrix}, \qquad P = AA^\dagger = \frac{1}{6}\begin{bmatrix} 5 & 2 & -1 \\ 2 & 2 & 2 \\ -1 & 2 & 5 \end{bmatrix}.$$

The matrix $P$ is symmetric by inspection. It is idempotent because, by Proposition 2.6, any $AA^\dagger$ built from the full-rank formula satisfies $P^2 = P$ (direct multiplication confirms it). Its trace is $\frac{1}{6}(5+2+5) = 2$, which equals $\mathrm{rank}(A) = 2$

since the two columns of $A$ are independent. The two unit eigenvalues correspond to the two-dimensional $\mathrm{Col}(A)$, and the third eigenvalue is 0 for the one-dimensional $\mathrm{Col}(A)^{\perp}$. □

**Exercise 2.3.** Let $P$ be an orthogonal projection ($P^2 = P$, $P^\top = P$). Prove that $Q = I - P$ is also an orthogonal projection, that $PQ = QP = 0$, and that $Q$ projects onto $\mathrm{Col}(P)^{\perp}$.

*Solution.* *Idempotent:* $Q^2 = (I-P)^2 = I - 2P + P^2 = I - P = Q$. *Symmetric:* $Q^\top = (I-P)^\top = I - P^\top = I - P = Q$. So $Q$ is an orthogonal projection. *Orthogonality of the two:* $PQ = P(I-P) = P - P^2 = P - P = 0$, and likewise $QP = (I-P)P = P - P^2 = 0$. Finally, $Q\boldsymbol{v} = \mathbf{0} \iff (I-P)\boldsymbol{v} = \mathbf{0} \iff P\boldsymbol{v} = \boldsymbol{v} \iff \boldsymbol{v} \in \mathrm{Col}(P)$, so $\mathrm{Nul}(Q) = \mathrm{Col}(P)$ and therefore $\mathrm{Range}(Q) = \mathrm{Nul}(Q)^{\perp} = \mathrm{Col}(P)^{\perp}$. Thus $I - P$ projects onto the orthogonal complement; applied to a target $\boldsymbol{b}$ it returns exactly the least-squares residual $\boldsymbol{r} = (I-P)\boldsymbol{b}$. □

**Exercise 2.4.** Derive the normal equations Eq. (2.5) purely geometrically, without calculus, from the requirement that the residual $\boldsymbol{b} - A\boldsymbol{x}^\star$ be orthogonal to $\mathrm{Col}(A)$.

*Solution.* The prediction $A\boldsymbol{x}^\star$ is the closest point of $\mathrm{Col}(A)$ to $\boldsymbol{b}$ exactly when the error $\boldsymbol{b} - A\boldsymbol{x}^\star$ is orthogonal to the subspace $\mathrm{Col}(A)$. A vector is orthogonal to $\mathrm{Col}(A)$ if and only if it is orthogonal to each column $\boldsymbol{a}_j$ of $A$, i.e. $\boldsymbol{a}_j^\top(\boldsymbol{b} - A\boldsymbol{x}^\star) = 0$ for every $j$. Stacking these $n$ scalar equations is precisely

$$A^\top(\boldsymbol{b} - A\boldsymbol{x}^\star) = \mathbf{0} \quad\Longleftrightarrow\quad A^\top A\boldsymbol{x}^\star = A^\top\boldsymbol{b},$$

the normal equations. This recovers Theorem 2.1 from the projection picture of Fig. 2.1 alone, and it is the reason the calculus and the geometry agree. □

**Exercise 2.5.** For the ridge solution $\boldsymbol{x}_\lambda = (A^\top A + \lambda I)^{-1}A^\top\boldsymbol{b}$, show that $\|\boldsymbol{x}_\lambda\|_2$ is nonincreasing in $\lambda$ and that $\boldsymbol{x}_\lambda \to \mathbf{0}$ as $\lambda \to \infty$. Interpret both facts.

*Solution.* Use the SVD $A = U\Sigma V^\top$ (Chapter 4); the singular values are $\sigma_1, \dots, \sigma_n \geq 0$. A short computation gives $\boldsymbol{x}_\lambda = \sum_i \dfrac{\sigma_i}{\sigma_i^2 + \lambda}\,(\boldsymbol{u}_i^\top\boldsymbol{b})\,\boldsymbol{v}_i$, so

$$\|\boldsymbol{x}_\lambda\|_2^2 = \sum_i \left(\frac{\sigma_i}{\sigma_i^2 + \lambda}\right)^2 (\boldsymbol{u}_i^\top\boldsymbol{b})^2.$$

Each factor $\sigma_i/(\sigma_i^2 + \lambda)$ decreases in $\lambda$, so the whole sum does, proving $\|\boldsymbol{x}_\lambda\|_2$ is nonincreasing. As $\lambda \to \infty$ each factor $\to 0$, so $\boldsymbol{x}_\lambda \to \mathbf{0}$. Interpretation: larger $\lambda$ trusts the data less and shrinks the parameters toward the prior mean $\mathbf{0}$; the directions with small $\sigma_i$ (the ill-conditioned ones) are shrunk hardest, which is exactly the instability that regularization is meant to suppress. □

**Exercise 2.6.** Suppose the noise model Eq. (2.10) is changed so that observation $i$ has its own variance $\sigma_i^2$ (heteroscedastic noise). Show that the maximum-likelihood estimate is the *weighted* least-squares solution $\boldsymbol{x}^\star = (A^\top W A)^{-1}A^\top W\boldsymbol{b}$ with $W = \mathrm{diag}(1/\sigma_1^2, \dots, 1/\sigma_m^2)$.

*Solution.* With independent $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, the log-likelihood is $\log L(\boldsymbol{x}) = \text{const} - \tfrac{1}{2}\sum_i (y_i - \boldsymbol{a}_i^\top\boldsymbol{x})^2/\sigma_i^2$, mirroring Eq. (2.12) but with each squared error divided by its own variance. Maximizing is minimizing $\sum_i (y_i - \boldsymbol{a}_i^\top\boldsymbol{x})^2/\sigma_i^2 = (A\boldsymbol{x} - \boldsymbol{b})^\top W(A\boldsymbol{x} - \boldsymbol{b})$. Its gradient is $2A^\top W(A\boldsymbol{x} - \boldsymbol{b})$; setting it to zero gives $A^\top W A\boldsymbol{x} = A^\top W\boldsymbol{b}$, hence $\boldsymbol{x}^\star = (A^\top W A)^{-1}A^\top W\boldsymbol{b}$. Noisier observations (large $\sigma_i^2$) get small weight $1/\sigma_i^2$ and influence the fit less, which is the sensible thing to do. □

**Exercise 2.7.** Let $A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix}$ and $\boldsymbol{b} = (2, 0, 4)^\top$. Find the $QR$ factorization of $A$ by Gram–Schmidt and use it to solve the least-squares problem $R\boldsymbol{x} = Q^\top\boldsymbol{b}$ without forming $A^\top A$.

*Solution.* The first column is $(1,1,1)^\top$ with norm $\sqrt{3}$, so $\boldsymbol{q}_1 = \frac{1}{\sqrt{3}}(1,1,1)^\top$. The second column $\boldsymbol{a}_2 = (1,-1,1)^\top$ has $\boldsymbol{q}_1^\top \boldsymbol{a}_2 = (1-1+1)/\sqrt{3} = 1/\sqrt{3}$; removing that component, $\boldsymbol{a}_2 - \frac{1}{\sqrt{3}}\boldsymbol{q}_1 = (1,-1,1)^\top - \frac{1}{3}(1,1,1)^\top = \frac{1}{3}(2,-4,2)^\top$, with norm $\frac{1}{3}\sqrt{24} = \frac{2\sqrt{6}}{3}$, so $\boldsymbol{q}_2 = \frac{1}{\sqrt{6}}(1,-2,1)^\top$. Hence

$$R = \begin{bmatrix} \sqrt{3} & 1/\sqrt{3} \\ 0 & 2\sqrt{6}/3 \end{bmatrix}, \qquad Q^\top \boldsymbol{b} = \begin{bmatrix} \boldsymbol{q}_1^\top \boldsymbol{b} \\ \boldsymbol{q}_2^\top \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} 6/\sqrt{3} \\ (2-0+4)/\sqrt{6} \end{bmatrix} = \begin{bmatrix} 2\sqrt{3} \\ 6/\sqrt{6} \end{bmatrix}.$$

Back-substitution: $\frac{2\sqrt{6}}{3}x_2 = \frac{6}{\sqrt{6}} = \sqrt{6} \Rightarrow x_2 = \frac{3}{2}$; then $\sqrt{3}\,x_1 + \frac{1}{\sqrt{3}}\cdot\frac{3}{2} = 2\sqrt{3} \Rightarrow x_1 = 2 - \frac{1}{2} = \frac{3}{2}$. So $\boldsymbol{x}^\star = (\frac{3}{2}, \frac{3}{2})^\top$, the fit $y = \frac{3}{2} + \frac{3}{2}u$ where $u \in \{1,-1,1\}$ indexes the second column. One checks $A^\top(\boldsymbol{b} - A\boldsymbol{x}^\star) = \boldsymbol{0}$. □

**Exercise 2.8.** Fit a single constant $c$ to the three measurements $y = (2,5,11)$ by least squares: find the $c$ minimizing $\sum_i (y_i - c)^2$. What is the minimum value of the objective, and what familiar statistic is $c$?

*Solution.* This is the smallest least-squares problem of all, with design matrix $A = (1,1,1)^\top$ (a single column of ones) and the model $y_i \approx c$. The normal equation $A^\top A\, c = A^\top \boldsymbol{y}$ reads $3c = \sum_i y_i = 2+5+11 = 18$, so $c = 6$. The minimizer is the *sample mean*, which is the general fact that the least-squares constant fit is always $\bar{y}$: setting the derivative $\frac{d}{dc}\sum_i (y_i - c)^2 = -2\sum_i (y_i - c)$ to zero gives $c = \frac{1}{n}\sum_i y_i$. The minimum objective is the sum of squared deviations from the mean,

$$\sum_i (y_i - 6)^2 = (2-6)^2 + (5-6)^2 + (11-6)^2 = 16 + 1 + 25 = 42,$$

which is exactly the total sum of squares $\mathrm{SS}_{\mathrm{tot}}$ (the statistic defined in Example 2.10) and, divided by $n$, the variance of the data. Geometrically the column space of $A$ is the line through $(1,1,1)^\top$, and $\hat{\boldsymbol{y}} = (6,6,6)^\top$ is the orthogonal projection of $\boldsymbol{y}$ onto it: the mean is where the constant model touches the data cloud most closely. □

PART II

# Matrix Decompositions

CHAPTER 3

# Eigenvalues, Diagonalization, and the Spectral Theorem

*In the direction of an eigenvector, the matrix acts like scalar multiplication: it just stretches or compresses.*

*Professor Chin, on eigenvectors*

**Roadmap**

A matrix usually mixes the coordinates of a vector together. For a few special directions, the *eigenvectors*, it does something far simpler: it scales. This chapter finds those directions through the characteristic polynomial, measures how many independent ones each eigenvalue contributes (the gap between *algebraic* and *geometric* multiplicity), and determines exactly when the eigenvectors are plentiful enough to *diagonalize* the matrix. For symmetric matrices the answer is always yes, and better: the spectral theorem provides an *orthonormal* eigenbasis. That single fact powers principal component analysis, the geometry of covariance, the SVD of the next chapter, and the second-order tests of optimization.

**Suggested reading:** Deisenroth et al. [5], Sections 4.2–4.4; Strang [21] for the change-of-basis picture; Horn and Johnson [12] for the spectral theorem in full generality.

## 3.1 Eigenvalues and eigenvectors

**Definition 3.1 (Eigenvalue and eigenvector).** Let $A \in \mathbb{R}^{n\times n}$. A scalar $\lambda$ and a *nonzero* vector $\boldsymbol{v} \in \mathbb{R}^n$ (or $\mathbb{C}^n$) form an *eigenvalue–eigenvector pair* of $A$ if

$$A\boldsymbol{v} = \lambda \boldsymbol{v}, \qquad \boldsymbol{v} \neq \mathbf{0}. \tag{3.1}$$

The vector $\boldsymbol{v}$ is an *eigenvector* and $\lambda$ its *eigenvalue*.

Equation Eq. (3.1) says that along the direction $\boldsymbol{v}$ the map $A$ acts as pure scaling by $\lambda$: no rotation, no shear, just a stretch (if $|\lambda| > 1$), a compression ($0 < |\lambda| < 1$), a flip ($\lambda < 0$), or collapse to zero ($\lambda = 0$, which happens exactly when $A$ is singular). Most directions are warped by $A$; eigenvectors are the ones that come out pointing the way they went in.

## 3.2 The characteristic polynomial

To find eigenvalues, rewrite Eq. (3.1) as a homogeneous system,

$$A\boldsymbol{v} = \lambda \boldsymbol{v} \iff (A - \lambda I)\boldsymbol{v} = \mathbf{0}. \tag{3.2}$$

A nonzero solution $\boldsymbol{v}$ exists precisely when the matrix $A - \lambda I$ is singular, that is when its determinant vanishes.

**Definition 3.2 (Characteristic polynomial).** The *characteristic polynomial* of $A \in \mathbb{R}^{n\times n}$ is

$$p_A(\lambda) = \det(A - \lambda I), \tag{3.3}$$

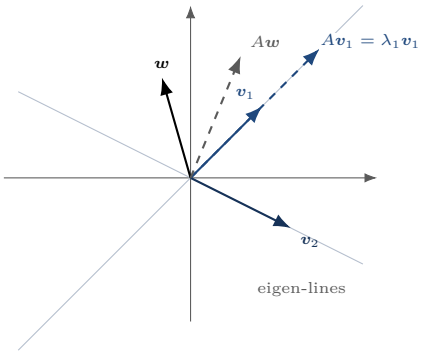



**Figure 3.1.** Eigenvectors are invariant directions. The map $A$ sends $\boldsymbol{v}_1$ to a multiple of itself, $A\boldsymbol{v}_1 = \lambda_1\boldsymbol{v}_1$, so $\boldsymbol{v}_1$ stays on its line; the same holds for $\boldsymbol{v}_2$. A generic vector $\boldsymbol{w}$ is rotated off its own line. The eigen-lines are the skeleton along which $A$ acts by simple scaling.

a polynomial of degree $n$ in $\lambda$. Its roots are the eigenvalues of $A$.

The equation $p_A(\lambda) = 0$ is the *characteristic equation*. By the fundamental theorem of algebra it has $n$ roots in $\mathbb{C}$ counted with multiplicity, so an $n \times n$ matrix has $n$ eigenvalues over the complex numbers, though some may be repeated or non-real.

**Example 3.3 (A $2 \times 2$ characteristic polynomial).** Let $A = \left[\begin{smallmatrix} 4 & 1 \\ 2 & 3 \end{smallmatrix}\right]$. Then

$$p_A(\lambda) = \det \begin{bmatrix} 4-\lambda & 1 \\ 2 & 3-\lambda \end{bmatrix} = (4-\lambda)(3-\lambda) - (1)(2) = \lambda^2 - 7\lambda + 10,$$

and $\lambda^2 - 7\lambda + 10 = (\lambda - 2)(\lambda - 5) = 0$ gives eigenvalues $\lambda_1 = 2$ and $\lambda_2 = 5$.

### 3.2.1 Finding the eigenvectors

Once an eigenvalue $\lambda$ is known, its eigenvectors are the nonzero vectors in $\mathrm{Nul}(A - \lambda I)$, found by solving $(A - \lambda I)\boldsymbol{v} = \boldsymbol{0}$.

**Example 3.4 (Eigenvector for $\lambda = 2$).** Continuing Example 3.3 with $\lambda = 2$,

$$A - 2I = \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix} \xrightarrow{\text{reduce}} \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 0 \end{bmatrix},$$

so $x = -\frac{1}{2}y$ and the eigenvectors are the nonzero multiples of $\boldsymbol{v} = \left[\begin{smallmatrix} -1 \\ 2 \end{smallmatrix}\right]$ (taking $y = 2$). Eigenvectors are never unique: any nonzero scalar multiple of $\boldsymbol{v}$ works, since $A(c\boldsymbol{v}) = cA\boldsymbol{v} = c\lambda\boldsymbol{v} = \lambda(c\boldsymbol{v})$. We often normalize to unit length.

## 3.3 Eigenspaces and multiplicity

The eigenvectors for a fixed $\lambda$, together with $\mathbf{0}$, form a subspace.

**Definition 3.5 (Eigenspace, algebraic and geometric multiplicity).** For an eigenvalue $\lambda$ of $A$, the *eigenspace* is $E_\lambda = \mathrm{Nul}(A - \lambda I)$. Its

- *algebraic multiplicity* $\mathrm{AM}(\lambda)$ is the number of times $\lambda$ is a root of $p_A$;
- *geometric multiplicity* $\mathrm{GM}(\lambda) = \dim E_\lambda$ is the number of independent eigenvectors for $\lambda$.

**Proposition 3.6 (Multiplicity inequality).** For every eigenvalue, $1 \le \mathrm{GM}(\lambda) \le \mathrm{AM}(\lambda)$.

The lower bound holds because a root of $p_A$ makes $A - \lambda I$ singular, so $E_\lambda$ contains a nonzero vector. The upper bound is proved by extending a basis of $E_\lambda$ to all of $\mathbb{R}^n$ and reading off that $(\lambda - t)^{\mathrm{GM}(\lambda)}$ divides $p_A(t)$. When $\mathrm{GM}(\lambda) < \mathrm{AM}(\lambda)$ for some $\lambda$, the matrix is short of eigenvectors and is called *defective* (Example 3.12).

## 3.4 A full $3 \times 3$ example

**Example 3.7 (Eigenstructure of an upper-triangular matrix).** Let

$$A = \begin{bmatrix} 3 & 1 & -2 \\ 0 & 5 & -1 \\ 0 & 0 & 3 \end{bmatrix}.$$

**Eigenvalues.** Since $A - \lambda I$ is upper triangular, its determinant is the product of the diagonal, so

$$p_A(\lambda) = (3-\lambda)(5-\lambda)(3-\lambda) = (3-\lambda)^2(5-\lambda).$$

Thus $\lambda = 3$ with $\mathrm{AM}(3) = 2$ and $\lambda = 5$ with $\mathrm{AM}(5) = 1$. (For any triangular matrix the eigenvalues are simply the diagonal entries.)

**Geometric multiplicity of $\lambda = 5$.** Reduce

$$A - 5I = \begin{bmatrix} -2 & 1 & -2 \\ 0 & 0 & -1 \\ 0 & 0 & -2 \end{bmatrix} \xrightarrow{\text{reduce}} \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

giving $z = 0$, $x = \frac{1}{2}y$, so $E_5 = \mathrm{span}\{(1,2,0)^\top\}$ and $\mathrm{GM}(5) = 1 = \mathrm{AM}(5)$.

**Geometric multiplicity of $\lambda = 3$.** Reduce

$$A - 3I = \begin{bmatrix} 0 & 1 & -2 \\ 0 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Solving $(A - 3I)\boldsymbol{v} = \mathbf{0}$ for $\boldsymbol{v} = (x, y, z)^\top$, the first two rows read $y - 2z = 0$ and $2y - z = 0$. Substituting $y = 2z$ into the second gives $4z - z = 3z = 0$, so $z = 0$, then $y = 0$, while $x$ is free. Hence $E_3 = \mathrm{span}\{(1,0,0)^\top\}$ is only one-dimensional, and

$$\mathrm{GM}(3) = 1 \;<\; 2 = \mathrm{AM}(3).$$

**Conclusion.** The eigenvalue $\lambda = 3$ is repeated twice but supplies only one independent eigenvector. The total number of independent eigenvectors is $1 + 1 = 2 < 3 = n$, so by Theorem 3.9 this matrix is *defective* and *not* diagonalizable, even though it is triangular and its eigenvalues were free to read off. The lesson is exactly the one Example 3.12 makes in two dimensions: a repeated eigenvalue is a warning rather than a verdict. You must compute GM by solving $(A - \lambda I)\boldsymbol{v} = \mathbf{0}$; the eigenvalues alone never settle diagonalizability.

> A common slip is to assume that because a triangular matrix hands you its eigenvalues for free, it must be diagonalizable. Example 3.7 shows otherwise. Triangularity makes the eigenvalues easy; it says nothing about whether enough eigenvectors exist.

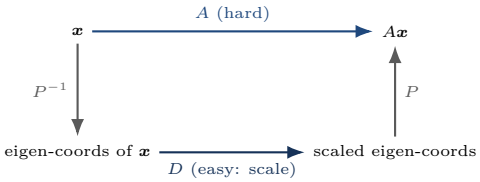



**Figure 3.2.** Diagonalization as a change of basis. Going directly across the top (applying $A$) is hard; the diagonal route through the eigen-coordinates is easy. The two paths agree, which is exactly the statement $A = PDP^{-1}$.

## 3.5 Diagonalization

**Definition 3.8 (Diagonalizable matrix).** A matrix $A \in \mathbb{R}^{n\times n}$ is *diagonalizable* if it is similar to a diagonal matrix: there exist an invertible $P$ and a diagonal $D$ with

$$A = PDP^{-1}, \qquad D = \operatorname{diag}(\lambda_1, \dots, \lambda_n), \qquad P = [\,\boldsymbol{v}_1 \ \cdots \ \boldsymbol{v}_n\,], \tag{3.4}$$

where the $\boldsymbol{v}_i$ are eigenvectors and the $\lambda_i$ the matching eigenvalues.

Reading $AP = PD$ column by column recovers $A\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$, so Eq. (3.4) holds exactly when $P$'s columns are $n$ independent eigenvectors. This gives the diagonalizability criterion.

*Intuition.* Read $A = PDP^{-1}$ right to left as three steps applied to a vector $\boldsymbol{x}$. First $P^{-1}$ rewrites $\boldsymbol{x}$ in the eigenvector basis, reporting how much of each eigenvector it contains. Then $D$ scales each of those coordinates by its eigenvalue, the one thing $A$ does simply. Finally $P$ translates back to the standard basis. A complicated map becomes "change to the right coordinates, stretch along the axes, change back." The whole difficulty of a matrix lives in the change of basis $P$; in the eigen-coordinates it is just a diagonal scaling.

**Theorem 3.9 (Diagonalizability criterion).** $A \in \mathbb{R}^{n\times n}$ is diagonalizable if and only if it has $n$ linearly independent eigenvectors, equivalently $\sum_\lambda \mathrm{GM}(\lambda) = n$, equivalently $\mathrm{GM}(\lambda) = \mathrm{AM}(\lambda)$ for every eigenvalue.

The chief payoff is that powers and polynomials of $A$ become trivial. From Eq. (3.4),

$$A^k = (PDP^{-1})^k = PD^kP^{-1}, \qquad D^k = \operatorname{diag}(\lambda_1^k, \dots, \lambda_n^k), \tag{3.5}$$

since the inner $P^{-1}P$ factors cancel. This is what makes diagonalization the tool of choice for linear recurrences, Markov chains (long-run behavior follows the dominant eigenvalue), and the matrix exponential $e^{At} = Pe^{Dt}P^{-1}$ in differential equations.

**Example 3.10 (Diagonalizing a $2\times 2$ matrix).** Return to $A = \left[\begin{smallmatrix} 4 & 1 \\ 2 & 3 \end{smallmatrix}\right]$ from Example 3.3, with eigenvalues 2 and 5. The eigenvector for $\lambda = 2$ is $(-1, 2)^\top$ (Example 3.4); for $\lambda = 5$, $(A - 5I)\boldsymbol{v} = \boldsymbol{0}$ gives $\left[\begin{smallmatrix} -1 & 1 \\ 2 & -2 \end{smallmatrix}\right]\boldsymbol{v} = \boldsymbol{0}$, i.e. $x = y$, so the eigenvector is $(1, 1)^\top$. Two distinct eigenvalues give two independent eigenvectors, so $A$ is diagonalizable with

$$P = \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} 2 & 0 \\ 0 & 5 \end{bmatrix}, \qquad A = PDP^{-1}.$$

Here $\det P = -3$, so $P^{-1} = -\frac{1}{3}\left[\begin{smallmatrix} 1 & -1 \\ -2 & -1 \end{smallmatrix}\right]$, and multiplying out $PDP^{-1}$ returns $\left[\begin{smallmatrix} 4 & 1 \\ 2 & 3 \end{smallmatrix}\right]$, as it must. Distinct eigenvalues always force independence of the eigenvectors, so a matrix with $n$ distinct eigenvalues is automatically diagonalizable; trouble can only arise at repeated eigenvalues, as in the next example.

**Example 3.11 (The payoff: computing $A^k$).** Take the same $A = \left[\begin{smallmatrix} 4 & 1 \\ 2 & 3 \end{smallmatrix}\right]$ with $P = \left[\begin{smallmatrix} -1 & 1 \\ 2 & 1 \end{smallmatrix}\right]$, $D = \operatorname{diag}(2, 5)$, and $P^{-1} = -\frac{1}{3}\left[\begin{smallmatrix} 1 & -1 \\ -2 & -1 \end{smallmatrix}\right]$. To cube it, we do not multiply $A$ by itself twice; we cube the *diagonal* and sandwich:

$$A^3 = PD^3P^{-1} = \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 8 & 0 \\ 0 & 125 \end{bmatrix} \left(-\tfrac{1}{3}\right) \begin{bmatrix} 1 & -1 \\ -2 & -1 \end{bmatrix}.$$

Working left to right, $PD^3 = \left[\begin{smallmatrix} -8 & 125 \\ 16 & 125 \end{smallmatrix}\right]$, and

$$A^3 = -\tfrac{1}{3} \begin{bmatrix} -8 & 125 \\ 16 & 125 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -2 & -1 \end{bmatrix} = -\tfrac{1}{3} \begin{bmatrix} -258 & -117 \\ -234 & -141 \end{bmatrix} = \begin{bmatrix} 86 & 39 \\ 78 & 47 \end{bmatrix}.$$

The direct route confirms it: $A^2 = \left[\begin{smallmatrix} 18 & 7 \\ 14 & 11 \end{smallmatrix}\right]$ and $A^3 = A^2A = \left[\begin{smallmatrix} 86 & 39 \\ 78 & 47 \end{smallmatrix}\right]$. For $k = 3$ the two routes cost about the same, but for large $k$ the diagonal route is decisive: $A^k = PD^kP^{-1}$ costs two matrix products and $n$ scalar $k$th powers, while repeated multiplication costs $k-1$ matrix products. The same trick gives the matrix exponential $e^{At} = Pe^{Dt}P^{-1}$ (differential equations) and the long-run behavior of Markov chains, where the largest $|\lambda_i|$ dominates $A^k$ as $k \to \infty$.

**Example 3.12 (A defective matrix).** Not every matrix is diagonalizable. Let $A = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$. Then $p_A(\lambda) = \det\left[\begin{smallmatrix} -\lambda & 1 \\ 0 & -\lambda \end{smallmatrix}\right] = \lambda^2$, so $\lambda = 0$ with $\mathrm{AM}(0) = 2$. But solving $A\boldsymbol{v} = \mathbf{0}$ gives $\left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] = \left[\begin{smallmatrix} y \\ 0 \end{smallmatrix}\right] = \mathbf{0}$, forcing $y = 0$ with $x$ free, so $E_0 = \mathrm{span}\{(1,0)^\top\}$ and $\mathrm{GM}(0) = 1$. Here $\mathrm{GM}(0) = 1 < 2 = \mathrm{AM}(0)$: there are not enough eigenvectors to span $\mathbb{R}^2$, so $A$ is *defective* and cannot be diagonalized.

**Example 3.13 (Complex eigenvalues: a rotation has none that are real).** A defect of repeated eigenvalues is one way real diagonalization fails; leaving the real line entirely is another. Take the matrix that rotates the plane by $30^\circ$,

$$A = \begin{bmatrix} \cos 30^\circ & -\sin 30^\circ \\ \sin 30^\circ & \cos 30^\circ \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}.$$

Its characteristic polynomial uses $\mathrm{tr}(A) = \sqrt{3}$ and $\det(A) = \frac{3}{4} + \frac{1}{4} = 1$:

$$p_A(\lambda) = \lambda^2 - \sqrt{3}\,\lambda + 1, \qquad \lambda = \frac{\sqrt{3} \pm \sqrt{3-4}}{2} = \frac{\sqrt{3} \pm i}{2}.$$

The discriminant $3 - 4 = -1$ is negative, so the eigenvalues are the complex conjugate pair $\lambda = \frac{\sqrt{3}}{2} \pm \frac{1}{2}i = \cos 30^\circ \pm i \sin 30^\circ = e^{\pm i30^\circ}$, and there is *no* real eigenvalue. That is exactly what geometry demands: a genuine rotation sends every nonzero real vector to a turned version of itself, so no real vector can satisfy $A\boldsymbol{v} = \lambda\boldsymbol{v}$ with real $\lambda$, because nothing is left pointing along its original line. The eigenvalues sit on the unit circle, $|\lambda| = 1$, which records that a rotation preserves length, and their angle is the rotation angle itself. The same matrix *is* diagonalizable over $\mathbb{C}$, with complex eigenvectors $(1, \mp i)$, and this is the everyday situation for the matrices behind oscillations and AC circuits: a complex-conjugate eigenpair $re^{\pm i\omega}$ encodes a rotation by $\omega$ scaled by $r$ each step, which is an undamped oscillation when $r = 1$, a decaying one when $r < 1$, and a growing one when $r > 1$.

## 3.6 Eigenvalues, trace, determinant, and rank

The eigenvalues carry the three scalar summaries of a matrix.

**Proposition 3.14 (Eigenvalue identities).** For $A \in \mathbb{R}^{n\times n}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$ (counted with algebraic multiplicity, over $\mathbb{C}$):

$$\mathrm{tr}(A) = \sum_{i=1}^{n} \lambda_i, \qquad \det(A) = \prod_{i=1}^{n} \lambda_i. \tag{3.6}$$

If $A$ is diagonalizable, $\mathrm{rank}(A)$ equals the number of nonzero eigenvalues.

*Proof.* Both identities read off coefficients of the characteristic polynomial. Factoring $p_A(\lambda) = \prod_i (\lambda_i - \lambda)$, the constant term is $p_A(0) = \det(A) = \prod_i \lambda_i$. Expanding $\det(A - \lambda I)$ instead, the coefficient of $\lambda^{n-1}$ equals $(-1)^{n-1}\,\mathrm{tr}(A)$ and also $(-1)^{n-1}\sum_i \lambda_i$, giving the trace identity. For the rank claim, if $A = PDP^{-1}$ then $\mathrm{rank}(A) = \mathrm{rank}(D)$ (multiplying by invertible $P, P^{-1}$ preserves rank), and a diagonal matrix has rank equal to its number of nonzero entries. ∎

A zero eigenvalue therefore signals singularity: $\det(A) = 0$, the null space is nontrivial, and $\mathrm{rank}(A) < n$.

**Example 3.15 (Rank from eigenvalues).** $A = \left[\begin{smallmatrix} 1 & 2 \\ 0 & 0 \end{smallmatrix}\right]$ is triangular with eigenvalues 1 and 0. So $\det(A) = 0$ (singular) and $\mathrm{rank}(A) = 1$, the count of nonzero eigenvalues.

| **Property** | **General** $A \in \mathbb{R}^{n\times n}$ | **Symmetric** $A^\top = A$ |
|---|---|---|
| Eigenvalues | possibly complex | always real |
| Eigenvectors | may be short (defective) | always a full set |
| Eigenvectors of distinct $\lambda$ | merely independent | orthogonal |
| Diagonalization | not guaranteed | always, orthogonally: $A = Q\Lambda Q^\top$ |
| Inverse of $P$ | compute $P^{-1}$ | free: $Q^{-1} = Q^\top$ |

**Table 3.1.** Why symmetry is special. Every entry in the right column is a strict improvement on the left, which is why symmetric matrices, covariance matrices, and Gram matrices are so much easier to work with than general ones.

**Application: PageRank as a dominant-eigenvector problem**

The original Google ranking treats the web as a Markov chain: a random surfer at a page follows one of its outgoing links uniformly at random. If $M$ is the column-stochastic *link matrix* ($M_{ij}$ is the probability of stepping from page $j$ to page $i$, so each column sums to 1), the long-run fraction of time spent at each page is the stationary distribution $\boldsymbol{\pi} = M\boldsymbol{\pi}$: the eigenvector of $M$ for eigenvalue 1. Importance is an eigenvector.
**A three-page web.** Let page 1 link to 2; page 2 link to 3; page 3 link to both 1 and 2. The link matrix and stationary equation are

$$M = \begin{bmatrix} 0 & 0 & \frac{1}{2} \\ 1 & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix}, \qquad \boldsymbol{\pi} = M\boldsymbol{\pi} \implies \pi_1 = \tfrac{1}{2}\pi_3, \quad \pi_2 = \pi_1 + \tfrac{1}{2}\pi_3, \quad \pi_3 = \pi_2.$$

Solving with $\pi_1 + \pi_2 + \pi_3 = 1$ gives $\boldsymbol{\pi} = (\frac{1}{5}, \frac{2}{5}, \frac{2}{5}) = (0.2, 0.4, 0.4)$: pages 2 and 3 are twice as important as page 1. One checks $M\boldsymbol{\pi} = \boldsymbol{\pi}$ directly. Because $M$ is a stochastic matrix its largest eigenvalue is exactly 1, so *power iteration* $\boldsymbol{v} \leftarrow M\boldsymbol{v}$ (Example 3.11's "largest eigenvalue dominates $A^k$" idea) converges to $\boldsymbol{\pi}$ from any start, which is how the ranking is actually computed on billions of pages. Real PageRank adds a small "teleportation" term $(1-\alpha)\frac{1}{n}\mathbf{1}\mathbf{1}^\top$ to guarantee a unique positive $\boldsymbol{\pi}$.

## 3.7 The spectral theorem for symmetric matrices

Symmetric matrices, the ones with $A^\top = A$, are the best behaved in all of linear algebra. They are never defective, and their eigenvectors can be chosen orthonormal. Table 3.1 contrasts them with general square matrices: the symmetric case removes every pathology of the general one.

**Theorem 3.16 (Spectral theorem).** Let $A \in \mathbb{R}^{n\times n}$ be symmetric, $A^\top = A$. Then

(i) all eigenvalues of $A$ are real;
(ii) eigenvectors for distinct eigenvalues are orthogonal;
(iii) there is an orthonormal basis of eigenvectors, i.e. an orthogonal matrix $Q$ ($Q^\top Q = QQ^\top = I$) and a diagonal $\Lambda$ with

$$A = Q\Lambda Q^\top = \sum_{i=1}^{n} \lambda_i \, \boldsymbol{q}_i \boldsymbol{q}_i^\top, \tag{3.7}$$

where $\boldsymbol{q}_i$ are the orthonormal eigenvectors and $\lambda_i$ the eigenvalues.

*Proof of (i) and (ii).* (i) Let $A\boldsymbol{v} = \lambda\boldsymbol{v}$ with $\boldsymbol{v} \neq \mathbf{0}$, possibly complex. Using the conjugate transpose and $A^\top = A$ real, $\lambda\,\overline{\boldsymbol{v}}^\top\boldsymbol{v} = \overline{\boldsymbol{v}}^\top A\boldsymbol{v} = (A\overline{\boldsymbol{v}})^\top\boldsymbol{v} = \overline{\lambda}\,\overline{\boldsymbol{v}}^\top\boldsymbol{v}$. Since $\overline{\boldsymbol{v}}^\top\boldsymbol{v} = \|\boldsymbol{v}\|^2 > 0$, we get $\lambda = \overline{\lambda}$, so $\lambda$ is real. (ii) If $A\boldsymbol{v}_i = \lambda_i\boldsymbol{v}_i$ and $A\boldsymbol{v}_j = \lambda_j\boldsymbol{v}_j$ with $\lambda_i \neq \lambda_j$, then $\lambda_i\,\boldsymbol{v}_i^\top\boldsymbol{v}_j = (A\boldsymbol{v}_i)^\top\boldsymbol{v}_j = \boldsymbol{v}_i^\top A\boldsymbol{v}_j = \lambda_j\,\boldsymbol{v}_i^\top\boldsymbol{v}_j$, so $(\lambda_i - \lambda_j)\boldsymbol{v}_i^\top\boldsymbol{v}_j = 0$ forces $\boldsymbol{v}_i^\top\boldsymbol{v}_j = 0$. Part (iii) follows by inducting on dimension within each eigenspace; Eq. (3.7) is just Eq. (3.4) with $P = Q$ orthogonal so that $P^{-1} = Q^\top$. ∎

The sum form in Eq. (3.7) is the *spectral decomposition*: it writes $A$ as a weighted sum of the rank-one orthogonal projectors $\boldsymbol{q}_i\boldsymbol{q}_i^\top$ (Example 2.5), each weight being its eigenvalue. Truncating the sum to the largest few $|\lambda_i|$ gives the best low-rank approximation of a symmetric matrix, the symmetric special case of the SVD story in Chapter 4.

*Intuition.* The spectral theorem is diagonalization with the change of basis made rigid. Because $Q$ is orthogonal, the change of coordinates $Q^\top$ is a pure rotation (or reflection), not a general skew. So a symmetric matrix acts as "rotate to the principal axes, scale along each axis by its eigenvalue, rotate back." Nothing is sheared. This is why symmetric matrices map spheres to axis-aligned ellipsoids (Fig. 3.3) and why their geometry is so clean.

**Example 3.17 (Spectral decomposition of a $2\times 2$ symmetric matrix).** Let $A = \left[\begin{smallmatrix} 0 & 2 \\ 2 & 0 \end{smallmatrix}\right]$ (symmetric). The characteristic polynomial is $\lambda^2 - 4$, so $\lambda_1 = 2$, $\lambda_2 = -2$. For $\lambda_1 = 2$, $(A - 2I)\boldsymbol{v} = \boldsymbol{0}$ gives $x = y$, normalized $\boldsymbol{q}_1 = \frac{1}{\sqrt{2}}(1,1)^\top$. For $\lambda_2 = -2$, $(A + 2I)\boldsymbol{v} = \boldsymbol{0}$ gives $x = -y$, normalized $\boldsymbol{q}_2 = \frac{1}{\sqrt{2}}(-1,1)^\top$. They are orthogonal, $\boldsymbol{q}_1^\top\boldsymbol{q}_2 = \frac{1}{2}(-1+1) = 0$, as the theorem promises. The spectral decomposition is

$$A = 2\,\boldsymbol{q}_1\boldsymbol{q}_1^\top + (-2)\,\boldsymbol{q}_2\boldsymbol{q}_2^\top = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix},$$

recovering $A$.

**Example 3.18 (A $3\times 3$ symmetric matrix with a repeated eigenvalue).** Let $A = \left[\begin{smallmatrix} 2 & 0 & 0 \\ 0 & 3 & 1 \\ 0 & 1 & 3 \end{smallmatrix}\right]$, symmetric. Its block structure makes the characteristic polynomial factor:

$$\det(A - \lambda I) = (2-\lambda)\big[(3-\lambda)^2 - 1\big] = (2-\lambda)(\lambda-2)(\lambda-4) = -(\lambda-2)^2(\lambda-4),$$

so $\lambda = 2$ is a *double* root and $\lambda = 4$ is simple. The crucial question is whether the repeated eigenvalue 2 supplies two independent eigenvectors. Solving $(A - 2I)\boldsymbol{v} = \boldsymbol{0}$, i.e. $\left[\begin{smallmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{smallmatrix}\right]\boldsymbol{v} = \boldsymbol{0}$, gives the single condition $v_2 + v_3 = 0$ with $v_1$ free, a *two*-dimensional eigenspace $E_2 = \operatorname{span}\{(1,0,0),(0,1,-1)\}$. So $\mathrm{GM}(2) = 2 = \mathrm{AM}(2)$ and the matrix is diagonalizable, unlike the defective triangular matrix of Example 3.7; a repeated eigenvalue is only a *warning*, and here the spectral theorem guarantees enough eigenvectors. For $\lambda = 4$, $(A - 4I)\boldsymbol{v} = \boldsymbol{0}$ gives $v_1 = 0$, $v_2 = v_3$, so $E_4 = \operatorname{span}\{(0,1,1)\}$. Orthonormalizing (the three are already mutually orthogonal),

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, \qquad \Lambda = \operatorname{diag}(2,2,4), \qquad A = Q\Lambda Q^\top.$$

The columns of $Q$ are an orthonormal eigenbasis, exactly what the spectral theorem promises for any symmetric matrix.

## 3.8 The Rayleigh quotient

The spectral theorem has a variational twin that we will lean on heavily in principal component analysis (Chapter 11). It answers the question: over all unit vectors, in which direction does a symmetric matrix stretch the most?

**Definition 3.19 (Rayleigh quotient).** For a symmetric $A \in \mathbb{R}^{n\times n}$ and $\boldsymbol{x} \neq \boldsymbol{0}$, the *Rayleigh quotient* is

$$R_A(\boldsymbol{x}) = \frac{\boldsymbol{x}^\top A\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}. \tag{3.8}$$

**Theorem 3.20 (Variational characterization of eigenvalues).** Let $A$ be symmetric with eigenvalues $\lambda_{\min} = \lambda_n \le \cdots \le \lambda_1 = \lambda_{\max}$. Then for every $\boldsymbol{x} \neq \boldsymbol{0}$,

$$\lambda_{\min} \;\le\; R_A(\boldsymbol{x}) \;\le\; \lambda_{\max},$$

the maximum is attained at a top eigenvector and the minimum at a bottom one:

$$\lambda_{\max} = \max_{\boldsymbol{x}\neq\boldsymbol{0}} R_A(\boldsymbol{x}) = \max_{\|\boldsymbol{x}\|=1} \boldsymbol{x}^\top A\boldsymbol{x}, \qquad \lambda_{\min} = \min_{\|\boldsymbol{x}\|=1} \boldsymbol{x}^\top A\boldsymbol{x}. \tag{3.9}$$

*Proof.* Diagonalize $A = Q\Lambda Q^\top$ and substitute $\boldsymbol{y} = Q^\top\boldsymbol{x}$, so $\|\boldsymbol{y}\| = \|\boldsymbol{x}\|$ because $Q$ is orthogonal. Then $\boldsymbol{x}^\top A\boldsymbol{x} = \boldsymbol{y}^\top\Lambda\boldsymbol{y} = \sum_i \lambda_i y_i^2$ and $\boldsymbol{x}^\top\boldsymbol{x} = \sum_i y_i^2$, so

$$R_A(\boldsymbol{x}) = \frac{\sum_i \lambda_i y_i^2}{\sum_i y_i^2}$$

is a weighted average of the eigenvalues with nonnegative weights $y_i^2$. A weighted average lies between the smallest and largest values being averaged, giving the bounds. Putting all the weight on the top eigenvalue (take $\boldsymbol{y} = \boldsymbol{e}_1$, i.e. $\boldsymbol{x} = \boldsymbol{q}_1$) attains $\lambda_{\max}$; $\boldsymbol{x} = \boldsymbol{q}_n$ attains $\lambda_{\min}$. ■

**Example 3.21 (Rayleigh extremes).** For $A = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ (eigenvalues 3 and 1, eigenvectors $\frac{1}{\sqrt{2}}(1,1)^\top$ and $\frac{1}{\sqrt{2}}(1,-1)^\top$), the Rayleigh quotient at the eigenvectors gives $R_A(1,1) = \frac{6}{2} = 3 = \lambda_{\max}$ and $R_A(1,-1) = \frac{2}{2} = 1 = \lambda_{\min}$. At the in-between direction $(1,0)$, $R_A(1,0) = 2$, comfortably between. No unit vector stretches more than the top eigenvector.

This is exactly the optimization that PCA solves: the first principal component is the unit direction maximizing $\boldsymbol{x}^\top C\boldsymbol{x}$ for the covariance matrix $C$, which by Theorem 3.20 is its top eigenvector (Chapter 11). The Rayleigh quotient is the hinge between the algebra of eigenvalues and the geometry of "directions of greatest variation."

**Enrichment: how eigenvalues are computed**

We find small eigenvalues by hand through the characteristic polynomial, but that is hopeless beyond $n = 4$ or so (there is no general formula for polynomial roots of degree $\geq 5$). Real software never forms $p_A$. The simplest practical idea is *power iteration*: pick a random $\boldsymbol{x}_0$ and repeat $\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k / \|A\boldsymbol{x}_k\|$. Writing $\boldsymbol{x}_0$ in the eigenbasis, each step multiplies the coefficient of $\boldsymbol{q}_i$ by $\lambda_i$, so the largest $|\lambda_i|$ dominates and $\boldsymbol{x}_k$ aligns with the top eigenvector; the Rayleigh quotient $R_A(\boldsymbol{x}_k)$ then converges to $\lambda_{\max}$. This is the seed of the algorithms that rank web pages and compute principal components at scale; production solvers refine it into the $QR$ algorithm [8, 24].

**Example 3.22 (Power iteration converging by hand).** Watch the method find the top eigenvalue of $A = \left(\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right)$, whose true eigenvalues are $\lambda = 3$ (eigenvector $(1,1)$) and $\lambda = 1$ (eigenvector $(1,-1)$). Start from $\boldsymbol{x}_0 = (1,0)$, which deliberately points along neither eigenvector. Each step applies $A$; normalization only rescales, so we track the raw iterates and report the Rayleigh quotient $R_A(\boldsymbol{x}) = \boldsymbol{x}^\top A\boldsymbol{x} / \boldsymbol{x}^\top\boldsymbol{x}$, which is the running estimate of $\lambda_{\max}$:

| $\boldsymbol{x}_k$ | $A\boldsymbol{x}_k$ | $R_A(\boldsymbol{x}_k)$ |
|---|---|---|
| $(1,0)$ | $(2,1)$ | $2/1 = 2.000$ |
| $(2,1)$ | $(5,4)$ | $14/5 = 2.800$ |
| $(5,4)$ | $(14,13)$ | $122/41 = 2.976$ |
| $(14,13)$ | $(41,40)$ | $1094/365 = 2.997$ |
| $(41,40)$ | $(122,121)$ | $2.99970$ |

Two things happen at once. The direction tilts toward $(1,1)$, since $41:40$ is nearly $1:1$, and the Rayleigh quotient climbs to 3. The convergence rate is governed by the ratio $|\lambda_2/\lambda_1| = 1/3$: the gap to 3 shrinks by roughly that factor each step, which is why the error falls from 1.0 to 0.2 to 0.024 to 0.003. A wider spectral gap means faster convergence, and a tiny gap means painfully slow progress, which is the practical reason production solvers add shifts and deflation rather than iterating naively. The same multiply-and-renormalize loop, run on the web's link matrix, is PageRank (Example 3.11 and the box above).

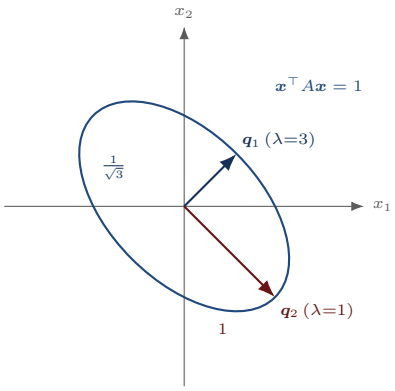



**Figure 3.3.** The quadratic form of $A = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$. The level set $\boldsymbol{x}^\top A\boldsymbol{x} = 1$ is an ellipse whose axes are the eigenvectors $\boldsymbol{q}_1, \boldsymbol{q}_2$ and whose semi-axes have length $1/\sqrt{\lambda_i}$. The steep direction $\boldsymbol{q}_1$ ($\lambda = 3$) gives the short axis $1/\sqrt{3}$; the gentle direction $\boldsymbol{q}_2$ ($\lambda = 1$) gives the long axis 1. Positive definiteness is exactly the statement that this level set is a bounded ellipse rather than an open hyperbola.

| Type | Notation | Eigenvalues | $\boldsymbol{x}^\top A\boldsymbol{x}$ for $\boldsymbol{x} \neq \mathbf{0}$ |
|---|---|---|---|
| Positive definite | $A \succ 0$ | all $\lambda_i > 0$ | always $> 0$ |
| Positive semidefinite | $A \succeq 0$ | all $\lambda_i \geq 0$ | always $\geq 0$ |
| Indefinite | none | mixed signs | both signs occur |
| Negative (semi)definite | $A \preceq 0$ | all $\lambda_i \leq 0$ | always $\leq 0$ |

**Table 3.2.** The definiteness types of a symmetric matrix, read off from the signs of its eigenvalues. Definiteness is the matrix version of the sign of a number, and it decides whether a critical point is a minimum, maximum, or saddle (Chapter 8).

## 3.9 Positive definite and positive semidefinite matrices

A recurring question is whether a symmetric matrix represents a "bowl" that curves upward in every direction. This is definiteness, and the spectral theorem makes it an eigenvalue condition.

**Definition 3.23 (Definiteness).** A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is

- *positive semidefinite* (PSD), written $A \succeq 0$, if $\boldsymbol{x}^\top A\boldsymbol{x} \geq 0$ for all $\boldsymbol{x}$;
- *positive definite* (PD), written $A \succ 0$, if $\boldsymbol{x}^\top A\boldsymbol{x} > 0$ for all $\boldsymbol{x} \neq \mathbf{0}$.

**Proposition 3.24 (Definiteness via eigenvalues).** A symmetric $A$ is PSD if and only if all its eigenvalues satisfy $\lambda_i \geq 0$, and PD if and only if all $\lambda_i > 0$.

*Proof.* Diagonalize $A = Q\Lambda Q^\top$ and substitute $\boldsymbol{y} = Q^\top\boldsymbol{x}$. Since $Q$ is orthogonal, $\boldsymbol{y}$ ranges over all of $\mathbb{R}^n$ as $\boldsymbol{x}$ does, and

$$\boldsymbol{x}^\top A\boldsymbol{x} = \boldsymbol{x}^\top Q\Lambda Q^\top\boldsymbol{x} = \boldsymbol{y}^\top\Lambda\boldsymbol{y} = \sum_{i=1}^{n} \lambda_i y_i^2.$$

This is nonnegative for all $\boldsymbol{y}$ iff every $\lambda_i \geq 0$, and strictly positive for $\boldsymbol{y} \neq \mathbf{0}$ iff every $\lambda_i > 0$. ∎

Definiteness has a clean geometric meaning, collected in Table 3.2. The level set $\{\boldsymbol{x} : \boldsymbol{x}^\top A\boldsymbol{x} = 1\}$ of a positive definite matrix is an ellipsoid whose axes are the eigenvectors and whose semi-axis lengths are $1/\sqrt{\lambda_i}$: large eigenvalues (steep curvature) give short axes, small eigenvalues (gentle curvature) give long ones (Fig. 3.3). This is the same ellipse that describes a Gaussian's contours of equal probability (Chapter 6) and the local bowl of a function at a minimum (Chapter 8).

Two facts we will reuse: for any matrix $B$, the Gram matrix $B^\top B$ is always PSD (because $\boldsymbol{x}^\top B^\top B\boldsymbol{x} = \|B\boldsymbol{x}\|^2 \geq 0$), which is why singular values are real and nonnegative (Chapter 4); and the Hessian of a function is PSD exactly

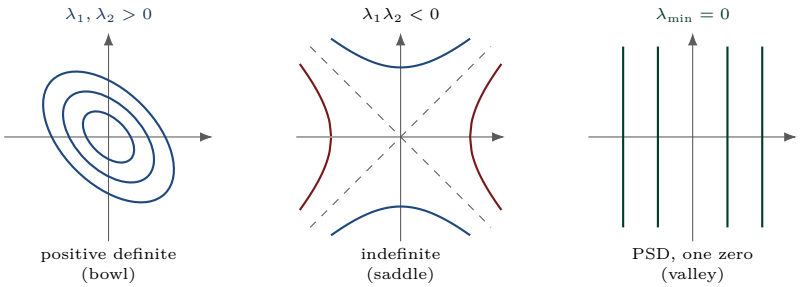



**Figure 3.4.** The three shapes of a quadratic form $\boldsymbol{x}^\top A\boldsymbol{x}$, fixed by the eigenvalue signs. Left: both positive, level sets are ellipses around a single lowest point (a bowl). Middle: opposite signs, level sets are hyperbolas straddling the asymptotes (a saddle). Right: a zero eigenvalue, level sets are parallel lines and the surface is flat along that direction (a valley). This is the second-derivative test read off the spectrum.

at the inputs where the function curves upward, the test for a local minimum (Chapter 8). Covariance matrices (Chapter 6) are PSD for the same Gram-matrix reason.

**Example 3.25 (Reading the shape of a quadratic form from the eigenvalues).** The sign pattern of the eigenvalues tells you the *shape* of the surface $z = \boldsymbol{x}^\top A\boldsymbol{x}$ without plotting anything. Classify three symmetric matrices.

*(a) Both eigenvalues positive.* For $A = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, the characteristic polynomial is $(2-\lambda)^2 - 1 = 0$, so $\lambda = 3, 1$, both positive. Expanding the form, $\boldsymbol{x}^\top A\boldsymbol{x} = 2x_1^2 + 2x_1x_2 + 2x_2^2 = (x_1+x_2)^2 + x_1^2 + x_2^2 > 0$ for $\boldsymbol{x} \neq \mathbf{0}$: a **bowl** opening upward, with elliptical level sets (Fig. 3.4, left). The matrix is positive definite.

*(b) Mixed signs.* For $A = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$ the eigenvalues are $+1$ and $-1$, and $\boldsymbol{x}^\top A\boldsymbol{x} = x_1^2 - x_2^2$ is positive along $x_1$ but negative along $x_2$: a **saddle**, neither a max nor a min. The form is *indefinite*; at $\boldsymbol{x} = (1, 0)$ it equals $+1$, at $(0, 1)$ it equals $-1$.

*(c) One zero eigenvalue.* For $A = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right]$ the eigenvalues are 1 and 0, and $\boldsymbol{x}^\top A\boldsymbol{x} = x_1^2 \geq 0$ but vanishes along the *whole* line $x_1 = 0$: a **flat-bottomed valley**, PSD but not PD. The zero eigenvalue is the flat direction.

Like reading a topographic map from a few numbers, the eigenvalue signs alone fix the terrain: all positive is a basin, mixed is a mountain pass, a zero is a level trough. This is exactly the second-derivative test in many variables (Chapter 8): a critical point is a minimum when its Hessian is positive definite, a saddle when the Hessian is indefinite.

## 3.10 Summary

- Eigenvectors are the directions $A$ merely scales: $A\boldsymbol{v} = \lambda\boldsymbol{v}$. The eigenvalues are the roots of $p_A(\lambda) = \det(A - \lambda I)$, and eigenvectors are found in $\mathrm{Nul}(A - \lambda I)$.
- Each eigenvalue has $1 \leq \mathrm{GM} \leq \mathrm{AM}$. The matrix is diagonalizable, $A = PDP^{-1}$, exactly when it has $n$ independent eigenvectors, i.e. GM = AM everywhere; otherwise it is defective.
- Diagonalization trivializes powers: $A^k = PD^kP^{-1}$. The trace is the sum and the determinant the product of the eigenvalues; rank counts the nonzero ones.
- The spectral theorem: a symmetric matrix has real eigenvalues and an orthonormal eigenbasis, $A = Q\Lambda Q^\top = \sum_i \lambda_i \boldsymbol{q}_i\boldsymbol{q}_i^\top$, acting as rotate–scale–rotate-back.
- The Rayleigh quotient $R_A(\boldsymbol{x}) = \boldsymbol{x}^\top A\boldsymbol{x}/\boldsymbol{x}^\top\boldsymbol{x}$ lies between $\lambda_{\min}$ and $\lambda_{\max}$, with extremes at the corresponding eigenvectors. This is the optimization PCA solves.
- $A \succeq 0$ iff all eigenvalues are $\geq 0$; $A \succ 0$ iff all are $> 0$. Gram matrices $B^\top B$, covariance matrices, and minimizing Hessians are all PSD; a PD form's level set $\boldsymbol{x}^\top A\boldsymbol{x} = 1$ is an ellipsoid with semi-axes $1/\sqrt{\lambda_i}$.

## Exercises

**Exercise 3.1.** Let $A = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$. Find its rank, trace, and determinant. Is $A$ positive definite or positive semidefinite? Finally, find an orthogonal diagonalization of $A$.

*Solution.* **Trace, determinant, rank.** $\operatorname{tr}(A) = 2 + 2 + 2 = 6$. The rows sum to zero, so $A(1,1,1)^\top = \mathbf{0}$: the all-ones vector is in the null space, hence $\lambda = 0$ is an eigenvalue, $\det(A) = 0$, and $A$ is singular with $\operatorname{rank}(A) \le 2$. The first two rows are independent, so $\operatorname{rank}(A) = 2$.

**Definiteness.** For any $\boldsymbol{x} = (x_1, x_2, x_3)^\top$, a direct expansion gives the sum-of-squares form

$$\boldsymbol{x}^\top A \boldsymbol{x} = (x_1 - x_2)^2 + (x_1 - x_3)^2 + (x_2 - x_3)^2 \ge 0,$$

so $A$ is positive *semi*definite. It is not positive definite, since the quadratic form vanishes on $\boldsymbol{x} = (1,1,1)^\top \neq \mathbf{0}$. (This $A$ is the graph Laplacian of a triangle.)

**Orthogonal diagonalization.** Because $\operatorname{tr}(A) = 6$ and one eigenvalue is 0, the remaining two sum to 6; solving $p_A(\lambda) = 0$ gives eigenvalues $0, 3, 3$. The eigenvector for $\lambda = 0$ is $\boldsymbol{q}_1 = \frac{1}{\sqrt{3}}(1,1,1)^\top$. The eigenspace for $\lambda = 3$ is the orthogonal complement of $(1,1,1)^\top$; an orthonormal basis of it is

$$\boldsymbol{q}_2 = \tfrac{1}{\sqrt{2}}(1,-1,0)^\top, \qquad \boldsymbol{q}_3 = \tfrac{1}{\sqrt{6}}(1,1,-2)^\top,$$

which one checks satisfy $A\boldsymbol{q}_2 = 3\boldsymbol{q}_2$, $A\boldsymbol{q}_3 = 3\boldsymbol{q}_3$, and are mutually orthonormal. With $Q = [\boldsymbol{q}_1\ \boldsymbol{q}_2\ \boldsymbol{q}_3]$ and $\Lambda = \operatorname{diag}(0,3,3)$, the spectral theorem gives $A = Q\Lambda Q^\top$. The nonnegative eigenvalues $0, 3, 3$ confirm $A \succeq 0$ via Proposition 3.24. □

**Exercise 3.2.** Let $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 \end{bmatrix}$. Show that $A^2 = A$. Use this to determine the eigenvalues of $A$, and find its eigendecomposition.

*Solution.* **Idempotence.** Direct multiplication gives $A^2 = A$ (the lower-right block $\begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$ squares to itself, and the 1 in the corner is unchanged). So $A$ is a projection.

**Eigenvalues.** If $A\boldsymbol{v} = \lambda\boldsymbol{v}$ then $A^2\boldsymbol{v} = \lambda^2\boldsymbol{v}$; but $A^2 = A$ gives $A^2\boldsymbol{v} = \lambda\boldsymbol{v}$, so $\lambda^2 = \lambda$ and every eigenvalue is 0 or 1. The number of 1's equals $\operatorname{tr}(A) = 1 + \frac{1}{2} + \frac{1}{2} = 2 = \operatorname{rank}(A)$, so the eigenvalues are $1, 1, 0$.

**Eigendecomposition.** The eigenvectors are found directly: $A\,(1,0,0)^\top = (1,0,0)^\top$ and $A\,(0,1,1)^\top = (0,1,1)^\top$ give $\lambda = 1$, while $A\,(0,1,-1)^\top = \mathbf{0}$ gives $\lambda = 0$. Hence, with

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}, \quad D = \operatorname{diag}(1,1,0), \qquad A = PDP^{-1}.$$

Since $A$ is symmetric, the eigenvectors can be normalized to an orthonormal set, making this an orthogonal diagonalization $A = Q\Lambda Q^\top$; $A$ is the orthogonal projection onto $\operatorname{span}\{(1,0,0)^\top, (0,1,1)^\top\}$. □

**Exercise 3.3.** Let $T$ be an $n \times n$ real matrix and $S$ an invertible $n \times n$ real matrix. (a) Prove that $T$ and $S^{-1}TS$ have the same eigenvalues. (b) What is the relationship between the eigenvectors of the two matrices?

*Solution.* (a) Suppose $\lambda$ is an eigenvalue of $S^{-1}TS$, so $(S^{-1}TS)\boldsymbol{v} = \lambda\boldsymbol{v}$ for some $\boldsymbol{v} \neq \mathbf{0}$. Multiply on the left by $S$: $TS\boldsymbol{v} = \lambda S\boldsymbol{v}$. Writing $\boldsymbol{w} = S\boldsymbol{v}$, this is $T\boldsymbol{w} = \lambda\boldsymbol{w}$, and $\boldsymbol{w} \neq \mathbf{0}$ because $S$ is invertible. So $\lambda$ is also an eigenvalue of $T$. The argument reverses (replace $S$ by $S^{-1}$), so $T$ and $S^{-1}TS$ have exactly the same eigenvalues. (Equivalently, similar matrices share the characteristic polynomial: $\det(S^{-1}TS - \lambda I) = \det(S^{-1}(T - \lambda I)S) = \det(T - \lambda I)$.) (b) The computation shows $\boldsymbol{v}$ is an eigenvector of $S^{-1}TS$ if and only if $S\boldsymbol{v}$ is an eigenvector of $T$ for the same eigenvalue: the change of basis $S$ maps one set of eigenvectors to the other. □

**Exercise 3.4.** Let $T \in \mathbb{R}^{n\times n}$ and $\lambda \in \mathbb{R}$. Prove there exists $\alpha \in \mathbb{R}$ with $|\alpha - \lambda| < \frac{1}{1000}$ such that $T - \alpha I$ is invertible.

*Solution.* The matrix $T - \alpha I$ is singular exactly when $\alpha$ is an eigenvalue of $T$, by Definition 3.2. But $T$ has at most $n$ distinct eigenvalues (the characteristic polynomial has degree $n$). The interval $(\lambda - \frac{1}{1000},\ \lambda + \frac{1}{1000})$ contains infinitely

many real numbers, so it certainly contains some $\alpha$ that is *not* one of the finitely many eigenvalues of $T$. For that $\alpha$, $T - \alpha I$ is nonsingular, i.e. invertible, and $|\alpha - \lambda| < \frac{1}{1000}$ as required. □

**Exercise 3.5.** Let $A$ be an invertible $n \times n$ matrix. (a) If $\lambda \neq 0$ is an eigenvalue of $A$, show $1/\lambda$ is an eigenvalue of $A^{-1}$. (b) Do $A$ and $A^{-1}$ have the same eigenvectors? (c) Must a real invertible matrix with $n > 2$ have a real eigenvalue? Explain how the answer depends on whether $n$ is odd or even.

*Solution.* (a) From $A\boldsymbol{v} = \lambda\boldsymbol{v}$ with $\boldsymbol{v} \neq \mathbf{0}$, multiply by $A^{-1}$: $\boldsymbol{v} = \lambda A^{-1}\boldsymbol{v}$, so $A^{-1}\boldsymbol{v} = \frac{1}{\lambda}\boldsymbol{v}$ (valid since $\lambda \neq 0$, which holds because $A$ is invertible and has no zero eigenvalue). Thus $1/\lambda$ is an eigenvalue of $A^{-1}$. (b) Yes. The computation in (a) shows the same $\boldsymbol{v}$ is an eigenvector of both $A$ and $A^{-1}$, only the eigenvalue is reciprocated. (c) The characteristic polynomial has real coefficients and degree $n$. If $n$ is *odd*, an odd-degree real polynomial has at least one real root, so a real eigenvalue must exist. If $n$ is *even*, it need not: for $n = 4$ the block-diagonal rotation $\left[\begin{smallmatrix} R & 0 \\ 0 & R \end{smallmatrix}\right]$ with $R = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$ is real and invertible but has only the complex eigenvalues $\pm i$. So the guarantee holds for odd $n$ and can fail for even $n$. □

**Exercise 3.6.** Let $A = \left[\begin{smallmatrix} 5 & 2 \\ 2 & 2 \end{smallmatrix}\right]$. (a) Find the unit vector maximizing $\boldsymbol{x}^{\top}A\boldsymbol{x}$ and the maximum value. (b) Find the minimizing unit vector and minimum value. (c) Sketch the level set $\boldsymbol{x}^{\top}A\boldsymbol{x} = 1$ and mark its axes.

*Solution.* The characteristic polynomial is $\lambda^2 - 7\lambda + (10 - 4) = \lambda^2 - 7\lambda + 6 = (\lambda - 6)(\lambda - 1)$, so $\lambda_{\max} = 6$ and $\lambda_{\min} = 1$. For $\lambda = 6$, $(A - 6I)\boldsymbol{v} = \left[\begin{smallmatrix} -1 & 2 \\ 2 & -4 \end{smallmatrix}\right]\boldsymbol{v} = \mathbf{0}$ gives $\boldsymbol{v} \propto (2, 1)^{\top}$, normalized $\boldsymbol{q}_1 = \frac{1}{\sqrt{5}}(2, 1)^{\top}$; for $\lambda = 1$, $\boldsymbol{q}_2 = \frac{1}{\sqrt{5}}(-1, 2)^{\top}$. (a) By Theorem 3.20 the maximum of $\boldsymbol{x}^{\top}A\boldsymbol{x}$ over unit vectors is $\lambda_{\max} = 6$, attained at $\boldsymbol{q}_1 = \frac{1}{\sqrt{5}}(2, 1)^{\top}$. (b) The minimum is $\lambda_{\min} = 1$, attained at $\boldsymbol{q}_2 = \frac{1}{\sqrt{5}}(-1, 2)^{\top}$. (c) The level set is an ellipse (the form is positive definite, both eigenvalues $> 0$) with axes along $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$; the semi-axis along $\boldsymbol{q}_1$ has length $1/\sqrt{6}$ and along $\boldsymbol{q}_2$ length $1/\sqrt{1} = 1$, so it is short in the steep $\boldsymbol{q}_1$ direction and long in the gentle $\boldsymbol{q}_2$ direction, exactly as in Fig. 3.3. □

**Exercise 3.7.** Let $A = \left[\begin{smallmatrix} 0 & 2 \\ 1 & 1 \end{smallmatrix}\right]$. (a) Find its eigenvalues and eigenvectors. (b) Write the diagonalization $A = PDP^{-1}$ explicitly, including $P^{-1}$.

*Solution.* (a) With $\operatorname{tr} A = 1$ and $\det A = -2$ the characteristic polynomial is

$$p_A(\lambda) = \lambda^2 - (\operatorname{tr} A)\lambda + \det A = \lambda^2 - \lambda - 2 = (\lambda - 2)(\lambda + 1),$$

so $\lambda_1 = 2$ and $\lambda_2 = -1$; the negative eigenvalue says $A$ flips orientation along one direction. For $\lambda = 2$, $(A - 2I)\boldsymbol{v} = \left[\begin{smallmatrix} -2 & 2 \\ 1 & -1 \end{smallmatrix}\right]\boldsymbol{v} = \mathbf{0}$ gives $v_1 = v_2$, so $\boldsymbol{v}_1 = (1, 1)$. For $\lambda = -1$, $(A + I)\boldsymbol{v} = \left[\begin{smallmatrix} 1 & 2 \\ 1 & 2 \end{smallmatrix}\right]\boldsymbol{v} = \mathbf{0}$ gives $v_1 = -2v_2$, so $\boldsymbol{v}_2 = (2, -1)$. The matrix is not symmetric, so these eigenvectors are not orthogonal, but they are independent, hence a valid eigenbasis. (b) Put the eigenvectors in $P$ and the eigenvalues in $D$:

$$P = \begin{bmatrix} 1 & 2 \\ 1 & -1 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & 0 \\ 0 & -1 \end{bmatrix}, \quad P^{-1} = \frac{1}{\det P}\begin{bmatrix} -1 & -2 \\ -1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & 2 \\ 1 & -1 \end{bmatrix},$$

since $\det P = (1)(-1) - (2)(1) = -3$. Multiplying back confirms it: $PD = \left[\begin{smallmatrix} 2 & -2 \\ 2 & 1 \end{smallmatrix}\right]$ and $PDP^{-1} = \frac{1}{3}\left[\begin{smallmatrix} 0 & 6 \\ 3 & 3 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0 & 2 \\ 1 & 1 \end{smallmatrix}\right] = A$. □

CHAPTER 4

# The Singular Value Decomposition

*A matrix is not a grid of numbers. It is a sum of concepts, each a direction of variation in your data.*

*Professor Chin, on the meaning of the SVD*

**Roadmap**

The spectral theorem of Chapter 3 diagonalized symmetric matrices. But data matrices are rectangular and never symmetric, so they have no eigenvectors at all. The *singular value decomposition* repairs this: *every* real matrix factors as $A = U\Sigma V^\top$ with $U, V$ orthogonal and $\Sigma$ diagonal and nonnegative. We construct it from the eigendecomposition of $A^\top A$, work a full example by hand, read it as a sum of rank-one pieces and as the image of a sphere becoming an ellipsoid, and prove the Eckart–Young theorem: truncating the SVD gives the best low-rank approximation there is. Along the way the SVD finally delivers the general pseudoinverse promised in Chapter 2 and sets up principal component analysis in Chapter 11.

**Suggested reading:** Deisenroth et al. [5], Section 4.5; Strang [21] for the four fundamental subspaces; Austin [1] for a gentle, picture-first tour.

## 4.1 Why every matrix needs the SVD

Eigenvalues require a square matrix, and a clean orthonormal eigenbasis requires a symmetric one (Theorem 3.16). A data matrix $A \in \mathbb{R}^{m\times n}$ with $m$ examples and $n$ features is neither: it is rectangular, so $A\boldsymbol{v} = \lambda\boldsymbol{v}$ does not even typecheck. Yet we still want to find the dominant directions of variation in the data. The singular value decomposition provides them for any matrix whatsoever, by separating the input directions (in $\mathbb{R}^n$) from the output directions (in $\mathbb{R}^m$) and the scaling between them.

**Theorem 4.1 (Singular value decomposition).** Every matrix $A \in \mathbb{R}^{m\times n}$ of rank $r$ can be written

$$A = U\Sigma V^\top, \tag{4.1}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal ($U^\top U = I_m$, $V^\top V = I_n$) and $\Sigma \in \mathbb{R}^{m\times n}$ is rectangular diagonal with entries

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0, \qquad \sigma_{r+1} = \cdots = 0,$$

the *singular values*. The columns $\boldsymbol{u}_i$ of $U$ are the *left singular vectors* and the columns $\boldsymbol{v}_i$ of $V$ are the *right singular vectors*.

Unlike eigenvalues, which can be negative or complex, the singular values are always real and nonnegative, and unlike the eigendecomposition, which fails for defective matrices (Example 3.12), the SVD exists for *every* matrix. That universality is why it sits at the center of numerical linear algebra and machine learning.

*Intuition.* The SVD says that *every* linear map, however complicated, is the same three moves in sequence: rotate (by $V^\top$), stretch the axes (by $\Sigma$), rotate again (by $U$). No shears, no skews, no surprises. The right singular vectors $\boldsymbol{v}_i$ are the special input directions that come out unrotated relative to each other; the map merely scales them by $\sigma_i$ and re-aims them along the $\boldsymbol{u}_i$. The spectral theorem (Theorem 3.16) was this story for symmetric matrices, where

the two rotations were forced to be the same; the SVD frees them, which is exactly what lets it handle rectangular and non-symmetric matrices that have no eigenvectors at all.

*Remark* 4.2 *(Full versus thin SVD).* When $m > n$, the bottom $m - n$ rows of $\Sigma$ are zero, so the last $m - n$ columns of $U$ multiply zeros. Dropping them gives the *thin* (or reduced) SVD $A = U_n \Sigma_n V^\top$ with $U_n \in \mathbb{R}^{m\times n}$, $\Sigma_n \in \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{n\times n}$. When the rank $r$ is smaller still, the trailing $n - r$ singular values vanish and one drops further to the *compact* SVD $A = U_r \Sigma_r V_r^\top$, which keeps only the $r$ directions that matter. The thin form is what one computes in practice; the full form is tidier for proofs.

## 4.2 Constructing the SVD from $A^\top A$

The recipe comes straight from the spectral theorem applied to the symmetric matrix $A^\top A$.

**Proposition 4.3 (Where the pieces come from).** Let $A \in \mathbb{R}^{m\times n}$. The matrix $A^\top A \in \mathbb{R}^{n\times n}$ is symmetric and positive semidefinite, so by Theorem 3.16 it has an orthonormal eigenbasis $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$ with real eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$. Then the SVD of $A$ is given by

$$\sigma_i = \sqrt{\lambda_i}, \qquad \boldsymbol{u}_i = \frac{1}{\sigma_i} A\boldsymbol{v}_i \quad (\sigma_i > 0), \tag{4.2}$$

with the $\boldsymbol{v}_i$ as the columns of $V$ and the $\boldsymbol{u}_i$ as the columns of $U$ (completed to an orthonormal basis of $\mathbb{R}^m$ by Gram–Schmidt where $\sigma_i = 0$).

*Proof.* That $A^\top A$ is symmetric is immediate, $(A^\top A)^\top = A^\top A$, and it is PSD because $\boldsymbol{x}^\top A^\top A\boldsymbol{x} = \|A\boldsymbol{x}\|^2 \geq 0$ (Proposition 3.24), so its eigenvalues $\lambda_i$ are nonnegative and $\sigma_i = \sqrt{\lambda_i}$ is well defined. For a nonzero singular value set $\boldsymbol{u}_i = A\boldsymbol{v}_i/\sigma_i$. These $\boldsymbol{u}_i$ are unit vectors,

$$\|\boldsymbol{u}_i\|^2 = \frac{1}{\sigma_i^2}\, \boldsymbol{v}_i^\top A^\top A\boldsymbol{v}_i = \frac{1}{\lambda_i}\, \boldsymbol{v}_i^\top (\lambda_i \boldsymbol{v}_i) = \frac{\lambda_i}{\lambda_i}\|\boldsymbol{v}_i\|^2 = 1,$$

and they are orthogonal, since for $i \neq j$

$$\boldsymbol{u}_i^\top \boldsymbol{u}_j = \frac{1}{\sigma_i\sigma_j} \boldsymbol{v}_i^\top A^\top A\boldsymbol{v}_j = \frac{\lambda_j}{\sigma_i\sigma_j} \boldsymbol{v}_i^\top \boldsymbol{v}_j = 0$$

because the $\boldsymbol{v}_i$ are orthonormal. Finally, $A\boldsymbol{v}_i = \sigma_i \boldsymbol{u}_i$ holds for every $i$ (trivially when $\sigma_i = 0$, since then $\|A\boldsymbol{v}_i\|^2 = \lambda_i = 0$). Collecting these as $AV = U\Sigma$ and right-multiplying by $V^\top$ (orthogonal, so $VV^\top = I$) gives $A = U\Sigma V^\top$. ∎

The same computation with $AA^\top$ instead of $A^\top A$ produces the left singular vectors directly: $AA^\top \boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i$. So $U$ holds the eigenvectors of $AA^\top$, $V$ holds the eigenvectors of $A^\top A$, and the two share the nonzero eigenvalues $\lambda_i = \sigma_i^2$.

**Computing the SVD by hand**

1. Form $A^\top A$ and find its eigenvalues $\lambda_i$ and orthonormal eigenvectors $\boldsymbol{v}_i$. These $\boldsymbol{v}_i$ are the columns of $V$.
2. Set the singular values $\sigma_i = \sqrt{\lambda_i}$, in decreasing order, on the diagonal of $\Sigma$.
3. For each $\sigma_i > 0$, set $\boldsymbol{u}_i = A\boldsymbol{v}_i/\sigma_i$. These are the first columns of $U$.
4. If $r < m$, extend $\{\boldsymbol{u}_i\}$ to an orthonormal basis of $\mathbb{R}^m$ by Gram–Schmidt to fill out $U$.

## 4.3 A worked example

**Example 4.4 (Full SVD of a $2 \times 2$ matrix).** Let $A = \left[\begin{smallmatrix} 3 & 0 \\ 4 & 5 \end{smallmatrix}\right]$.

**Step 1: $A^\top A$ and its eigenstructure.**

$$A^\top A = \begin{bmatrix} 3 & 4 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 25 & 20 \\ 20 & 25 \end{bmatrix}.$$

Its characteristic polynomial is $(25-\lambda)^2 - 20^2 = 0$, i.e. $25 - \lambda = \pm 20$, giving $\lambda_1 = 45$ and $\lambda_2 = 5$. The eigenvectors of the symmetric matrix $\left[\begin{smallmatrix} 25 & 20 \\ 20 & 25 \end{smallmatrix}\right]$ are $(1,1)$ for $\lambda_1$ (since equal coordinates) and $(1,-1)$ for $\lambda_2$; normalized,

$$\boldsymbol{v}_1 = \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{v}_2 = \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

**Step 2: singular values.** $\sigma_1 = \sqrt{45} = 3\sqrt{5}$ and $\sigma_2 = \sqrt{5}$, so $\Sigma = \left[\begin{smallmatrix} 3\sqrt{5} & 0 \\ 0 & \sqrt{5} \end{smallmatrix}\right]$.

**Step 3: left singular vectors $\boldsymbol{u}_i = A\boldsymbol{v}_i/\sigma_i$.**

$$A\boldsymbol{v}_1 = \tfrac{1}{\sqrt{2}} \begin{bmatrix} 3 \\ 9 \end{bmatrix}, \quad \boldsymbol{u}_1 = \frac{1}{3\sqrt{5}} \cdot \tfrac{1}{\sqrt{2}} \begin{bmatrix} 3 \\ 9 \end{bmatrix} = \tfrac{1}{\sqrt{10}} \begin{bmatrix} 1 \\ 3 \end{bmatrix},$$

$$A\boldsymbol{v}_2 = \tfrac{1}{\sqrt{2}} \begin{bmatrix} 3 \\ -1 \end{bmatrix}, \quad \boldsymbol{u}_2 = \frac{1}{\sqrt{5}} \cdot \tfrac{1}{\sqrt{2}} \begin{bmatrix} 3 \\ -1 \end{bmatrix} = \tfrac{1}{\sqrt{10}} \begin{bmatrix} 3 \\ -1 \end{bmatrix}.$$

Both are unit vectors and $\boldsymbol{u}_1^\top \boldsymbol{u}_2 = \frac{1}{10}(3-3) = 0$, as the construction guarantees.

**Step 4: assemble and verify.**

$$A = U\Sigma V^\top = \tfrac{1}{\sqrt{10}} \begin{bmatrix} 1 & 3 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 3\sqrt{5} & 0 \\ 0 & \sqrt{5} \end{bmatrix} \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Multiplying $U\Sigma = \frac{1}{\sqrt{2}} \left[\begin{smallmatrix} 3 & 3 \\ 9 & -1 \end{smallmatrix}\right]$ and then by $V^\top$ recovers $\left[\begin{smallmatrix} 3 & 0 \\ 4 & 5 \end{smallmatrix}\right]$, confirming the decomposition.

When $A$ is rectangular the only change is bookkeeping. For $A \in \mathbb{R}^{3\times 2}$, $\Sigma$ is $3 \times 2$ with the singular values on the leading diagonal and a zero bottom row, and the third column of $U$ is obtained by Gram–Schmidt as a unit vector orthogonal to $\boldsymbol{u}_1, \boldsymbol{u}_2$; it spans the part of $\mathbb{R}^3$ that $A$ cannot reach (Exercise 4.2).

**Example 4.5 (SVD of a wide matrix).** Let $A = \left[\begin{smallmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{smallmatrix}\right] \in \mathbb{R}^{2\times 3}$. Because $A$ is short and wide, work from the smaller Gram matrix $AA^\top \in \mathbb{R}^{2\times 2}$:

$$AA^\top = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

with eigenvalues 3 and 1, so $\sigma_1 = \sqrt{3}$, $\sigma_2 = 1$, and the left singular vectors are $\boldsymbol{u}_1 = \frac{1}{\sqrt{2}}(1,1)$, $\boldsymbol{u}_2 = \frac{1}{\sqrt{2}}(1,-1)$. The right singular vectors come from $\boldsymbol{v}_i = A^\top \boldsymbol{u}_i/\sigma_i$:

$$\boldsymbol{v}_1 = \frac{1}{\sqrt{3}} \cdot \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \tfrac{1}{\sqrt{6}}(1,2,1), \qquad \boldsymbol{v}_2 = \frac{1}{1} \cdot \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = \tfrac{1}{\sqrt{2}}(1,0,-1).$$

These are orthonormal, and a third right vector $\boldsymbol{v}_3 = \frac{1}{\sqrt{3}}(1,-1,1)$ spanning $\mathrm{Nul}(A)$ completes $V$. Then $A = U\Sigma V^\top$ with $\Sigma = [\,\mathrm{diag}(\sqrt{3}, 1) \mid \mathbf{0}\,]$ a $2 \times 3$ matrix. The lone zero "column" of $\Sigma$ is the null direction $\boldsymbol{v}_3$: $A\boldsymbol{v}_3 = \mathbf{0}$, which is why a wide matrix always has a nontrivial null space (here one-dimensional, since $3 - \mathrm{rank}(A) = 3 - 2 = 1$).

## 4.4 Two ways to read the SVD

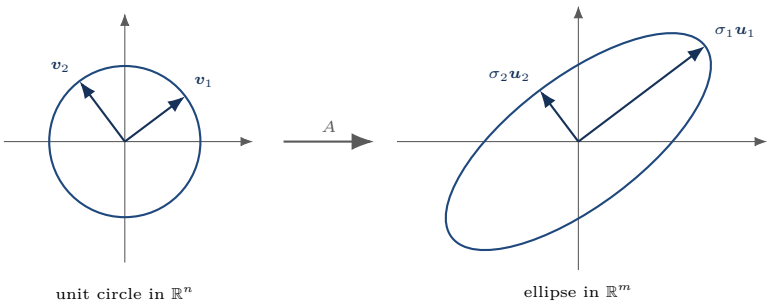



**Figure 4.1.** The geometry of the SVD. The right singular vectors $\boldsymbol{v}_i$ are the orthonormal directions in the input space that $A$ sends to the orthogonal directions $\boldsymbol{u}_i$ in the output space, stretched by the singular values: $A\boldsymbol{v}_i = \sigma_i \boldsymbol{u}_i$. The unit sphere maps to an ellipsoid whose semi-axis lengths are the singular values.

### 4.4.1 As a sum of rank-one matrices

Expanding $U\Sigma V^\top$ by the rank-one rule Eq. (1.9) writes $A$ as a weighted sum of outer products,

$$A = \sum_{i=1}^{r} \sigma_i \, \boldsymbol{u}_i \boldsymbol{v}_i^\top, \tag{4.3}$$

ordered so that $\sigma_1$ multiplies the most important rank-one piece. This is the exact analogue of the spectral decomposition Eq. (3.7); indeed for a symmetric PSD matrix the two coincide. Each term $\sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$ reads the input along $\boldsymbol{v}_i$, scales by $\sigma_i$, and writes the output along $\boldsymbol{u}_i$.

### 4.4.2 As a sphere becoming an ellipsoid

Because $V^\top$ is orthogonal it rotates (or reflects) the input without changing lengths, $\Sigma$ stretches the coordinate axes by the singular values, and $U$ rotates the result into the output space. The net effect of $A$ on the unit sphere is an ellipsoid (Fig. 4.1) whose axes point along the left singular vectors $\boldsymbol{u}_i$ and whose semi-axis lengths are the singular values $\sigma_i$. The largest singular value $\sigma_1$ is the largest factor by which $A$ can stretch any unit vector, which is exactly the matrix 2-norm $\|A\|_2 = \sigma_1$.

## 4.5 What the SVD reveals

The SVD is a Swiss-army knife: rank, norms, conditioning, the pseudoinverse, and the four fundamental subspaces all read straight off the factors. We collect them here.

### 4.5.1 The four fundamental subspaces

Split the singular vectors at the rank $r$. The first $r$ right singular vectors and the last $n - r$ partition the input space; the first $r$ left singular vectors and the last $m - r$ partition the output space:

$$\begin{aligned} \mathrm{Col}(A) &= \mathrm{span}\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r\}, & \mathrm{Nul}(A^\top) &= \mathrm{span}\{\boldsymbol{u}_{r+1}, \ldots, \boldsymbol{u}_m\}, \\ \mathrm{Row}(A) &= \mathrm{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r\}, & \mathrm{Nul}(A) &= \mathrm{span}\{\boldsymbol{v}_{r+1}, \ldots, \boldsymbol{v}_n\}. \end{aligned} \tag{4.4}$$

The map $A$ sends each row-space direction $\boldsymbol{v}_i$ to the column-space direction $\sigma_i \boldsymbol{u}_i$ ($i \le r$) and annihilates every null-space direction ($A\boldsymbol{v}_i = \boldsymbol{0}$ for $i > r$). So $A$ is a clean scaling isomorphism between $\mathrm{Row}(A)$ and $\mathrm{Col}(A)$, blind to the rest (Fig. 4.2). This is the *fundamental theorem of linear algebra*, and the SVD hands it to us with orthonormal bases for all four spaces at once.

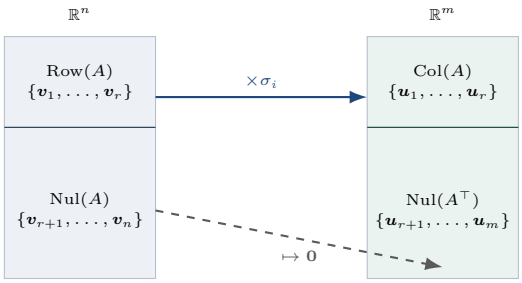



**Figure 4.2.** The four fundamental subspaces, read off the SVD. The map $A$ scales the row space onto the column space by the singular values and crushes the null space to zero. Least squares (Chapter 2) lives here: the residual is the part of the target in $\mathrm{Nul}(A^\top)$, the one piece $A$ cannot produce.

### 4.5.2 Rank, norms, and conditioning

Three scalar summaries of a matrix are singular values in disguise:

$$\operatorname{rank}(A) = \#\{i : \sigma_i > 0\}, \quad \|A\|_2 = \sigma_1, \quad \|A\|_F = \sqrt{\textstyle\sum_i \sigma_i^2}, \quad \kappa(A) = \frac{\sigma_1}{\sigma_r}. \tag{4.5}$$

The rank is the count of nonzero singular values (a far more honest test than a determinant, which only says zero-or-not). The spectral norm $\|A\|_2 = \sigma_1$ is the worst-case stretch; the Frobenius norm collects the total "energy"; and the *condition number* $\kappa(A) = \sigma_1/\sigma_r$ measures near-singularity, the quantity that made $A^\top A$ dangerous to invert in Remark 1.20. For the worked matrix $A = \left[\begin{smallmatrix} 3 & 0 \\ 4 & 5 \end{smallmatrix}\right]$ with $\sigma = (3\sqrt{5}, \sqrt{5})$, these are rank $= 2$, $\|A\|_2 = 3\sqrt{5} \approx 6.71$, $\|A\|_F = \sqrt{50} = 5\sqrt{2}$, and $\kappa = 3$, a well-conditioned matrix.

## 4.6 Low-rank approximation: the Eckart–Young theorem

Truncating the sum Eq. (4.3) after $k$ terms gives a rank-$k$ matrix

$$A_k = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top. \tag{4.6}$$

The remarkable fact is that no rank-$k$ matrix approximates $A$ better.

**Theorem 4.6 (Eckart–Young–Mirsky).** Among all matrices $B$ of rank at most $k$, the truncated SVD $A_k$ minimizes both the Frobenius and the spectral distance to $A$:

$$\min_{\operatorname{rank}(B)\le k} \|A - B\|_F = \|A - A_k\|_F = \sqrt{\sum_{i=k+1}^{r} \sigma_i^2}, \qquad \min_{\operatorname{rank}(B)\le k} \|A - B\|_2 = \sigma_{k+1}.$$

The error is governed entirely by the discarded singular values. When they decay quickly, a small $k$ captures almost all of $A$, which is the mathematical reason compression and denoising work: the dominant singular components carry the signal, and the small ones carry redundancy and noise. The decay is usually displayed as a *scree plot* (Fig. 4.3); the "elbow" where it flattens is the natural place to truncate.

**Example 4.7 (Best rank-one approximation).** For $A = \left[\begin{smallmatrix} 3 & 0 \\ 4 & 5 \end{smallmatrix}\right]$ of Example 4.4, the best rank-one approximation is

$$A_1 = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top = 3\sqrt{5} \cdot \tfrac{1}{\sqrt{10}} \begin{bmatrix} 1 \\ 3 \end{bmatrix} \cdot \tfrac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.5 \\ 4.5 & 4.5 \end{bmatrix}.$$

The approximation error is $\|A - A_1\|_F = \sigma_2 = \sqrt{5}$, exactly the singular value we dropped, in agreement with Theorem 4.6.

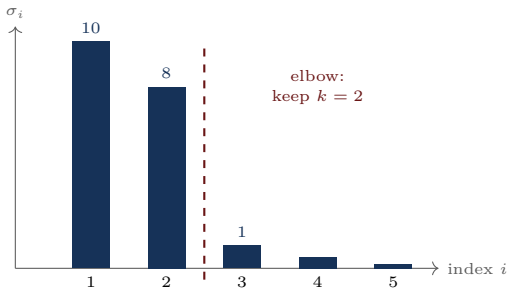



**Figure 4.3.** A scree plot of singular values $\sigma = (10, 8, 1, 0.5, 0.2)$. The sharp drop after the second value (the "elbow") says two components capture almost all the matrix; truncating there discards little (Exercise 4.4 computes the retained energy at over 99%).

## 4.7 Applications

- **Compression.** Storing $A_k$ needs only $k(m+n)+k$ numbers (the $\boldsymbol{u}_i$, $\boldsymbol{v}_i$, and $\sigma_i$) instead of $mn$. For an image with slowly decaying singular values a modest $k$ reproduces it almost exactly, the idea behind transform-based image coding.
- **Denoising.** Noise spreads its energy across all directions and so lives mostly in the small singular values; truncating to the top $k$ keeps the coherent signal and discards the noise floor.
- **Eigenfaces.** Stack many face images as the rows of $A$. The top right singular vectors $\boldsymbol{v}_i$ are the directions along which faces differ most; reshaped back into images they look like ghostly composite faces ("eigenfaces"), and every face is well approximated by a few of their coefficients. This is principal component analysis, developed in full in Chapter 11.
- **Latent structure.** In a term–document matrix the top singular vectors group words and documents into latent topics, the basis of latent semantic indexing.

**Example 4.8 (Image compression by the numbers).** A grayscale photo is a matrix; take a $512 \times 512$ image, so $A \in \mathbb{R}^{512\times 512}$ costs $512^2 = 262{,}144$ numbers to store in full. The rank-$k$ truncation $A_k = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\top}$ stores instead $k$ left vectors of length 512, $k$ right vectors of length 512, and $k$ singular values: $k(512+512+1) = 1025k$ numbers. Comparing:

| rank $k$ | numbers stored | vs. full | compression |
|---|---|---|---|
| 10 | 10,250 | 3.9% | 25.6× |
| 50 | 51,250 | 19.6% | 5.1× |
| 100 | 102,500 | 39.1% | 2.6× |

Because photographs have rapidly decaying singular values (most of their Frobenius energy $\sum_i \sigma_i^2$ sits in the first few dozen terms), a rank-50 approximation already looks nearly identical to the original while using a fifth of the storage. By Eckart–Young (Theorem 4.6) this is the *best* possible rank-50 image in squared error, and the relative error is exactly $\sqrt{\sum_{i>50} \sigma_i^2 \,/\, \sum_i \sigma_i^2}$, the fraction of energy in the discarded tail.

The thread running through all of these is principal component analysis, and the SVD computes it directly. If the rows of $A$ are data points with the column means subtracted (*centered* data), then the covariance matrix is $C = \frac{1}{m} A^{\top} A$, whose eigenvectors are exactly the right singular vectors $\boldsymbol{v}_i$ of $A$, with variances $\sigma_i^2/m$ along them. The top right singular vectors are therefore the directions of greatest variance, the *principal components*, and projecting onto the first $k$ of them is the best $k$-dimensional linear summary of the data by Eckart–Young. We build this out in full in Chapter 11; see also Jolliffe [14]. Computing principal components through the SVD of $A$, rather than by forming $C = \frac{1}{m} A^{\top} A$, is preferred for the same conditioning reason that $QR$ beat the normal equations in Section 2.11.

## 4.8 The pseudoinverse, finally in general

Chapter 2 built the Moore–Penrose pseudoinverse only for full-column-rank matrices, leaving the rank-deficient case (Example 2.12) unresolved. The SVD closes the gap. If $A = U\Sigma V^\top$, define

$$A^\dagger = V\Sigma^\dagger U^\top, \qquad \Sigma^\dagger = \operatorname{diag}\Big(\tfrac{1}{\sigma_1}, \dots, \tfrac{1}{\sigma_r}, 0, \dots, 0\Big), \tag{4.7}$$

that is, invert each nonzero singular value and transpose. This $A^\dagger$ satisfies all four Penrose conditions Eq. (2.17) for *any* matrix, and when $A$ has full column rank it agrees with $(A^\top A)^{-1}A^\top$. The least-squares solution $\boldsymbol{x}^\star = A^\dagger \boldsymbol{b}$ is then the one of *minimum norm* among all minimizers, the canonical choice the rank-deficient problem lacked.

**Example 4.9 (Pseudoinverse of a rank-one matrix).** Take the rank-deficient $A = \left[\begin{smallmatrix} 1 & 2 \\ 2 & 4 \end{smallmatrix}\right]$ from Example 2.12. It is the single rank-one term $A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top$ with $\sigma_1 = 5$, $\boldsymbol{u}_1 = \boldsymbol{v}_1 = \frac{1}{\sqrt{5}}(1, 2)$ (one checks $\|A\|_F = 5$ and the column and row directions are both $(1, 2)$). By Eq. (4.7),

$$A^\dagger = \frac{1}{\sigma_1}\boldsymbol{v}_1\boldsymbol{u}_1^\top = \frac{1}{5}\cdot\frac{1}{5}\begin{bmatrix}1\\2\end{bmatrix}\begin{bmatrix}1 & 2\end{bmatrix} = \frac{1}{25}\begin{bmatrix}1 & 2\\2 & 4\end{bmatrix} = \frac{A}{25}.$$

A direct check confirms $AA^\dagger A = A$: since $A^2 = 5A$ (as $\boldsymbol{v}_1^\top \boldsymbol{u}_1$ gives the factor), $AA^\dagger A = A^3/25 = 25A/25 = A$. The naive formula $(A^\top A)^{-1}A^\top$ is unavailable here because $A^\top A$ is singular, yet the SVD pseudoinverse exists and is unique.

**Example 4.10 (Minimum-norm solution of an underdetermined system).** The phrase "minimum norm among all minimizers" deserves a picture. Consider the single equation in two unknowns

$$x_1 + x_2 = 1, \qquad \text{that is} \qquad A\boldsymbol{x} = \boldsymbol{b} \text{ with } A = \begin{bmatrix}1 & 1\end{bmatrix},\ \boldsymbol{b} = 1.$$

Infinitely many vectors satisfy it: $(1, 0)$, $(0, 1)$, $(\frac{1}{2}, \frac{1}{2})$, $(2, -1)$, and every other point of the line $x_1 + x_2 = 1$. With no extra criterion the problem has no single answer, so the pseudoinverse supplies one. Here $A$ has full row rank, so $A^\dagger = A^\top(AA^\top)^{-1}$, and since $AA^\top = 1^2 + 1^2 = 2$,

$$A^\dagger = \frac{1}{2}\begin{bmatrix}1\\1\end{bmatrix}, \qquad \boldsymbol{x}^\star = A^\dagger \boldsymbol{b} = \Big(\tfrac{1}{2}, \tfrac{1}{2}\Big).$$

Among all solutions this one is shortest: $\|\boldsymbol{x}^\star\| = \sqrt{\frac{1}{4} + \frac{1}{4}} = \frac{1}{\sqrt{2}} \approx 0.707$, against $\|(1, 0)\| = \|(0, 1)\| = 1$. Geometrically the solution set is a line, and $(\frac{1}{2}, \frac{1}{2})$ is the foot of the perpendicular dropped from the origin onto it, the point of the line closest to $\mathbf{0}$. The pseudoinverse always returns that foot-of-perpendicular solution, which is exactly the behavior minimum-energy control and the zero-noise limit of ridge regression rely on. Of all the ways to split one unit across two accounts, the most balanced split, half and half, is also the one with the least total squared size.

**Example 4.11 (Synthesis: one least-squares fit, three routes that agree).** The book has built three different machines for the same job. Run all three on the line fit $y = a + bx$ to $(0, 1), (1, 1), (2, 3)$, with $A = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ and $\boldsymbol{y} = (1, 1, 3)$, and watch them land on the same answer. *Route 1, normal equations* (Chapter 2). With $A^\top A = \left[\begin{smallmatrix} 3 & 3 \\ 3 & 5 \end{smallmatrix}\right]$ and $A^\top \boldsymbol{y} = (5, 7)$, the determinant is $15 - 9 = 6$, so $a = \frac{5\cdot 5 - 3\cdot 7}{6} = \frac{4}{6} = \frac{2}{3}$ and $b = \frac{3\cdot 7 - 3\cdot 5}{6} = 1$. *Route 2, QR* (Example 1.5). Gram–Schmidt on the columns gives $\boldsymbol{q}_1 = \frac{1}{\sqrt{3}}(1, 1, 1)$ and $\boldsymbol{q}_2 = \frac{1}{\sqrt{2}}(-1, 0, 1)$, so $R = \left[\begin{smallmatrix} \sqrt{3} & \sqrt{3} \\ 0 & \sqrt{2} \end{smallmatrix}\right]$ and $Q^\top \boldsymbol{y} = (\frac{5}{\sqrt{3}}, \sqrt{2})$. Back-substitution gives $\sqrt{2}\, b = \sqrt{2}$, so $b = 1$, then $\sqrt{3}\, a + \sqrt{3} = \frac{5}{\sqrt{3}}$, so $a = \frac{2}{3}$. *Route 3, the pseudoinverse* (this chapter). $A$ has full column rank, so $A^\dagger = (A^\top A)^{-1}A^\top$ and $\boldsymbol{x}^\star = A^\dagger \boldsymbol{y} = (\frac{2}{3}, 1)$ directly. All three return $(a, b) = (\frac{2}{3}, 1)$ with fitted values $\hat{\boldsymbol{y}} = (\frac{2}{3}, \frac{5}{3}, \frac{8}{3})$, because they are three computations of the *same* orthogonal projection of $\boldsymbol{y}$ onto $\operatorname{Col}(A)$, differing only in numerical path. Normal equations are fastest but square the condition number; $QR$ is the stable workhorse; the SVD pseudoinverse is the most robust and the only one that still returns a (minimum-norm) answer when $A$ is rank deficient. Same destination, different roads, and the SVD is the road that never washes out.

| | **Eigendecomposition** $A = PDP^{-1}$ | **SVD** $A = U\Sigma V^\top$ |
|---|---|---|
| Applies to | square matrices | every matrix (any shape) |
| Always exists? | no (defective matrices fail) | yes |
| Bases | one basis $P$, generally not orthogonal | two orthonormal bases $U, V$ |
| Scalars | eigenvalues $\lambda_i$ (real or complex) | singular values $\sigma_i \geq 0$ |
| Symmetric PSD $A$ | $P = Q$ orthogonal, $D = \Lambda$ | coincides: $U = V = Q$, $\sigma_i = \lambda_i$ |

**Table 4.1.** Eigendecomposition versus SVD. The SVD trades a single (possibly skew) basis for two orthonormal ones, and in return works for every matrix. The two agree exactly on symmetric positive semidefinite matrices.

## 4.9 Relation to the eigendecomposition

The SVD and the eigendecomposition coincide exactly when $A$ is symmetric positive semidefinite: then $A = Q\Lambda Q^\top$ with $\Lambda \geq 0$ is already of the form $U\Sigma V^\top$ with $U = V = Q$ and $\sigma_i = \lambda_i$. For a symmetric matrix with a negative eigenvalue the singular value is its absolute value, $\sigma_i = |\lambda_i|$, and the sign is absorbed into $\boldsymbol{u}_i = \pm \boldsymbol{v}_i$. For a general matrix the eigenvalues of $A$ (if it has any) and its singular values are unrelated. Two facts transfer cleanly: the rank of $A$ equals the number of nonzero singular values, and the nonzero singular values of $A$ are the square roots of the nonzero eigenvalues of both $A^\top A$ and $AA^\top$. Table 4.1 lays the two decompositions side by side.

## 4.10 Summary

- Every matrix factors as $A = U\Sigma V^\top$ with $U, V$ orthogonal and $\Sigma$ diagonal with nonnegative singular values $\sigma_1 \geq \cdots \geq \sigma_r > 0$.
- Construct it from $A^\top A$: its orthonormal eigenvectors are the columns of $V$, $\sigma_i = \sqrt{\lambda_i}$, and $\boldsymbol{u}_i = A\boldsymbol{v}_i/\sigma_i$.
- The SVD writes $A = \sum_i \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$, a sum of rank-one pieces ordered by importance, and sends the unit sphere to an ellipsoid with semi-axes $\sigma_i \boldsymbol{u}_i$. Every matrix acts as rotate–stretch–rotate.
- The SVD reveals everything: $\operatorname{rank}(A)$ is the number of nonzero $\sigma_i$, $\|A\|_2 = \sigma_1$, $\|A\|_F = \sqrt{\sum_i \sigma_i^2}$, $\kappa(A) = \sigma_1/\sigma_r$, and orthonormal bases for all four fundamental subspaces Eq. (4.4).
- Eckart–Young: the truncation $A_k = \sum_{i \leq k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$ is the best rank-$k$ approximation, with error set by the discarded singular values. This powers compression, denoising, eigenfaces, and PCA.
- The general pseudoinverse is $A^\dagger = V\Sigma^\dagger U^\top$, inverting the nonzero singular values; it exists for every matrix and gives the minimum-norm least-squares solution.

## Exercises

**Exercise 4.1.** Find the singular values and a singular value decomposition of

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

*Solution.* Compute $B^\top B$. Column 1 of $B$ is zero, and columns $2, 3, 4$ are $\boldsymbol{e}_1, 2\boldsymbol{e}_2, 3\boldsymbol{e}_3$, so $B^\top B = \operatorname{diag}(0, 1, 4, 9)$ is already diagonal. Its eigenvalues are $0, 1, 4, 9$ with eigenvectors the standard basis vectors, so the singular values are their square roots, $3, 2, 1, 0$ in decreasing order, with right singular vectors $\boldsymbol{v}_1 = \boldsymbol{e}_4$, $\boldsymbol{v}_2 = \boldsymbol{e}_3$, $\boldsymbol{v}_3 = \boldsymbol{e}_2$, $\boldsymbol{v}_4 = \boldsymbol{e}_1$. The left singular vectors for the nonzero singular values are $\boldsymbol{u}_i = B\boldsymbol{v}_i/\sigma_i$:

$$\boldsymbol{u}_1 = \tfrac{1}{3} B\boldsymbol{e}_4 = \boldsymbol{e}_3, \quad \boldsymbol{u}_2 = \tfrac{1}{2} B\boldsymbol{e}_3 = \boldsymbol{e}_2, \quad \boldsymbol{u}_3 = B\boldsymbol{e}_2 = \boldsymbol{e}_1,$$

and $\boldsymbol{u}_4 = \boldsymbol{e}_4$ completes the basis (it spans $\operatorname{Nul}(B^\top)$). Thus $B = U\Sigma V^\top$ with $\Sigma = \operatorname{diag}(3, 2, 1, 0)$, $U = [\boldsymbol{e}_3\ \boldsymbol{e}_2\ \boldsymbol{e}_1\ \boldsymbol{e}_4]$, and $V = [\boldsymbol{e}_4\ \boldsymbol{e}_3\ \boldsymbol{e}_2\ \boldsymbol{e}_1]$. The matrix simply shifts and rescales coordinates, and the zero column forces $\operatorname{rank}(B) = 3$. □

**Exercise 4.2.** Find the singular values of $A = \begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 1 & 0 \end{bmatrix}$ and explain why $\Sigma$ is $3 \times 2$ while $U$ is $3 \times 3$.

*Solution.* Here $A^\top A = \begin{bmatrix} 3 & -1 \\ -1 & 1 \end{bmatrix}$, with characteristic polynomial $(3-\lambda)(1-\lambda) - 1 = \lambda^2 - 4\lambda + 2 = 0$, so $\lambda = 2 \pm \sqrt{2}$. The singular values are $\sigma_1 = \sqrt{2+\sqrt{2}} \approx 1.848$ and $\sigma_2 = \sqrt{2-\sqrt{2}} \approx 0.765$; since $A$ has rank 2 there are two nonzero singular values. Because $A$ maps $\mathbb{R}^2 \to \mathbb{R}^3$, the matrix $\Sigma$ must be $3 \times 2$ (a $2 \times 2$ diagonal block of singular values atop a zero row), and $U$ must be $3 \times 3$ to be orthogonal in the output space $\mathbb{R}^3$. Only two left singular vectors come from $\boldsymbol{u}_i = A\boldsymbol{v}_i/\sigma_i$; the third column of $U$ is any unit vector orthogonal to both, obtained by Gram–Schmidt, and it spans $\mathrm{Nul}(A^\top)$, the one direction in $\mathbb{R}^3$ that $A$ never reaches. □

**Exercise 4.3.** Let $A = U\Sigma V^\top$ be the SVD of an invertible square matrix. Express $A^{-1}$, $A^\top$, and $\|A\|_2$ in terms of $U$, $\Sigma$, $V$, and the singular values.

*Solution.* Since $U, V$ are orthogonal, $U^{-1} = U^\top$ and $V^{-1} = V^\top$, so

$$A^{-1} = (U\Sigma V^\top)^{-1} = V\Sigma^{-1}U^\top,$$

where $\Sigma^{-1} = \mathrm{diag}(1/\sigma_1, \dots, 1/\sigma_n)$ (all $\sigma_i > 0$ because $A$ is invertible). The transpose is $A^\top = V\Sigma^\top U^\top = V\Sigma U^\top$ (a diagonal $\Sigma$ equals its transpose for a square matrix). The spectral norm is the largest singular value, $\|A\|_2 = \sigma_1$, while $\|A^{-1}\|_2 = 1/\sigma_n$; their product $\sigma_1/\sigma_n$ is the *condition number* of $A$, the quantity that measured near-singularity in Remark 1.20. A matrix is ill-conditioned exactly when its smallest singular value is tiny next to its largest. □

**Exercise 4.4.** A data matrix has singular values $\sigma = (10, 8, 1, 0.5, 0.2)$. What fraction of the Frobenius "energy" $\sum_i \sigma_i^2$ is retained by the best rank-2 approximation, and what is the resulting relative error $\|A - A_2\|_F / \|A\|_F$?

*Solution.* The total energy is $\sum_i \sigma_i^2 = 100 + 64 + 1 + 0.25 + 0.04 = 165.29$. The rank-2 truncation keeps $\sigma_1, \sigma_2$, retaining $100 + 64 = 164$, a fraction $164/165.29 \approx 0.992$, so over 99% of the energy lives in the top two components. By Theorem 4.6 the error is $\|A - A_2\|_F = \sqrt{\sigma_3^2 + \sigma_4^2 + \sigma_5^2} = \sqrt{1.29} \approx 1.136$, and the relative error is $\sqrt{1.29/165.29} \approx 0.088$, about 8.8%. Rapidly decaying singular values are exactly what make a low-rank approximation faithful. □

**Exercise 4.5.** Let $A = \begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$. (a) Find the singular values and the condition number $\kappa_2(A) = \sigma_{\max}/\sigma_{\min}$. (b) The system $A\boldsymbol{x} = \boldsymbol{b}$ is solved, then $\boldsymbol{b}$ is measured with a 1% relative error. What is the largest relative error in $\boldsymbol{x}$ that conditioning permits?

*Solution.* (a) The matrix is diagonal, so it already is its own SVD (with $U = V = I$): the singular values are the magnitudes of the diagonal entries, $\sigma = (2, \frac{1}{2})$. Hence

$$\kappa_2(A) = \frac{\sigma_{\max}}{\sigma_{\min}} = \frac{2}{1/2} = 4.$$

(b) The condition number is exactly the worst-case amplification of relative error in solving a linear system,

$$\frac{\|\Delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \le \kappa_2(A)\, \frac{\|\Delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|} = 4 \times 1\% = 4\%.$$

The bound is attained when $\boldsymbol{b}$ points along the weakly-amplified output direction $(0, 1)$ (the $\sigma = \frac{1}{2}$ axis) while the perturbation $\Delta\boldsymbol{b}$ points along the strongly-amplified $(1, 0)$ axis. A matrix with $\kappa$ near 1 barely magnifies input noise; a large $\kappa$, which means a tiny $\sigma_{\min}$ relative to $\sigma_{\max}$, makes the system dangerously sensitive, the numerical reason near-singular matrices are avoided and regularization (Section 2.8) is used to lift $\sigma_{\min}$ away from zero. □

**Exercise 4.6.** A grayscale image is a $100 \times 80$ matrix. You approximate it by its best rank-10 truncation. How many numbers must you store, and what is the compression ratio against the full image?

*Solution.* The full image is $100 \times 80 = 8000$ numbers. A rank-10 truncation $A_{10} = \sum_{i=1}^{10} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$ needs only the pieces of those ten terms: ten left singular vectors of length 100, ten right singular vectors of length 80, and ten singular values, which is

$$10\,(100 + 80 + 1) = 10(181) = 1810 \text{ numbers}.$$

The compression ratio is $8000/1810 \approx 4.4\times$, a saving of about 77%, and the image still looks faithful whenever the singular values past the tenth are small (Exercise 4.4). The general rule for an $m \times n$ image at rank $k$ is $k(m+n+1)$ stored numbers against $mn$, so truncation pays off precisely when $k \ll \frac{mn}{m+n}$; this is the arithmetic behind every low-rank image and model compression scheme. □

PART III

# Probability and Distributions

CHAPTER 5

# Probability, Bayes, and Estimation

*Bayes' theorem is not a formula. It is a systematic way to update what you believe in the presence of uncertainty.*

*Professor Chin*

**Roadmap**

The course now turns from deterministic linear algebra to reasoning under uncertainty. We model data with random variables, starting with the humble coin flip, and ask the central question of statistics: given the data, what is the parameter? Two answers compete. The *frequentist* treats the parameter as a fixed unknown and maximizes the likelihood; the *Bayesian* treats it as a random variable and updates a prior into a posterior through Bayes' theorem. We develop both for the Bernoulli model, prove they agree as data accumulates, and make the ideas concrete with disease testing and the Monty Hall problem.

**Suggested reading:** Deisenroth et al. [5], Sections 6.1–6.3; Bishop [2], Sections 1.2 and 2.1 for the Bayesian coin in detail.

## 5.1 The rules of probability

All of probability is built from two rules. For random variables $X, Y$ with a joint distribution $p(X, Y)$, the *sum rule* recovers one variable by summing (integrating) out the other, and the *product rule* factors a joint into a conditional times a marginal:

$$p(X) = \sum_{y} p(X, Y{=}y) \quad \text{(sum rule)}, \qquad p(X, Y) = p(X \mid Y)\, p(Y) \quad \text{(product rule)}. \tag{5.1}$$

Bayes' theorem (Section 5.5) is nothing more than the product rule written both ways. Two variables are *independent* if $p(X, Y) = p(X)p(Y)$, equivalently $p(X \mid Y) = p(X)$: knowing one says nothing about the other. The *expectation* and *variance* of a random variable summarize its center and spread,

$$\mathbb{E}[X] = \sum_{x} x\, p(x), \qquad \operatorname{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2, \tag{5.2}$$

with the integral replacing the sum for continuous variables. Expectation is *linear*, $\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$, always, whether or not $X$ and $Y$ are independent; variance adds only when they are. These few rules, applied to the coin flip, generate the whole chapter.

## 5.2 Random variables and the Bernoulli model

A *random variable* assigns a numerical outcome to a random experiment. The simplest is the *Bernoulli* variable, the model of a single coin flip.

**Definition 5.1 (Bernoulli random variable).** A random variable $X \in \{0, 1\}$ is *Bernoulli* with parameter $\mu \in [0, 1]$, written $X \sim \text{Bernoulli}(\mu)$, if

$$\mathbb{P}(X = 1) = \mu, \qquad \mathbb{P}(X = 0) = 1 - \mu.$$

Its probability mass function can be written in one line as $p(x \mid \mu) = \mu^x (1-\mu)^{1-x}$ for $x \in \{0, 1\}$.

A short computation gives its mean and variance:

$$\mathbb{E}[X] = 0 \cdot (1-\mu) + 1 \cdot \mu = \mu, \qquad \mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \mu - \mu^2 = \mu(1-\mu), \tag{5.3}$$

where $\mathbb{E}[X^2] = \mu$ because $X^2 = X$ for a 0/1 variable. The variance is largest at $\mu = \frac{1}{2}$ (maximum uncertainty) and zero at $\mu \in \{0, 1\}$ (a certain outcome).

**Example 5.2 (Mean and variance of a fair die).** The Bernoulli has only two outcomes; the same two formulas handle any discrete variable. Let $X$ be the roll of a fair six-sided die, uniform on $\{1, 2, 3, 4, 5, 6\}$ with probability $\frac{1}{6}$ each. The expectation is the probability-weighted average

$$\mathbb{E}[X] = \sum_{k=1}^{6} k \cdot \tfrac{1}{6} = \frac{1+2+3+4+5+6}{6} = \frac{21}{6} = \frac{7}{2} = 3.5.$$

For the variance, first take the second moment $\mathbb{E}[X^2] = \sum_k k^2 \cdot \frac{1}{6} = \frac{1+4+9+16+25+36}{6} = \frac{91}{6}$, then subtract the squared mean:

$$\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \frac{91}{6} - \left(\frac{7}{2}\right)^2 = \frac{182 - 147}{12} = \frac{35}{12} \approx 2.917,$$

so the standard deviation is $\sqrt{35/12} \approx 1.708$. Notice the mean 3.5 is not a value the die can ever show: expectation is the balance point of the six equal weights placed at 1 through 6, the center of mass of the histogram rather than a typical outcome. The variance, meanwhile, is the average squared distance from that balance point, and its square root reports spread in the original units (pips), which is why the standard deviation, rather than the variance, is the one quoted alongside a mean.

We rarely observe $\mu$ directly. Instead we see a *dataset* of $n$ independent flips of the same coin,

$$\mathcal{D} = \{x_1, \ldots, x_n\}, \qquad x_i \overset{\text{iid}}{\sim} \text{Bernoulli}(\mu), \tag{5.4}$$

and must infer $\mu$. Write $m = \sum_{i=1}^{n} x_i$ for the number of ones (heads). The whole chapter is about turning $\mathcal{D}$ into a statement about $\mu$.

## 5.3 Maximum likelihood estimation

The *likelihood* is the probability of the observed data viewed as a function of the parameter. Because the flips are independent, their joint probability is the product of the individual masses:

$$p(\mathcal{D} \mid \mu) = \prod_{i=1}^{n} \mu^{x_i} (1-\mu)^{1-x_i} = \mu^m (1-\mu)^{n-m}. \tag{5.5}$$

The *maximum likelihood estimate* (MLE) is the parameter that makes the data most probable, $\hat{\mu} = \arg\max_{\mu \in [0,1]} p(\mathcal{D} \mid \mu)$.

*Intuition.* The same expression $p(\mathcal{D} \mid \mu)$ is read two ways. Fix $\mu$ and let the data vary: it is a *probability*, summing to one over all datasets. Fix the observed data and let $\mu$ vary: it is a *likelihood*, a function of the parameter that does *not* integrate to one. Maximum likelihood asks the natural question, "which coin bias would have made what I actually saw the least surprising?" and answers by climbing that curve to its peak.

**Proposition 5.3 (Bernoulli MLE).** The maximum likelihood estimate of the Bernoulli parameter is the sample mean,

$$\hat{\mu}_{\text{MLE}} = \frac{m}{n} = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{5.6}$$

*Proof.* Maximizing the likelihood is the same as maximizing its logarithm, since log is increasing; the log turns the product into a sum and is far easier to differentiate. The *log-likelihood* is

$$\ell(\mu) = \log p(\mathcal{D} \mid \mu) = m \log \mu + (n-m) \log(1-\mu).$$

| | **Frequentist** | **Bayesian** |
|---|---|---|
| The parameter $\mu$ is | a fixed unknown constant | a random variable |
| Probability means | long-run frequency | degree of belief |
| What is random | the data | the parameter (given the data) |
| The output is | a point estimate ($\hat{\mu}_{\text{MLE}}$) | a posterior distribution |
| Small samples | can be brittle (3 heads $\Rightarrow \hat{\mu}$=1) | regularized by the prior |

**Table 5.1.** Two philosophies of inference. They are interpretations rather than competing theorems, and Section 5.7 shows their estimates coincide as the data grows.

Differentiating and setting to zero,

$$\ell'(\mu) = \frac{m}{\mu} - \frac{n-m}{1-\mu} = 0 \implies m(1-\mu) = (n-m)\mu \implies m = n\mu,$$

so $\hat{\mu} = m/n$. The second derivative $\ell''(\mu) = -m/\mu^2 - (n-m)/(1-\mu)^2 < 0$ confirms a maximum. ■

The MLE is intuitive but brittle in small samples. Flip a coin three times and see three heads, and the MLE declares $\hat{\mu} = 1$: the coin will *never* come up tails. It also reports a single number with no measure of confidence. Both shortcomings motivate the Bayesian treatment.

## 5.4 Two philosophies

- The **frequentist** regards $\mu$ as a fixed but unknown constant. Probability describes the long-run frequency of repeatable events, and the data is random while the parameter is not. The output is a point estimate like the MLE.
- The **Bayesian** regards $\mu$ as a random variable carrying our uncertainty. Probability describes a degree of belief. We start with a *prior* distribution over $\mu$ and update it to a *posterior* once the data is in. The output is a whole distribution rather than a single number.

These are not rival theorems, only rival interpretations (Table 5.1), and we will see they deliver the same answer once the data is plentiful. The Bayesian machinery rests on a single identity.

## 5.5 Bayes' theorem

The joint probability of two random quantities factors two ways, by the definition of conditional probability:

$$p(X, Y) = p(X)\,p(Y \mid X) = p(Y)\,p(X \mid Y).$$

Equating the last two expressions and dividing gives the result.

**Theorem 5.4 (Bayes' theorem).** For events or random variables with $p(Y) > 0$,

$$p(X \mid Y) = \frac{p(Y \mid X)\,p(X)}{p(Y)}. \tag{5.7}$$

When $X$ is a parameter $\theta$ and $Y$ is data $\mathcal{D}$, the pieces have names:

$$\underbrace{p(\theta \mid \mathcal{D})}_{\text{posterior}} = \frac{\overbrace{p(\mathcal{D} \mid \theta)}^{\text{likelihood}}\ \overbrace{p(\theta)}^{\text{prior}}}{\underbrace{p(\mathcal{D})}_{\text{evidence}}}, \qquad p(\theta \mid \mathcal{D}) \ \propto\ p(\mathcal{D} \mid \theta)\,p(\theta). \tag{5.8}$$

The evidence $p(\mathcal{D}) = \int p(\mathcal{D} \mid \theta) p(\theta)\, d\theta$ is a constant in $\theta$; it only normalizes the posterior to integrate to one, which is why the proportionality on the right is usually all we need. In words: the posterior is the prior reweighted by how well each parameter value explains the data.

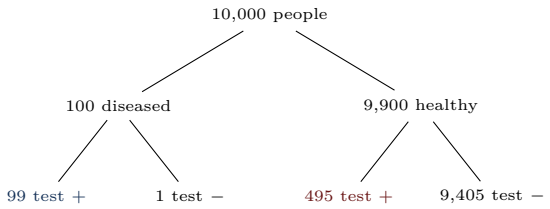



**Figure 5.1.** The disease test in natural frequencies. Splitting 10,000 people by disease status and then by test result makes the base-rate effect obvious: because only 1% are sick, the few true positives are buried under false positives drawn from the large healthy majority.

**Example 5.5 (Disease testing and base rates).** A test for a disease has sensitivity $\mathbb{P}(+ \mid D) = 0.99$ and false-positive rate $\mathbb{P}(+ \mid \neg D) = 0.05$, and the disease has prevalence $\mathbb{P}(D) = 0.01$. A patient tests positive. What is $\mathbb{P}(D \mid +)$? By Bayes' theorem, with $\mathbb{P}(\neg D) = 0.99$,

$$\mathbb{P}(D \mid +) = \frac{\mathbb{P}(+ \mid D)\mathbb{P}(D)}{\mathbb{P}(+ \mid D)\mathbb{P}(D) + \mathbb{P}(+ \mid \neg D)\mathbb{P}(\neg D)} = \frac{0.99 \cdot 0.01}{0.99 \cdot 0.01 + 0.05 \cdot 0.99} = \frac{0.01}{0.06} = \frac{1}{6} \approx 16.7\%.$$

Despite a highly accurate test, a positive result means only a one-in-six chance of disease, because the disease is rare to begin with. Ignoring the small prior $\mathbb{P}(D)$, the *base rate*, is the classic error this computation guards against. The arithmetic is most transparent in *natural frequencies* (Fig. 5.1): of 10,000 people, the 99 true positives are swamped by 495 false positives, and $99/(99 + 495) = 1/6$.

**Example 5.6 (A second test: yesterday's posterior is today's prior).** Suppose the patient of Example 5.5 takes a *second*, independent test and it is also positive. Bayesian updating is sequential: the posterior after the first test, $\mathbb{P}(D) = \frac{1}{6}$, becomes the prior for the second. With the same test ($\mathbb{P}(+ \mid D) = 0.99$, $\mathbb{P}(+ \mid \neg D) = 0.05$),

$$\mathbb{P}(D \mid ++) = \frac{0.99 \cdot \frac{1}{6}}{0.99 \cdot \frac{1}{6} + 0.05 \cdot \frac{5}{6}} = \frac{0.165}{0.165 + 0.0417} = \frac{0.165}{0.2067} \approx 0.80.$$

One positive test left the odds at one in six; a second pushes them to about four in five. Equivalently, fold both observations in at once with the prior $\frac{1}{100}$ and squared likelihoods, $\mathbb{P}(D \mid ++) = \dfrac{0.99^2(0.01)}{0.99^2(0.01) + 0.05^2(0.99)} \approx 0.80$, the same answer: updating one datum at a time or all at once agrees, which is the hallmark of Bayesian consistency. Evidence compounds, and a rare disease can become probable once two independent tests concur.

**Example 5.7 (A naive-Bayes spam filter).** The same updating classifies email. Suppose 40% of mail is spam, so the prior is $\mathbb{P}(\text{spam}) = 0.4$, $\mathbb{P}(\text{ham}) = 0.6$. Two words have class-conditional rates

$$\mathbb{P}(\text{“free”} \mid \text{spam}) = 0.8,\ \mathbb{P}(\text{“free”} \mid \text{ham}) = 0.1, \quad \mathbb{P}(\text{“money”} \mid \text{spam}) = 0.7,\ \mathbb{P}(\text{“money”} \mid \text{ham}) = 0.2.$$

An email contains both words. The *naive* assumption is that the words are conditionally independent given the class, so the likelihoods multiply. Up to the shared normalizer,

$$\mathbb{P}(\text{spam} \mid \text{free}, \text{money}) \propto 0.4 \cdot 0.8 \cdot 0.7 = 0.224, \qquad \mathbb{P}(\text{ham} \mid \cdots) \propto 0.6 \cdot 0.1 \cdot 0.2 = 0.012.$$

Normalizing, $\mathbb{P}(\text{spam} \mid \text{free}, \text{money}) = \dfrac{0.224}{0.224 + 0.012} \approx 0.949$: about 95% spam. Each word's likelihood ratio multiplies the odds, so evidence accumulates exactly as in the repeated medical test. The independence assumption is "naive" (words are correlated in real text) yet works remarkably well, because the classifier needs only to get the *ranking* of the two posteriors right, not their exact values.

**Example 5.8 (Two factories: total probability, then Bayes).** A company sources a part from two factories. Factory $A$ makes 60% of the parts with a 2% defect rate; factory $B$ makes the other 40% with a 5% defect rate. A part is drawn at random and found defective. Which factory did it most likely come from?

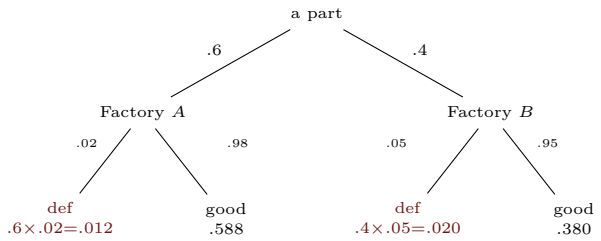



**Figure 5.2.** The two-factory problem as a tree. Split first by source (.6 vs .4), then by quality; each leaf multiplies the probabilities along its branch. The two "defective" leaves (red) sum to the total defect rate .032; the posterior $\mathbb{P}(A \mid \text{def})$ is factory $A$'s share of that total. Even though $A$ is the larger source, its lower defect rate makes it the minority of defectives.

*Step 1: the overall defect rate (law of total probability).* Splitting on the source,

$$\mathbb{P}(\text{def}) = \mathbb{P}(\text{def} \mid A)\mathbb{P}(A) + \mathbb{P}(\text{def} \mid B)\mathbb{P}(B) = (0.02)(0.60) + (0.05)(0.40) = 0.012 + 0.020 = 0.032,$$

so 3.2% of all parts are defective: a weighted blend of the two rates, tilted by how many parts each factory makes.

*Step 2: invert with Bayes' theorem.* The posterior probability the defective part came from $A$ is its share of that 0.032,

$$\mathbb{P}(A \mid \text{def}) = \frac{\mathbb{P}(\text{def} \mid A)\mathbb{P}(A)}{\mathbb{P}(\text{def})} = \frac{0.012}{0.032} = 0.375, \qquad \mathbb{P}(B \mid \text{def}) = \frac{0.020}{0.032} = 0.625.$$

Even though $A$ makes the *majority* of parts, a defective one more likely came from $B$ (62.5% versus 37.5%), because $B$'s higher defect rate outweighs its smaller share. This is the base-rate lesson again, mirror-imaged: the prior favors $A$, but the likelihood favors $B$, and Bayes weighs the two. Think of it as tracing a leak back upstream: more water flows from the bigger pipe, but the leakier pipe is the better suspect for any given drop.

## 5.6 The Beta distribution and conjugacy

To be Bayesian about the Bernoulli parameter we need a prior on $\mu \in [0, 1]$. The *Beta distribution* is the natural choice.

**Definition 5.9 (Beta distribution).** For shape parameters $a, b > 0$, the density of $\mu \sim \text{Beta}(a, b)$ on $[0, 1]$ is

$$p(\mu) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\, \mu^{a-1}(1-\mu)^{b-1}, \tag{5.9}$$

where $\Gamma$ is the gamma function ($\Gamma(k) = (k-1)!$ for integer $k$), and the leading constant makes the density integrate to one. Its mean is $\mathbb{E}[\mu] = \frac{a}{a+b}$.

The exponents $a-1$ and $b-1$ make the Beta density a rescaled copy of the Bernoulli likelihood Eq. (5.5), and that shared algebraic form is what makes the prior and posterior stay in the same family.

**Definition 5.10 (Conjugate prior).** A prior is *conjugate* to a likelihood if the posterior belongs to the same family as the prior. Conjugacy turns Bayesian updating into simple bookkeeping on the parameters.

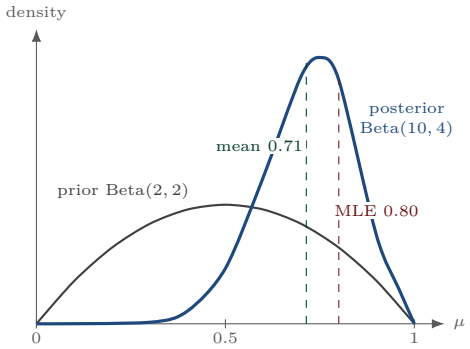



**Figure 5.3.** Bayesian updating for the coin. The gentle prior Beta(2, 2) becomes the sharper posterior Beta(10, 4) after seeing 8 heads in 10 flips. The data pulls belief toward 0.8, but the prior holds the posterior mean back to 0.71; more data would sharpen the posterior further and erase the gap.

## 5.7 Bayesian estimation of a Bernoulli parameter

Put a Beta$(a, b)$ prior on $\mu$ and observe $m$ heads in $n$ flips. Multiply the likelihood Eq. (5.5) by the prior Eq. (5.9) and drop constants:

$$p(\mu \mid \mathcal{D}) \;\propto\; \underbrace{\mu^{m}(1-\mu)^{n-m}}_{\text{likelihood}} \cdot \underbrace{\mu^{a-1}(1-\mu)^{b-1}}_{\text{prior}} = \mu^{(a+m)-1}(1-\mu)^{(b+n-m)-1}. \tag{5.10}$$

The right-hand side is the kernel of another Beta density, so the posterior is

**The Beta–Bernoulli update**

$$\mu \sim \text{Beta}(a, b) \quad \text{and} \quad m \text{ heads in } n \text{ flips} \qquad \Longrightarrow \qquad \mu \mid \mathcal{D} \sim \text{Beta}(a+m,\; b+n-m).$$

Updating just adds the observed heads to $a$ and the observed tails to $b$. The prior parameters act as *pseudo-counts*: $a$ imaginary prior heads and $b$ imaginary prior tails, mixed in with the real data. From the Beta mean,

$$\mathbb{E}[\mu \mid \mathcal{D}] = \frac{a+m}{a+b+n}, \qquad \text{Var}(\mu \mid \mathcal{D}) = \frac{(a+m)(b+n-m)}{(a+b+n)^2(a+b+n+1)}. \tag{5.11}$$

Two limits confirm the picture. With no data ($n = 0$) the posterior mean is the prior mean $a/(a+b)$. As $n \to \infty$ the pseudo-counts $a, b$ become negligible and $\mathbb{E}[\mu \mid \mathcal{D}] \to m/n$, the MLE, while the variance shrinks to zero: with enough data the posterior collapses onto the frequentist answer, and the choice of prior stops mattering. The Bayesian estimate is, in effect, a smoothed MLE that behaves sanely in small samples (the three-heads coin no longer forces $\hat{\mu} = 1$).

**Example 5.11 (Posterior, posterior mean, and MAP).** Start from a mild prior Beta(2, 2) (centered at $\mu = \frac{1}{2}$, gently favoring fairness) and observe $m = 8$ heads in $n = 10$ flips. The posterior is Beta$(2+8,\; 2+2) =$ Beta$(10, 4)$. Three estimates of $\mu$ compare as follows:

$$\hat{\mu}_{\text{MLE}} = \frac{m}{n} = 0.80, \quad \mathbb{E}[\mu \mid \mathcal{D}] = \frac{10}{14} = \frac{5}{7} \approx 0.714, \quad \hat{\mu}_{\text{MAP}} = \frac{10-1}{14-2} = \frac{9}{12} = 0.75,$$

where the *maximum a posteriori* estimate is the mode of the posterior, $(a'-1)/(a'+b'-2)$ for Beta$(a', b')$. The Bayesian estimates sit between the data (0.80) and the prior mean (0.5): the prior pulls the estimate toward fairness, an effect that fades as $n$ grows (Fig. 5.3).

*Remark* 5.12 *(MAP estimation is regularization).* Taking the log of Bayes' rule Eq. (5.8) turns the MAP estimate into

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} \Big[ \underbrace{\log p(\mathcal{D} \mid \theta)}_{\text{fit}} + \underbrace{\log p(\theta)}_{\text{penalty}} \Big],$$

a data-fit term plus a prior term. This is exactly the shape of regularized least squares from Section 2.8: there the Gaussian noise model gave the squared-error fit and a Gaussian prior $\theta \sim \mathcal{N}(\mathbf{0}, \tau^2 I)$ gave the penalty $\lambda\|\theta\|^2$ with $\lambda = \sigma^2/\tau^2$. Ridge regression *is* MAP estimation under a Gaussian prior, and the Beta prior here plays the same regularizing role for the coin: the prior is a penalty that keeps the estimate sane when the data is thin.

## 5.8 The Monty Hall problem

Bayes' theorem resolves a famously counterintuitive puzzle and shows why *who* reveals information, and *how*, changes the answer.

### 5.8.1 Three doors

A car hides behind one of three doors, goats behind the others. You pick Door 1. The host, who knows the layout, opens a different door showing a goat, say Door 2, and offers you the switch. Let $H_i$ be the event "car behind Door $i$" and $E$ the event "host opens Door 2." The priors are uniform, $\mathbb{P}(H_i) = \frac{1}{3}$. The likelihoods encode the host's behavior:

$$\mathbb{P}(E \mid H_1) = \tfrac{1}{2} \text{ (car behind your door; host picks 2 or 3 freely)}, \quad \mathbb{P}(E \mid H_2) = 0, \quad \mathbb{P}(E \mid H_3) = 1.$$

The evidence is $\mathbb{P}(E) = \frac{1}{2} \cdot \frac{1}{3} + 0 + 1 \cdot \frac{1}{3} = \frac{1}{2}$, and Bayes' theorem gives

$$\mathbb{P}(H_1 \mid E) = \frac{\frac{1}{2} \cdot \frac{1}{3}}{\frac{1}{2}} = \frac{1}{3}, \qquad \mathbb{P}(H_3 \mid E) = \frac{1 \cdot \frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}.$$

Switching wins with probability $\frac{2}{3}$, double the $\frac{1}{3}$ from staying. The host's choice is not random; by never opening the car door he leaks information, and that information flows entirely onto the door you did not pick.

### 5.8.2 Four doors

With four doors (one car) you pick Door 1 and the host opens Door 2 on a goat. Now $\mathbb{P}(H_i) = \frac{1}{4}$, and the likelihoods of opening Door 2 are

$$\mathbb{P}(E \mid H_1) = \tfrac{1}{3}, \quad \mathbb{P}(E \mid H_2) = 0, \quad \mathbb{P}(E \mid H_3) = \tfrac{1}{2}, \quad \mathbb{P}(E \mid H_4) = \tfrac{1}{2},$$

since with the car behind your door the host chooses among Doors 2,3,4, while with the car behind Door 3 or 4 he chooses between the two remaining goat doors. The evidence is

$$\mathbb{P}(E) = \tfrac{1}{4}\left(\tfrac{1}{3} + 0 + \tfrac{1}{2} + \tfrac{1}{2}\right) = \tfrac{1}{3},$$

and the posteriors are

$$\mathbb{P}(H_1 \mid E) = \frac{\frac{1}{4} \cdot \frac{1}{3}}{\frac{1}{3}} = \frac{1}{4}, \qquad \mathbb{P}(H_3 \mid E) = \mathbb{P}(H_4 \mid E) = \frac{\frac{1}{4} \cdot \frac{1}{2}}{\frac{1}{3}} = \frac{3}{8}.$$

Staying keeps $\frac{1}{4}$, while either remaining door now carries $\frac{3}{8} > \frac{1}{4}$, so again you should switch. The opened door's prior mass redistributes onto the doors you neither picked nor saw opened.

## 5.9 Looking ahead: multiple outcomes

When outcomes have $K > 2$ categories rather than two, the Bernoulli becomes the *categorical* distribution and $n$ trials give the *multinomial*. Its conjugate prior is the *Dirichlet*, the multivariate cousin of the Beta, with density $p(\boldsymbol{\mu}) \propto \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}$ on the probability simplex $\{\boldsymbol{\mu} : \mu_k \geq 0, \sum_k \mu_k = 1\}$. Everything in this chapter, MLE, conjugacy, and the count-updating rule, carries over, and Chapter 6 develops it alongside the continuous workhorse of the course, the Gaussian. The conjugate pairs we will use are collected in Table 5.2.

| Likelihood | Parameter | Conjugate prior | Where |
|---|---|---|---|
| Bernoulli / Binomial | bias $\mu$ | Beta | this chapter |
| Categorical / Multinomial | probabilities $\boldsymbol{\mu}$ | Dirichlet | Chapter 6 |
| Gaussian (known variance) | mean $\mu$ | Gaussian | Chapter 6 |
| Gaussian (known mean) | variance | inverse-gamma | Chapter 6 |

**Table 5.2.** Conjugate prior families. In each row the posterior stays in the same family as the prior, so updating is just arithmetic on the parameters. The Beta and Dirichlet count successes; the Gaussian–Gaussian pair averages means.

## 5.10 Summary

- A Bernoulli variable models one binary trial: $\mathbb{E}[X] = \mu$, $\mathrm{Var}(X) = \mu(1-\mu)$. For $n$ iid flips the likelihood is $\mu^m(1-\mu)^{n-m}$.
- The MLE maximizes the (log-)likelihood and equals the sample mean, $\hat{\mu} = m/n$; it is brittle for small $n$ and reports no uncertainty.
- Bayes' theorem, $p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)p(\theta)$, updates a prior into a posterior. Base rates matter (Example 5.5).
- The Beta prior is conjugate to the Bernoulli likelihood: the posterior is $\mathrm{Beta}(a+m,\ b+n-m)$, the prior parameters acting as pseudo-counts. The posterior mean $\frac{a+m}{a+b+n}$ tends to the MLE as $n \to \infty$.
- Monty Hall shows that conditioning on an informed host's action shifts the posterior: switching wins with probability $\frac{2}{3}$ (three doors) or $\frac{3}{8}$ per remaining door (four doors).

## Exercises

**Exercise 5.1.** Let $x_1, \dots, x_n \overset{\text{iid}}{\sim} \mathrm{Bernoulli}(\mu)$ with $m = \sum_i x_i$ heads. Starting from the likelihood, derive the maximum likelihood estimate $\hat{\mu}$ and show the stationary point is a maximum.

*Solution.* The likelihood is $p(\mathcal{D} \mid \mu) = \prod_i \mu^{x_i}(1-\mu)^{1-x_i} = \mu^m(1-\mu)^{n-m}$. Its logarithm is $\ell(\mu) = m\log\mu + (n-m)\log(1-\mu)$. Then

$$\ell'(\mu) = \frac{m}{\mu} - \frac{n-m}{1-\mu} = 0 \implies m(1-\mu) = (n-m)\mu \implies \hat{\mu} = \frac{m}{n}.$$

The second derivative $\ell''(\mu) = -\dfrac{m}{\mu^2} - \dfrac{n-m}{(1-\mu)^2}$ is negative throughout $(0,1)$, so $\ell$ is concave and $\hat{\mu} = m/n$ is the global maximizer. The MLE is the sample mean. □

**Exercise 5.2.** A test has sensitivity $\mathbb{P}(+ \mid D) = 0.99$ and false-positive rate $\mathbb{P}(+ \mid \neg D) = 0.05$. The disease has prevalence $\mathbb{P}(D) = 0.01$. A patient tests positive. Find $\mathbb{P}(D \mid +)$, and explain why it is so much smaller than the test's accuracy.

*Solution.* By Bayes' theorem with the law of total probability in the denominator,

$$\mathbb{P}(D \mid +) = \frac{\mathbb{P}(+ \mid D)\mathbb{P}(D)}{\mathbb{P}(+ \mid D)\mathbb{P}(D) + \mathbb{P}(+ \mid \neg D)\mathbb{P}(\neg D)} = \frac{(0.99)(0.01)}{(0.99)(0.01) + (0.05)(0.99)} = \frac{0.01}{0.06} = \frac{1}{6}.$$

So a positive test gives only a 16.7% chance of disease. The reason is the small *base rate*: among 10,000 people only about 100 have the disease (almost all testing positive), while about 5% of the 9,900 healthy people, roughly 495, also test positive. True positives are outnumbered by false positives five-to-one, so a positive result is dominated by the large healthy population. □

**Exercise 5.3.** You place a $\mathrm{Beta}(2,2)$ prior on a coin's bias $\mu$ and then observe 8 heads in 10 flips. Find the posterior distribution, its mean, and the MAP estimate, and compare them with the MLE.

*Solution.* The Beta prior is conjugate, so the posterior adds heads to the first parameter and tails to the second: $\mathrm{Beta}(2+8,\ 2+(10-8)) = \mathrm{Beta}(10,4)$. Its mean is $\frac{10}{10+4} = \frac{5}{7} \approx 0.714$. The MAP estimate is the posterior

mode, $\frac{a'-1}{a'+b'-2} = \frac{9}{12} = 0.75$. The MLE is $\frac{8}{10} = 0.80$. The Bayesian estimates lie between the data (0.80) and the prior mean (0.5): the prior, worth four pseudo-flips, pulls the estimate toward fairness. With far more data the pull would vanish and all three would converge to $m/n$. □

**Exercise 5.4.** Four doors hide one car and three goats. You pick Door 1; the host opens Door 2 on a goat. Compute the posterior probability that the car is behind each door, and decide whether to switch.

*Solution.* Let $H_i$ be "car behind Door $i$" with uniform prior $\mathbb{P}(H_i) = \frac{1}{4}$, and $E$ the event "host opens Door 2." The host never opens your door or the car, and chooses uniformly among the doors he may open:

$$\mathbb{P}(E \mid H_1) = \tfrac{1}{3}, \quad \mathbb{P}(E \mid H_2) = 0, \quad \mathbb{P}(E \mid H_3) = \tfrac{1}{2}, \quad \mathbb{P}(E \mid H_4) = \tfrac{1}{2}.$$

The evidence is $\mathbb{P}(E) = \frac{1}{4}(\frac{1}{3} + 0 + \frac{1}{2} + \frac{1}{2}) = \frac{1}{3}$. Hence

$$\mathbb{P}(H_1 \mid E) = \frac{\frac{1}{4}\cdot\frac{1}{3}}{\frac{1}{3}} = \frac{1}{4}, \quad \mathbb{P}(H_2 \mid E) = 0, \quad \mathbb{P}(H_3 \mid E) = \mathbb{P}(H_4 \mid E) = \frac{\frac{1}{4}\cdot\frac{1}{2}}{\frac{1}{3}} = \frac{3}{8},$$

which sum to 1. Staying on Door 1 wins with probability $\frac{1}{4}$; switching to either Door 3 or Door 4 wins with probability $\frac{3}{8} > \frac{1}{4}$. You should switch. The mass that was on the opened door has moved onto the doors you neither chose nor saw opened. □

**Exercise 5.5.** Let $X \sim \text{Binomial}(10, 0.3)$ count the heads in ten flips of a coin that lands heads with probability 0.3. Find $\mathbb{E}[X]$, $\text{Var}(X)$, and the standard deviation, working from the fact that $X$ is a sum of ten independent Bernoulli variables.

*Solution.* Write $X = \sum_{i=1}^{10} B_i$ with each $B_i \sim \text{Bernoulli}(0.3)$ independent. Expectation is linear, so it adds whether or not the terms are independent:

$$\mathbb{E}[X] = \sum_{i=1}^{10} \mathbb{E}[B_i] = 10(0.3) = 3.$$

Variance adds too, but only because the flips are *independent*; using $\text{Var}(B_i) = p(1-p) = 0.3(0.7) = 0.21$,

$$\text{Var}(X) = \sum_{i=1}^{10} \text{Var}(B_i) = 10(0.21) = 2.1, \qquad \text{SD} = \sqrt{2.1} \approx 1.45.$$

These are the general binomial formulas $\mathbb{E}[X] = np$ and $\text{Var}(X) = np(1-p)$, recovered from first principles. The mean of 3 heads is the obvious center; the spread of about 1.45 says a run of ten flips will usually land within a head or two of three. Had the flips been correlated, the mean would be unchanged but the variance would not be a simple sum, the same lesson the Gaussian sum rule of Chapter 6 carries into continuous variables. □

**Exercise 5.6.** A game pays you $\$X$, where $X$ is the value rolled on a fair six-sided die, but it costs \$4 to play. Find the expected net gain per play and decide whether the game is favorable.

*Solution.* The payout has expectation $\mathbb{E}[X] = \frac{1+2+3+4+5+6}{6} = 3.5$ (the fair-die mean). The cost of \$4 is a constant, and expectation is linear, so the expected net gain is

$$\mathbb{E}[X - 4] = \mathbb{E}[X] - 4 = 3.5 - 4 = -0.5.$$

You lose half a dollar per play on average, so the game is *unfavorable*; a fair price to play would be exactly \$3.50, the expected payout. This is the logic behind every insurance premium and casino edge: the expected value, rather than any single lucky or unlucky roll, governs the long-run average by the law of large numbers (Chapter 7). A risk-neutral player declines this game; whether a risk-averse or risk-seeking player would is a separate question about utility, not expectation. □

**Exercise 5.7.** Roll two fair dice. Let $A$ be the event "the first die is even" and $B$ the event "the sum is 7." Are $A$ and $B$ independent?

*Solution.* Independence means $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$, so compute all three. The first die is even in half the outcomes, $\mathbb{P}(A) = \frac{1}{2}$. The sum is 7 in six of the 36 equally likely outcomes $((1,6), \ldots, (6,1))$, so $\mathbb{P}(B) = \frac{6}{36} = \frac{1}{6}$. For the intersection, a sum of 7 with an even first die means the first die is $2, 4$, or 6 paired with $5, 3, 1$, three outcomes, so $\mathbb{P}(A \cap B) = \frac{3}{36} = \frac{1}{12}$. Now check: $\mathbb{P}(A)\mathbb{P}(B) = \frac{1}{2} \cdot \frac{1}{6} = \frac{1}{12} = \mathbb{P}(A \cap B)$, so $A$ and $B$ *are* independent. This is mildly surprising, since the first die's value obviously affects the sum; the point is that a target sum of exactly 7 is special. Whatever the first die shows, there is always exactly one second-die value completing a 7, so conditioning on the first die never changes $\mathbb{P}(B)$. Try the sum 6 instead and independence breaks, because 6 cannot be reached from a first die of 6, so the symmetry is lost. □

CHAPTER 6

# Distributions, the Gaussian, and Covariance

*A data vector is semantically empty until you fix the basis, and a random vector is empty until you fix its mean and covariance.*

*Professor Chin*

**Roadmap**

Chapter 5 estimated a single binary parameter. This chapter scales up in two directions. First, from two outcomes to $K$: the categorical and multinomial distributions, their maximum likelihood estimate by Lagrange multipliers, and the Dirichlet prior that generalizes the Beta. Second, from discrete to continuous and from one dimension to many: the multivariate Gaussian, the workhorse distribution of machine learning, whose entire shape is fixed by a mean vector and a covariance matrix. Covariance ties the two halves together, and the spectral theorem of Chapter 3 returns to explain the geometry.

**Suggested reading:** Deisenroth et al. [5], Sections 6.4–6.6; Bishop [2], Section 2.3 for the Gaussian in depth.

## 6.1 From two outcomes to $K$: categorical and multinomial

A single draw from $K$ categories generalizes the Bernoulli.

**Definition 6.1 (Categorical distribution).** A *categorical* variable takes a value $x \in \{1, \dots, K\}$ with probabilities $\boldsymbol{\mu} = (\mu_1, \dots, \mu_K)$, $\mu_k \geq 0$, $\sum_k \mu_k = 1$, so $\mathbb{P}(x = k) = \mu_k$. Encoding the outcome as a one-hot vector $\boldsymbol{x} \in \{0, 1\}^K$ (a single 1 in position $x$) gives the compact form $p(\boldsymbol{x} \mid \boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k}$.

Repeating the draw $N$ times independently and recording only the counts $m_k = \sum_i \mathbf{1}[x_i = k]$ gives the multinomial.

**Definition 6.2 (Multinomial distribution).** The count vector $\boldsymbol{m} = (m_1, \dots, m_K)$ with $\sum_k m_k = N$ has probability

$$p(\boldsymbol{m} \mid \boldsymbol{\mu}, N) = \binom{N}{m_1, \dots, m_K} \prod_{k=1}^{K} \mu_k^{m_k}, \qquad \binom{N}{m_1, \dots, m_K} = \frac{N!}{m_1! \cdots m_K!}, \tag{6.1}$$

the multinomial coefficient counting the orderings that yield the same histogram. With $K = 2$ this is the binomial; with $N = 1$ it is the categorical.

The parameter $\boldsymbol{\mu}$ lives on the *probability simplex* $\Delta^K = \{\boldsymbol{\mu} \in \mathbb{R}^K : \mu_k \geq 0,\ \sum_k \mu_k = 1\}$, a flat $(K-1)$-dimensional triangle (for $K = 3$, an actual triangle). Estimating $\boldsymbol{\mu}$ therefore means optimizing over the simplex, a constrained problem.

## 6.2 Maximum likelihood for the multinomial

Dropping the constant multinomial coefficient, the log-likelihood is

$$\ell(\boldsymbol{\mu}) = \sum_{k=1}^{K} m_k \log \mu_k, \qquad \text{subject to} \quad \sum_{k=1}^{K} \mu_k = 1. \tag{6.2}$$

The constraint forbids maximizing each $\mu_k$ freely, so we use a Lagrange multiplier $\lambda$ for $\sum_k \mu_k = 1$ (the method is developed in full in Chapter 9).

**Proposition 6.3 (Multinomial MLE).** The maximum likelihood estimate is the empirical frequency,

$$\hat{\mu}_k = \frac{m_k}{N}. \tag{6.3}$$

*Proof.* Form the Lagrangian $\mathcal{L}(\boldsymbol{\mu}, \lambda) = \sum_k m_k \log \mu_k + \lambda\big(1 - \sum_k \mu_k\big)$. Setting $\partial \mathcal{L}/\partial \mu_k = m_k/\mu_k - \lambda = 0$ gives $\mu_k = m_k/\lambda$. Summing over $k$ and using the constraint, $1 = \sum_k \mu_k = \frac{1}{\lambda}\sum_k m_k = N/\lambda$, so $\lambda = N$ and $\hat{\mu}_k = m_k/N$. ∎

The MLE simply reports the fraction of trials landing in each category, the multi-class version of $\hat{\mu} = m/n$ from Proposition 5.3. Its weakness is the same: an unobserved category ($m_k = 0$) is assigned probability zero, an overconfident verdict the Bayesian update repairs.

**Example 6.4 (Estimating a die's biases, MLE and smoothed).** Roll a three-sided die $N = 100$ times and observe counts $(m_1, m_2, m_3) = (20, 30, 50)$. The MLE is just the empirical frequencies,

$$\hat{\mu}_k = \frac{m_k}{N}: \qquad \hat{\boldsymbol{\mu}}_{\text{MLE}} = (0.20,\ 0.30,\ 0.50).$$

A Dirichlet($\boldsymbol{\alpha}$) prior with $\boldsymbol{\alpha} = (2, 2, 2)$ adds two pseudo-observations per face; the posterior is Dirichlet$(m_k + \alpha_k) =$ Dirichlet$(22, 32, 52)$, with mean

$$\frac{m_k + \alpha_k}{N + \sum_\ell \alpha_\ell} = \frac{(22, 32, 52)}{106} = (0.208,\ 0.302,\ 0.491).$$

The smoothed estimate barely moves the well-sampled faces but would have rescued a *zero* count: a face never rolled gets MLE probability 0 (it "can never happen"), while the prior assigns it $\alpha_k/(N + \sum \alpha_\ell) > 0$. This *Laplace smoothing* is exactly what keeps a naive-Bayes text classifier (Example 5.7) from collapsing to zero on an unseen word.

## 6.3 The Dirichlet prior

The conjugate prior for the multinomial is the *Dirichlet*, the multivariate Beta.

**Definition 6.5 (Dirichlet distribution).** For concentration parameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)$ with $\alpha_k > 0$ and $\alpha_0 = \sum_k \alpha_k$, the density on the simplex is

$$p(\boldsymbol{\mu} \mid \boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}. \tag{6.4}$$

Multiplying the multinomial likelihood by this prior adds the exponents,

$$p(\boldsymbol{\mu} \mid \boldsymbol{m}) \propto \Big(\prod_k \mu_k^{m_k}\Big)\Big(\prod_k \mu_k^{\alpha_k - 1}\Big) = \prod_k \mu_k^{(\alpha_k + m_k) - 1},$$

so the posterior is again Dirichlet, $\boldsymbol{\mu} \mid \boldsymbol{m} \sim \text{Dirichlet}(\boldsymbol{\alpha} + \boldsymbol{m})$. The posterior mean is

$$\mathbb{E}[\mu_k \mid \boldsymbol{m}] = \frac{\alpha_k + m_k}{\alpha_0 + N}, \tag{6.5}$$

exactly the MLE smoothed by the pseudo-counts $\alpha_k$. Because $\alpha_k > 0$, no category is ever assigned probability zero, curing the MLE's overconfidence.

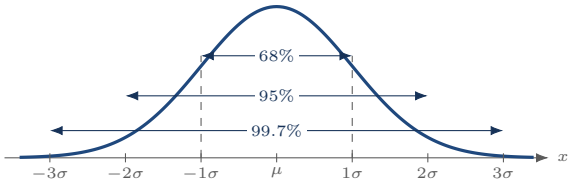



**Figure 6.1.** The one-dimensional Gaussian and the 68–95–99.7 rule: about 68% of the mass lies within $\pm 1\sigma$ of the mean, 95% within $\pm 2\sigma$, and 99.7% within $\pm 3\sigma$. The multivariate version replaces these intervals with the ellipsoidal shells of Fig. 6.3.

**Example 6.6 (Dirichlet smoothing).** Roll a three-sided die $N = 10$ times and observe counts $\boldsymbol{m} = (4, 2, 4)$. With a uniform prior $\boldsymbol{\alpha} = (1, 1, 1)$ ($\alpha_0 = 3$), the posterior is Dirichlet$(5, 3, 5)$ and

$$\mathbb{E}[\mu_1 \mid \boldsymbol{m}] = \frac{4+1}{10+3} = \frac{5}{13}, \quad \mathbb{E}[\mu_2 \mid \boldsymbol{m}] = \frac{3}{13}, \quad \mathbb{E}[\mu_3 \mid \boldsymbol{m}] = \frac{5}{13}.$$

The single pseudo-count per category pulls the estimates gently off the raw frequencies $(0.4, 0.2, 0.4)$ toward uniformity.

## 6.4 The multivariate Gaussian

For continuous data the central model is the Gaussian, and it is worth meeting in one dimension first. A scalar $X \sim \mathcal{N}(\mu, \sigma^2)$ has the bell-shaped density $p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\!\big(-(x-\mu)^2/2\sigma^2\big)$, centered at the mean $\mu$ and with width set by the standard deviation $\sigma$. The familiar 68–95–99.7 rule (Fig. 6.1) says how much mass lies within one, two, and three standard deviations of the mean.

*Intuition.* Why is the Gaussian everywhere? Two reasons. It is the distribution that assumes the least (it has the largest entropy) once you fix a mean and variance, so it encodes "I know the center and spread and nothing more." And by the central limit theorem, a sum of many small independent effects looks Gaussian regardless of their individual shapes, so measurement noise, aggregated errors, and averages drift toward it automatically. That is why least squares (Chapter 2) quietly assumed Gaussian noise and why so much of machine learning is Gaussian underneath.

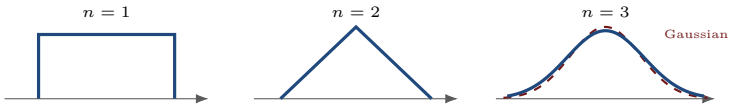



**Figure 6.2.** The central limit theorem in action. The density of the average of $n$ independent uniform draws is flat for $n = 1$, a triangle for $n = 2$, and already hugs the limiting Gaussian (dashed) by $n = 3$. A sum of many small independent effects converges to a Gaussian whatever the shape of each piece.

The multivariate version packages a mean vector and a covariance matrix into the same exponential shape.

**Definition 6.7 (Multivariate Gaussian).** A random vector $\boldsymbol{x} \in \mathbb{R}^d$ is Gaussian, $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, if it has density

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\!\Big(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\Big), \tag{6.6}$$

with mean vector $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{x}] \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{\Sigma} = \mathbb{E}[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\top] \in \mathbb{R}^{d\times d}$, symmetric and positive semidefinite. Here $|\boldsymbol{\Sigma}|$ is the determinant.

The exponent is a quadratic form in $\boldsymbol{x} - \boldsymbol{\mu}$, so the level sets $\{\boldsymbol{x} : (\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = c\}$ are ellipsoids centered at $\boldsymbol{\mu}$. By the spectral theorem applied to $\boldsymbol{\Sigma}$, their principal axes point along the eigenvectors of $\boldsymbol{\Sigma}$, with semi-axis lengths proportional to the square roots of the eigenvalues: large variance in a direction stretches the ellipsoid along it. This is exactly the quadratic-form ellipse of Fig. 3.3, now drawn by $\boldsymbol{\Sigma}^{-1}$.

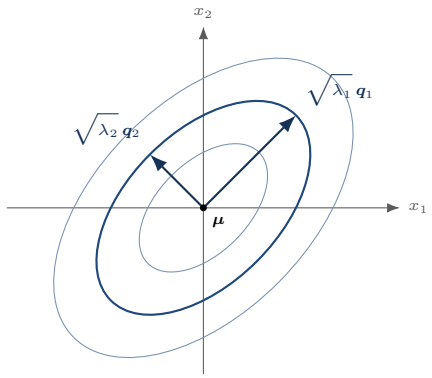



**Figure 6.3.** Density contours of a two-dimensional Gaussian with $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$. The contours are ellipses centered at the mean $\boldsymbol{\mu}$, with axes along the eigenvectors $\boldsymbol{q}_1 = \frac{1}{\sqrt{2}}(1,1)$, $\boldsymbol{q}_2 = \frac{1}{\sqrt{2}}(1,-1)$ of $\boldsymbol{\Sigma}$ and semi-axis lengths $\sqrt{\lambda_1} = \sqrt{3}$ and $\sqrt{\lambda_2} = 1$. Correlation tilts the ellipse off the coordinate axes.

**Definition 6.8 (Mahalanobis distance).** The quantity in the exponent,

$$d_{\boldsymbol{\Sigma}}(\boldsymbol{x}, \boldsymbol{\mu}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}, \tag{6.7}$$

is the *Mahalanobis distance* from $\boldsymbol{x}$ to the mean. It is ordinary Euclidean distance measured in units of standard deviation along each principal axis: a point two standard deviations out in a low-variance direction is "farther," in Mahalanobis terms, than a point the same Euclidean distance out along a high-variance direction. The Gaussian density depends on $\boldsymbol{x}$ only through this distance, so all points on one contour ellipsoid are equally probable. When $\boldsymbol{\Sigma} = I$ it reduces to the Euclidean distance and the contours are spheres.

**Example 6.9 (A 2-D Gaussian).** Let $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$. Then $|\boldsymbol{\Sigma}| = 4 - 1 = 3$ and $\boldsymbol{\Sigma}^{-1} = \frac{1}{3}\left[\begin{smallmatrix} 2 & -1 \\ -1 & 2 \end{smallmatrix}\right]$, so the density is $p(\boldsymbol{x}) = \frac{1}{2\pi\sqrt{3}} \exp\!\left(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x}\right)$. The eigenvalues of $\boldsymbol{\Sigma}$ are $\lambda_1 = 3$ (eigenvector $(1,1)$) and $\lambda_2 = 1$ (eigenvector $(1,-1)$), so the contours are ellipses elongated along the $45^\circ$ line, as in Fig. 6.3. The positive off-diagonal covariance $\Sigma_{12} = 1$ makes $x_1$ and $x_2$ tend to rise and fall together.

**Mahalanobis distance.** Compare two points the same Euclidean distance from the mean is misleading; the right yardstick is Mahalanobis. For $\boldsymbol{x} = (1,1)$ (along the correlation direction) and $\boldsymbol{z} = (2,0)$ (across it),

$$\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x} = \tfrac{1}{3}(1,1)\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \tfrac{1}{3}(1,1)\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \tfrac{2}{3}, \qquad \boldsymbol{z}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{z} = \tfrac{1}{3}(2,0)\begin{bmatrix} 2 \\ -2 \end{bmatrix} = \tfrac{8}{3}.$$

So $(1,1)$ has Mahalanobis distance $\sqrt{2/3} \approx 0.82$ while $(2,0)$ has $\sqrt{8/3} \approx 1.63$: the point lying along the cloud's long axis is *statistically nearer* even though both are a couple of units out in plain Euclidean terms.

**Conditioning.** Fixing $x_2$ gives a Gaussian on $x_1$ with mean $\Sigma_{12}/\Sigma_{22} \cdot x_2 = \frac{1}{2}x_2$ and variance $\Sigma_{11} - \Sigma_{12}^2/\Sigma_{22} = 2 - \frac{1}{2} = \frac{3}{2}$. So observing $x_2 = 2$ updates our belief about $x_1$ to $\mathcal{N}(1, \frac{3}{2})$: the mean shifts toward $\frac{1}{2}(2) = 1$ (the correlation pulls it up) and the variance shrinks from 2 to $\frac{3}{2}$ (knowing $x_2$ tells us something about $x_1$). This is Remark 6.12 on numbers.

The Gaussian is closed under the operations we care about: any affine image is Gaussian,

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \implies A\boldsymbol{x} + \boldsymbol{b} \sim \mathcal{N}(A\boldsymbol{\mu} + \boldsymbol{b},\ A\boldsymbol{\Sigma}A^\top), \tag{6.8}$$

and every marginal and every conditional of a Gaussian is again Gaussian (the conditional mean is a linear function of the conditioning variables). These closure properties, collected in Table 6.1, are why Gaussians are so tractable and why they anchor principal component analysis and Gaussian mixture models in Chapter 11.

**Example 6.10 (Pushing a Gaussian through a linear map).** Let $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\mu} = (1,2)$ and $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 3 \end{smallmatrix}\right]$ (so $x_1, x_2$ are independent with variances 2 and 3). The affine rule says $\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{b}$ is again Gaussian with mean $A\boldsymbol{\mu} + \boldsymbol{b}$ and covariance $A\boldsymbol{\Sigma}A^\top$; let us compute both for two maps.

| Operation | Input | Output |
|---|---|---|
| Affine map | $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ | $A\boldsymbol{x} + \boldsymbol{b} \sim \mathcal{N}(A\boldsymbol{\mu} + \boldsymbol{b},\ A\boldsymbol{\Sigma}A^\top)$ |
| Sum (independent) | $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ | $\sum_i \boldsymbol{x}_i \sim \mathcal{N}(\sum_i \boldsymbol{\mu}_i,\ \sum_i \boldsymbol{\Sigma}_i)$ |
| Marginal | drop some coordinates | Gaussian on the kept coordinates |
| Conditional | fix some coordinates | Gaussian, mean linear in the fixed ones |

**Table 6.1.** The Gaussian is closed under every linear operation we use. No other continuous distribution is this well behaved, which is the practical reason it dominates.

*Sum of the coordinates.* For the scalar $y = x_1 + x_2$, take the row $\boldsymbol{a}^\top = (1, 1)$. The mean is $\boldsymbol{a}^\top\boldsymbol{\mu} = 1 + 2 = 3$, and the variance is

$$\boldsymbol{a}^\top\boldsymbol{\Sigma}\boldsymbol{a} = (1,1)\begin{bmatrix}2 & 0\\ 0 & 3\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} = (1,1)\begin{bmatrix}2\\3\end{bmatrix} = 5,$$

so $y \sim \mathcal{N}(3, 5)$. The variances added because the coordinates are independent, exactly as $\mathrm{Var}(x_1 + x_2) = \mathrm{Var}(x_1) + \mathrm{Var}(x_2)$ predicts.

*A rotation-like map.* For $A = \left[\begin{smallmatrix}1 & 1\\ 1 & -1\end{smallmatrix}\right]$ and $\boldsymbol{b} = \boldsymbol{0}$, the mean is $A\boldsymbol{\mu} = (1{+}2,\ 1{-}2) = (3, -1)$, and the covariance is

$$A\boldsymbol{\Sigma}A^\top = \begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}2 & 0\\ 0 & 3\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = \begin{bmatrix}2 & 3\\ 2 & -3\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = \begin{bmatrix}5 & -1\\ -1 & 5\end{bmatrix}.$$

So $\boldsymbol{y} \sim \mathcal{N}\big((3,-1), \left[\begin{smallmatrix}5 & -1\\ -1 & 5\end{smallmatrix}\right]\big)$. The off-diagonal $-1$ is new: mixing the two independent coordinates has made the outputs *correlated*, even though the inputs were not. Think of shining a Gaussian cloud through a lens (the matrix $A$): it stays a Gaussian cloud, but the lens can stretch it and tilt its axes. This single fact, that Gaussians survive linear maps, is what makes them the workhorse of Chapter 11.

**Example 6.11 (Sums and combinations of independent Gaussians).** The affine rule has a one-dimensional cousin that comes up constantly: independent Gaussians combine into a Gaussian, with means adding always and variances adding only under independence. Let $X \sim \mathcal{N}(1, 4)$ and $Y \sim \mathcal{N}(2, 9)$ be independent, so their standard deviations are $\sigma_X = 2$ and $\sigma_Y = 3$. The sum is

$$X + Y \sim \mathcal{N}(1 + 2,\ 4 + 9) = \mathcal{N}(3,\ 13),$$

with standard deviation $\sqrt{13} \approx 3.61$. The standard deviations did *not* add to 5; the variances added, and the spread grew in quadrature. Any linear combination works the same way. Take $Z = 2X - Y$:

$$\mathbb{E}[Z] = 2(1) - 2 = 0, \qquad \mathrm{Var}(Z) = 2^2\,\mathrm{Var}(X) + (-1)^2\,\mathrm{Var}(Y) = 4(4) + 9 = 25,$$

so $Z \sim \mathcal{N}(0, 25)$ with standard deviation 5. Each coefficient enters the variance *squared*, which is why doubling $X$ quadrupled its contribution, and the cross term dropped out only because $X$ and $Y$ are independent. This is the precise sense in which independent noise sources add in quadrature: combine two independent measurement errors of 2 and 3 units and the result wobbles by $\sqrt{2^2 + 3^2} = \sqrt{13}$ units, not 5.

*Remark* 6.12 *(Conditioning is linear regression).* The conditional-mean fact is more than a curiosity. If $(\boldsymbol{x}, \boldsymbol{y})$ are jointly Gaussian, then $\mathbb{E}[\boldsymbol{y} \mid \boldsymbol{x}] = \boldsymbol{\mu}_{\boldsymbol{y}} + \boldsymbol{\Sigma}_{\boldsymbol{yx}}\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}})$, an affine function of $\boldsymbol{x}$. The best predictor of $\boldsymbol{y}$ from $\boldsymbol{x}$ under a Gaussian model is therefore *linear*, with coefficients $\boldsymbol{\Sigma}_{\boldsymbol{yx}}\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}$ that are exactly the least-squares weights of Chapter 2 written in terms of covariances. Linear regression is what Gaussian conditioning looks like.

### 6.4.1 Estimating the Gaussian

Given iid samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, maximizing the log-likelihood (a matrix-calculus computation carried out in Chapter 8) yields the natural estimates: the sample mean and the sample covariance,

$$\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i, \qquad \hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^\top. \tag{6.9}$$

The Bayesian treatment places a Gaussian prior on $\boldsymbol{\mu}$, and conjugacy makes the posterior Gaussian as well, with a mean that is a precision-weighted average of the prior mean and the sample mean, again converging to $\hat{\boldsymbol{\mu}}$ as $n \to \infty$.

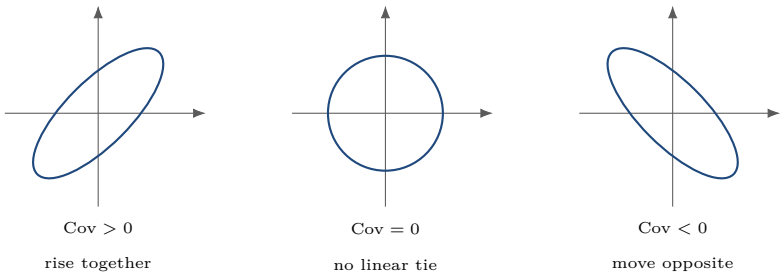



**Figure 6.4.** Covariance is the tilt of the data cloud. Positive covariance tilts the ellipse up the diagonal (the variables rise and fall together), negative covariance tilts it down, and zero covariance leaves it axis-aligned. The covariance matrix $\mathbf{\Sigma}$ records this tilt for every pair of coordinates at once.

**Example 6.13 (Gaussian–Gaussian update: a precision-weighted average).** Make that precision-weighted average concrete in one dimension. Suppose we believe a quantity $\mu$ is around 0 with prior $\mu \sim \mathcal{N}(0, 1)$, then take a single noisy measurement $x = 2$ whose likelihood is $x \mid \mu \sim \mathcal{N}(\mu, 1)$. Precision is the reciprocal of variance, and for the Gaussian–Gaussian pair precisions simply add:

$$\tau_{\text{post}} = \tau_{\text{prior}} + \tau_{\text{lik}} = \frac{1}{1} + \frac{1}{1} = 2, \qquad \sigma^2_{\text{post}} = \frac{1}{\tau_{\text{post}}} = \frac{1}{2}.$$

The posterior mean is the two means averaged in proportion to their precisions:

$$\mu_{\text{post}} = \frac{\tau_{\text{prior}}\,\mu_{\text{prior}} + \tau_{\text{lik}}\,x}{\tau_{\text{prior}} + \tau_{\text{lik}}} = \frac{1 \cdot 0 + 1 \cdot 2}{2} = 1,$$

so the posterior is $\mu \mid x \sim \mathcal{N}(1, \frac{1}{2})$. Two features are worth naming. The posterior mean 1 sits between the prior's 0 and the data's 2, pulled halfway because prior and likelihood are equally precise here; a sharper prior (larger $\tau_{\text{prior}}$) would hold the estimate nearer 0, a more precise measurement nearer 2. And the posterior variance $\frac{1}{2}$ is smaller than either input variance, because combining independent evidence can only sharpen belief. This single update is the scalar heart of the Kalman filter and of Bayesian linear regression: every step there is a Gaussian prior meeting a Gaussian likelihood and averaging by precision.

## 6.5 Covariance

**Definition 6.14 (Covariance).** The *covariance* of two scalar random variables is

$$\operatorname{Cov}(X, Y) = \mathbb{E}\big[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\big] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]. \tag{6.10}$$

It is positive when $X$ and $Y$ tend to deviate from their means in the same direction, negative when in opposite directions, and zero when there is no *linear* relationship. Its self-version is the variance, $\operatorname{Cov}(X, X) = \operatorname{Var}(X)$.

Independence is stronger than zero covariance.

**Proposition 6.15 (Independence implies uncorrelated; the converse can fail).** If $X \perp\!\!\!\perp Y$ then $\operatorname{Cov}(X, Y) = 0$. The converse fails in general.

*Proof.* Independence gives $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$, so $\operatorname{Cov}(X, Y) = 0$ by Eq. (6.10). For the converse, let $X \sim \text{Uniform}(-1, 1)$ and $Y = X^2$. Then $\mathbb{E}[X] = 0$ and $\mathbb{E}[XY] = \mathbb{E}[X^3] = 0$ (an odd moment of a symmetric distribution), so $\operatorname{Cov}(X, Y) = 0$; yet $Y$ is a deterministic function of $X$, so they are about as dependent as two variables can be. Zero covariance only rules out *linear* association. For jointly Gaussian variables, however, zero covariance does imply independence. ∎

## 6.6 The covariance matrix

For a random vector $\boldsymbol{x} \in \mathbb{R}^d$ the covariances of all coordinate pairs assemble into the covariance matrix

$$\boldsymbol{\Sigma} = \mathbb{E}\big[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\top\big], \qquad \Sigma_{ij} = \operatorname{Cov}(x_i, x_j), \tag{6.11}$$

with variances $\operatorname{Var}(x_i)$ on the diagonal and covariances off it. It is symmetric, and it is positive semidefinite.

**Proposition 6.16 (Covariance matrices are PSD).** For any random vector $\boldsymbol{x}$ with covariance matrix $\boldsymbol{\Sigma}$, and any fixed $\boldsymbol{a} \in \mathbb{R}^d$, $\boldsymbol{a}^\top \boldsymbol{\Sigma} \boldsymbol{a} = \operatorname{Var}(\boldsymbol{a}^\top \boldsymbol{x}) \geq 0$. Hence $\boldsymbol{\Sigma} \succeq 0$.

*Proof.* The scalar $\boldsymbol{a}^\top \boldsymbol{x}$ has variance $\operatorname{Var}(\boldsymbol{a}^\top \boldsymbol{x}) = \mathbb{E}[(\boldsymbol{a}^\top(\boldsymbol{x}-\boldsymbol{\mu}))^2] = \mathbb{E}[\boldsymbol{a}^\top(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{a}] = \boldsymbol{a}^\top \boldsymbol{\Sigma} \boldsymbol{a}$, which is nonnegative as the expectation of a square. So $\boldsymbol{a}^\top \boldsymbol{\Sigma} \boldsymbol{a} \geq 0$ for all $\boldsymbol{a}$, the definition of positive semidefinite (Definition 3.23). ∎

Because $\boldsymbol{\Sigma}$ is symmetric PSD, the spectral theorem (Theorem 3.16) gives $\boldsymbol{\Sigma} = Q \Lambda Q^\top$ with orthonormal eigenvectors and nonnegative eigenvalues. The eigenvectors are the *principal axes* of the data's spread and the eigenvalues are the variances along them, which is precisely the content of principal component analysis (Chapter 11). The affine rule Eq. (6.8) specializes to covariances for *any* distribution:

$$\operatorname{Cov}(A\boldsymbol{x} + \boldsymbol{b}) = A \operatorname{Cov}(\boldsymbol{x}) A^\top = A \boldsymbol{\Sigma} A^\top. \tag{6.12}$$

Choosing $A = Q^\top$ rotates the data into the eigenbasis, where the covariance becomes the diagonal $\Lambda$: the coordinates are then uncorrelated. This decorrelation is the engine of PCA.

**Example 6.17 (Sample covariance from data, and its principal axes).** Take five points in $\mathbb{R}^2$: $(1,2), (2,3), (3,5), (4,4), (5,6)$. Their mean is $\bar{\boldsymbol{x}} = \frac{1}{5}(15, 20) = (3, 4)$. Center each point ($\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}$):

$$(-2,-2),\ (-1,-1),\ (0,1),\ (1,0),\ (2,2).$$

The (unbiased) sample covariance sums their outer products and divides by $n - 1 = 4$:

$$\boldsymbol{\Sigma} = \frac{1}{4}\sum_i \tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^\top = \frac{1}{4}\begin{bmatrix} 10 & 9 \\ 9 & 10 \end{bmatrix} = \begin{bmatrix} 2.5 & 2.25 \\ 2.25 & 2.5 \end{bmatrix}.$$

The positive off-diagonal 2.25 says $x_1$ and $x_2$ rise together, as the upward drift of the points shows. Diagonalizing, the eigenvalues are $2.5 \pm 2.25 = \{4.75,\ 0.25\}$ with eigenvectors $\frac{1}{\sqrt{2}}(1,1)$ and $\frac{1}{\sqrt{2}}(1,-1)$. So the data's principal axis is the $45^\circ$ direction $(1,1)$, carrying $4.75/(4.75 + 0.25) = 95\%$ of the total variance, while the perpendicular direction holds the remaining 5%. This eigendecomposition of the covariance *is* principal component analysis (Chapter 11), here done by hand on five points.

**Example 6.18 (From covariance to correlation).** A covariance carries the units of both variables, so its raw size says little on its own: rescale $x_1$ from metres to millimetres and the covariance balloons by a thousand without the relationship changing at all. The *correlation coefficient* strips the units out,

$$\rho = \frac{\operatorname{Cov}(X_1, X_2)}{\sigma_{X_1}\, \sigma_{X_2}},$$

and always lands in $[-1, 1]$. Take the sample covariance just computed, $\boldsymbol{\Sigma} = \left(\begin{smallmatrix} 2.5 & 2.25 \\ 2.25 & 2.5 \end{smallmatrix}\right)$. The standard deviations are $\sigma_{X_1} = \sigma_{X_2} = \sqrt{2.5}$, so

$$\rho = \frac{2.25}{\sqrt{2.5}\,\sqrt{2.5}} = \frac{2.25}{2.5} = 0.9.$$

A value of 0.9 signals a strong positive linear relationship: the five points hug an upward line closely but not perfectly. The extremes are worth holding onto. $\rho = \pm 1$ happens exactly when the points lie on a single straight line, and $\rho = 0$ means no *linear* relationship, though a curved one such as points on a parabola can still hide behind a zero correlation. The useful mental picture: covariance measures co-movement in the original, unit-laden coordinates, while correlation measures the same co-movement after standardizing both axes to unit spread, the way a $z$-score makes two differently-scaled exam distributions comparable.

| Distribution | Support | Parameters | Mean | Conjugate prior |
|---|---|---|---|---|
| Bernoulli | $\{0,1\}$ | $\mu$ | $\mu$ | Beta |
| Binomial | $\{0,\dots,n\}$ | $\mu, n$ | $n\mu$ | Beta |
| Categorical | $\{1,\dots,K\}$ | $\boldsymbol{\mu}$ | – | Dirichlet |
| Multinomial | counts, $\sum = N$ | $\boldsymbol{\mu}, N$ | $N\boldsymbol{\mu}$ | Dirichlet |
| Beta | $[0,1]$ | $a, b$ | $\frac{a}{a+b}$ | – |
| Dirichlet | simplex $\Delta^K$ | $\boldsymbol{\alpha}$ | $\boldsymbol{\alpha}/\alpha_0$ | – |
| Gaussian | $\mathbb{R}^d$ | $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ | $\boldsymbol{\mu}$ | Gaussian (on $\boldsymbol{\mu}$) |

**Table 6.2.** The distributions used in the course. The discrete families pair with a counting conjugate prior (Beta, Dirichlet); the Gaussian is conjugate to itself for the mean. Every mean is the parameter that the corresponding MLE estimates by a frequency or an average.

## 6.7 The distributions at a glance

Table 6.2 collects the distributions of Chapter 5 and this chapter. Reading it top to bottom traces the two generalizations of the coin flip: across (more outcomes) from Bernoulli to categorical, and in aggregate from a single trial to counts. The continuous Gaussian sits apart, the limit that a sum of many such trials approaches.

## 6.8 Summary

- The categorical and multinomial distributions generalize Bernoulli and binomial to $K$ outcomes; the MLE is the empirical frequency $\hat{\mu}_k = m_k/N$, obtained by a Lagrange multiplier for the simplex constraint.
- The Dirichlet is the conjugate prior; the posterior is Dirichlet$(\boldsymbol{\alpha} + \boldsymbol{m})$ with mean $\frac{\alpha_k + m_k}{\alpha_0 + N}$, a smoothed MLE.
- The multivariate Gaussian is set by a mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$; its contours are ellipsoids with axes along the eigenvectors of $\boldsymbol{\Sigma}$, level sets of the Mahalanobis distance $\sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$. It is closed under affine maps, marginalization, and conditioning (Gaussian conditioning *is* linear regression), and its MLE is the sample mean and sample covariance.
- Covariance measures linear association; independence implies zero covariance but not conversely (except for jointly Gaussian variables).
- The covariance matrix is symmetric PSD; its spectral decomposition gives the principal axes of variation, and $\mathrm{Cov}(A\boldsymbol{x} + \boldsymbol{b}) = A\boldsymbol{\Sigma}A^\top$.

## Exercises

**Exercise 6.1.** A four-sided die is rolled 10 times, giving counts $\boldsymbol{m} = (2,3,4,1)$ for faces $A, B, C, D$. (a) Find the maximum likelihood estimate of the face probabilities. (b) Compute the multinomial probability of this exact count vector under $\boldsymbol{\mu} = (0.2, 0.3, 0.4, 0.1)$.

*Solution.* (a) By Proposition 6.3 the MLE is the empirical frequency, $\hat{\boldsymbol{\mu}} = \boldsymbol{m}/N = (0.2, 0.3, 0.4, 0.1)$ (which here equals the $\boldsymbol{\mu}$ in part (b)). (b) The multinomial coefficient is $\binom{10}{2,3,4,1} = \frac{10!}{2!\,3!\,4!\,1!} = \frac{3628800}{2\cdot 6\cdot 24\cdot 1} = 12600$, so

$$p(\boldsymbol{m} \mid \boldsymbol{\mu}) = 12600\,(0.2)^2(0.3)^3(0.4)^4(0.1)^1 = 12600\,(0.04)(0.027)(0.0256)(0.1) \approx 0.0348.$$

About a 3.5% chance of this precise histogram, which is large for a multinomial because $\boldsymbol{\mu}$ matches the observed frequencies exactly. □

**Exercise 6.2.** With a uniform Dirichlet prior $\boldsymbol{\alpha} = (1,1,1)$ and observed counts $\boldsymbol{m} = (4,2,4)$ ($N = 10$), find the posterior distribution and its mean. How does the mean differ from the MLE, and why?

*Solution.* The Dirichlet is conjugate, so the posterior is Dirichlet$(\boldsymbol{\alpha} + \boldsymbol{m})$ = Dirichlet$(5,3,5)$. Its mean is $\mathbb{E}[\mu_k \mid \boldsymbol{m}] = \frac{\alpha_k + m_k}{\alpha_0 + N}$, giving $\left(\frac{5}{13}, \frac{3}{13}, \frac{5}{13}\right) \approx (0.385, 0.231, 0.385)$. The MLE is $\boldsymbol{m}/N = (0.4, 0.2, 0.4)$. The posterior mean is pulled

slightly toward the uniform $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ because each pseudo-count $\alpha_k = 1$ adds imaginary evidence of one observation per class; the effect shrinks as $N$ grows. □

**Exercise 6.3.** For $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, compute $|\boldsymbol{\Sigma}|$ and $\boldsymbol{\Sigma}^{-1}$, find the eigenvalues and eigenvectors of $\boldsymbol{\Sigma}$, and describe the density contours.

*Solution.* The determinant is $|\boldsymbol{\Sigma}| = 2 \cdot 2 - 1 \cdot 1 = 3$, and $\boldsymbol{\Sigma}^{-1} = \frac{1}{3}\left[\begin{smallmatrix} 2 & -1 \\ -1 & 2 \end{smallmatrix}\right]$. The eigenvalues solve $(2-\lambda)^2 - 1 = 0$, so $\lambda = 3, 1$, with eigenvectors $(1,1)/\sqrt{2}$ and $(1,-1)/\sqrt{2}$ respectively. The density contours are ellipses centered at the origin with axes along these eigenvectors; the semi-axis along $(1,1)$ has length proportional to $\sqrt{3}$ and the one along $(1,-1)$ length proportional to 1, so the cloud is stretched along the $45^\circ$ line. The positive covariance makes the two coordinates positively correlated. □

**Exercise 6.4.** Prove that any covariance matrix $\boldsymbol{\Sigma} = \mathbb{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top]$ is symmetric and positive semidefinite.

*Solution.* *Symmetry:* $\Sigma_{ij} = \mathrm{Cov}(x_i, x_j) = \mathrm{Cov}(x_j, x_i) = \Sigma_{ji}$, since the defining product $(x_i - \mu_i)(x_j - \mu_j)$ is symmetric in $i, j$; equivalently $\boldsymbol{\Sigma}^\top = \mathbb{E}[((\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top)^\top] = \boldsymbol{\Sigma}$. *PSD:* for any $\boldsymbol{a} \in \mathbb{R}^d$,

$$\boldsymbol{a}^\top \boldsymbol{\Sigma} \boldsymbol{a} = \mathbb{E}\big[\boldsymbol{a}^\top (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{a}\big] = \mathbb{E}\big[(\boldsymbol{a}^\top (\boldsymbol{x} - \boldsymbol{\mu}))^2\big] = \mathrm{Var}(\boldsymbol{a}^\top \boldsymbol{x}) \geq 0,$$

the expectation of a nonnegative quantity. So $\boldsymbol{a}^\top \boldsymbol{\Sigma} \boldsymbol{a} \geq 0$ for all $\boldsymbol{a}$, which is the definition of $\boldsymbol{\Sigma} \succeq 0$. By Proposition 3.24 all its eigenvalues are therefore nonnegative, as a set of variances should be. □

**Exercise 6.5.** Give a pair of random variables that are uncorrelated but not independent, and verify both claims.

*Solution.* Let $X \sim \mathrm{Uniform}(-1, 1)$ and $Y = X^2$. By symmetry $\mathbb{E}[X] = 0$ and $\mathbb{E}[X^3] = 0$, so

$$\mathrm{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] = \mathbb{E}[X^3] - 0 \cdot \mathbb{E}[X^2] = 0,$$

and the variables are uncorrelated. They are not independent: $Y$ is completely determined by $X$, so for instance $\mathbb{P}(Y < \frac{1}{4} \mid X = 0) = 1 \neq \mathbb{P}(Y < \frac{1}{4})$. This shows zero covariance captures only *linear* dependence; the quadratic dependence here is invisible to it. (Were $(X, Y)$ jointly Gaussian, zero covariance would have forced independence, but $Y = X^2$ is not Gaussian.) □

**Exercise 6.6.** Let $X$ and $Y$ be independent with $\mathrm{Var}(X) = 4$ and $\mathrm{Var}(Y) = 9$. Compute $\mathrm{Var}(2X - 3Y)$ and its standard deviation. Why do the means of $X$ and $Y$ not enter, and would the answer change for $2X + 3Y$?

*Solution.* Variance uses squared coefficients, and for independent variables the cross-covariance term vanishes:

$$\mathrm{Var}(2X - 3Y) = 2^2 \,\mathrm{Var}(X) + (-3)^2 \,\mathrm{Var}(Y) = 4(4) + 9(9) = 16 + 81 = 97,$$

so the standard deviation is $\sqrt{97} \approx 9.85$. The means never appear because variance measures spread *about* the mean, not location: adding a constant to $X$ or $Y$ slides the whole distribution without changing its width, so $\mathrm{Var}(X + c) = \mathrm{Var}(X)$. And because each coefficient is squared, the sign of the $-3$ disappears; under independence $2X + 3Y$ has the identical variance 97. The sign would matter only if $X$ and $Y$ were correlated, through the term $2ab\,\mathrm{Cov}(X, Y)$ that independence here sets to zero. □

**Exercise 6.7.** A two-dimensional Gaussian has covariance $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$. Find the directions and the lengths of the axes of its one-standard-deviation ellipse.

*Solution.* The covariance ellipse has its axes along the eigenvectors of $\boldsymbol{\Sigma}$, with semi-axis lengths equal to the square roots of the eigenvalues. The characteristic polynomial is $(2-\lambda)^2 - 1 = 0$, so $2 - \lambda = \pm 1$ and $\lambda_1 = 3$, $\lambda_2 = 1$. For $\lambda_1 = 3$, $(\boldsymbol{\Sigma} - 3I)\boldsymbol{v} = \left[\begin{smallmatrix} -1 & 1 \\ 1 & -1 \end{smallmatrix}\right]\boldsymbol{v} = \mathbf{0}$ gives the direction $(1,1)$; for $\lambda_2 = 1$ the direction is $(1,-1)$, orthogonal as the spectral theorem promises for a symmetric matrix. The one-sigma ellipse therefore stretches farthest along the $45^\circ$ line $(1,1)$, with semi-axis $\sqrt{\lambda_1} = \sqrt{3} \approx 1.73$, and is narrowest along $(1,-1)$, with semi-axis $\sqrt{\lambda_2} = 1$. The positive off-diagonal tilts the cloud along the diagonal, exactly the elongated, rotated ellipse a scatter of positively-correlated data traces out, and these axes are the principal components of Chapter 11 in disguise. □

**Exercise 6.8.** A call center receives an average of $\lambda = 3$ calls per hour, modeled as a Poisson process. Find the probability of receiving (a) no calls and (b) exactly five calls in a given hour, and state the mean and variance of the count.

*Solution.* The Poisson probability of $k$ events when the rate is $\lambda$ is $\mathbb{P}(X = k) = \dfrac{\lambda^k e^{-\lambda}}{k!}$. (a) With $k = 0$,

$$\mathbb{P}(X = 0) = \frac{3^0 e^{-3}}{0!} = e^{-3} \approx 0.0498,$$

so a fully quiet hour happens about 5% of the time. (b) With $k = 5$,

$$\mathbb{P}(X = 5) = \frac{3^5 e^{-3}}{5!} = \frac{243}{120} e^{-3} = 2.025\,(0.0498) \approx 0.1008,$$

about a 10% chance. The Poisson is the rare-event limit of the binomial (many trials, each unlikely), and it has the memorable property that its mean and variance are both equal to the rate, $\mathbb{E}[X] = \mathrm{Var}(X) = \lambda = 3$, so the standard deviation is $\sqrt{3} \approx 1.73$ calls. A useful sanity check on any Poisson model is exactly this: if a count's sample variance is far from its mean, the data is over- or under-dispersed and Poisson is the wrong model. □

PART IV

# Concentration and Learning Theory

CHAPTER 7

# Concentration Inequalities and Learning Theory

*With enough samples we can guarantee, with overwhelming probability, that the training error reflects the true error.*

*the promise of learning theory*

**Roadmap**

Why does minimizing error on a finite training set say anything about error on unseen data? The answer is *concentration*: averages of independent random variables cluster tightly around their means. We build the tools in increasing sharpness, Markov, then Chebyshev, then the exponentially tight Chernoff bound, proving each. Combined with the union bound they yield generalization guarantees for empirical risk minimization, a sample-complexity rule, the bias–variance tradeoff, and finally the VC dimension, which measures the capacity of an infinite hypothesis class.

**Suggested reading:** Shalev-Shwartz and Ben-David [20], Chapters 4 and 6; Deisenroth et al. [5], Section 8.2 for the learning-theory framing.

## 7.1 Markov's inequality

The crudest tail bound uses only the mean of a nonnegative variable.

**Theorem 7.1 (Markov's inequality).** If $X \geq 0$ is a random variable and $a > 0$, then $\mathbb{P}(X \geq a) \leq \dfrac{\mathbb{E}[X]}{a}$.

*Proof.* Split the expectation at the threshold $a$. Because $X \geq 0$, both pieces below are nonnegative, and on the event $\{X \geq a\}$ the integrand is at least $a$:

$$\mathbb{E}[X] = \mathbb{E}[X\,\mathbf{1}[X < a]] + \mathbb{E}[X\,\mathbf{1}[X \geq a]] \geq \mathbb{E}[X\,\mathbf{1}[X \geq a]] \geq a\,\mathbb{E}[\mathbf{1}[X \geq a]] = a\,\mathbb{P}(X \geq a).$$

Dividing by $a$ gives the bound. ∎

**Example 7.2 (A server's response time).** If requests take $\mathbb{E}[X] = 2$ minutes on average, then without knowing anything else about the distribution, $\mathbb{P}(X \geq 10) \leq 2/10 = 0.2$: at most a fifth of requests can take ten minutes or longer.

## 7.2 Chebyshev's inequality

Bringing in the variance sharpens the bound and lets it apply to two-sided deviations.

**Theorem 7.3 (Chebyshev's inequality).** If $X$ has mean $\mu$ and variance $\sigma^2$, then for any $a > 0$, $\mathbb{P}(|X - \mu| \geq a) \leq \dfrac{\sigma^2}{a^2}$.

*Proof.* Apply Markov's inequality to the nonnegative variable $Y = (X - \mu)^2$, whose mean is $\mathbb{E}[Y] = \mathrm{Var}(X) = \sigma^2$. Since $|X - \mu| \geq a$ exactly when $Y \geq a^2$,

$$\mathbb{P}(|X - \mu| \geq a) = \mathbb{P}(Y \geq a^2) \leq \frac{\mathbb{E}[Y]}{a^2} = \frac{\sigma^2}{a^2}.$$ ■

**Example 7.4 (Test scores).** Scores with $\mu = 70$ and $\sigma = 10$ obey $\mathbb{P}(|X - 70| \geq 20) \leq 10^2/20^2 = 0.25$: at most a quarter of scores lie twenty or more points from the mean, whatever the score distribution. Applied to a sample mean of $n$ iid variables, whose variance is $\sigma^2/n$, Chebyshev gives $\mathbb{P}(|\bar{X} - \mu| \geq a) \leq \sigma^2/(na^2) \to 0$, which is the weak law of large numbers.

**Example 7.5 (Markov versus Chebyshev on the same variable).** A server logs a nonnegative number of errors per day, $X \geq 0$, with mean $\mathbb{E}[X] = 4$ and variance $\mathrm{Var}(X) = 4$. How likely is a very bad day, $X \geq 10$?

*Markov* uses only the mean. For a nonnegative variable and threshold $a > 0$, $\mathbb{P}(X \geq a) \leq \mathbb{E}[X]/a$, so

$$\mathbb{P}(X \geq 10) \leq \frac{\mathbb{E}[X]}{10} = \frac{4}{10} = 0.40.$$

*Chebyshev* also uses the variance. Writing the event in terms of the deviation from the mean, $X \geq 10$ means $X - 4 \geq 6$, which forces $|X - 4| \geq 6$, so

$$\mathbb{P}(X \geq 10) \leq \mathbb{P}(|X - 4| \geq 6) \leq \frac{\mathrm{Var}(X)}{6^2} = \frac{4}{36} = \frac{1}{9} \approx 0.11.$$

Chebyshev's bound (0.11) is more than three times tighter than Markov's (0.40), and the reason is information: Markov knows only *where* the distribution sits (its average), while Chebyshev also knows *how spread out* it is (its variance), so it can be far more confident that the variable will not stray six units out. It is the difference between "the average commute is 4 minutes, so it is at most 40% likely to take 10" and "the commute is 4 minutes give or take 2, so 10 is three standard deviations away and very unlikely." Neither bound assumes any particular distribution; the next section shows that knowing more shape, through the moment-generating function, tightens the tail all the way to exponential.

## 7.3 Why polynomial bounds are not enough

Markov's bound decays like $1/a$ and Chebyshev's like $1/a^2$, far too slowly for the high-confidence guarantees learning theory needs. Consider $n = 1000$ fair coin flips, so $X \sim \mathrm{Binomial}(1000, \frac{1}{2})$ with $\mu = 500$ and $\sigma^2 = 250$. How likely is $X \geq 600$?

$$\text{Markov: } \frac{500}{600} \approx 0.83, \qquad \text{Chebyshev: } \mathbb{P}(|X - 500| \geq 100) \leq \frac{250}{100^2} = 0.025.$$

Markov is useless and Chebyshev still says "up to 2.5%," while the true probability is about $10^{-10}$. We need a bound that decays *exponentially* in the number of trials. That is the Chernoff bound, and its engine is the moment-generating function. Figure 7.1 shows the qualitative gap, and Table 7.1 (filled in over the next two sections) lists what each inequality assumes and how fast it decays.

## 7.4 Moment-generating functions

**Definition 7.6 (Moment-generating function).** The *moment-generating function* (MGF) of a random variable $X$ is $M_X(s) = \mathbb{E}[e^{sX}]$, for $s \in \mathbb{R}$ where the expectation is finite.

Two properties make the MGF the right tool. First, for a Bernoulli variable $Y \sim \mathrm{Bernoulli}(p)$,

$$M_Y(s) = (1 - p) + p\,e^s = 1 + p(e^s - 1) \leq e^{p(e^s - 1)}, \tag{7.1}$$

where the inequality uses $1 + x \leq e^x$ with $x = p(e^s - 1)$. Second, the MGF of a sum of *independent* variables factorizes, since the exponential of a sum is a product and expectation of a product of independents is the product of expectations:

$$X = \sum_{i=1}^{n} X_i \text{ independent} \implies M_X(s) = \mathbb{E}\Big[e^{s\sum_i X_i}\Big] = \prod_{i=1}^{n} \mathbb{E}[e^{sX_i}] = \prod_{i=1}^{n} M_{X_i}(s). \tag{7.2}$$

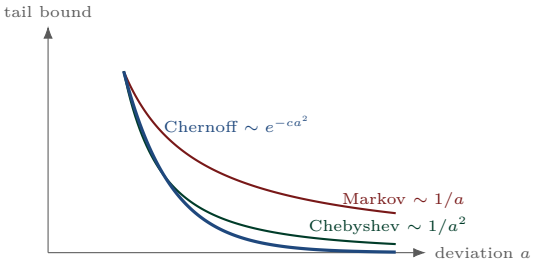



**Figure 7.1.** Why polynomial bounds are not enough (schematic). Markov decays like $1/a$, Chebyshev like $1/a^2$, but the Chernoff bound decays exponentially. For the 1000-coin example the three give 0.83, 0.025, and $\approx 10^{-4}$ at $a$ corresponding to 600 heads. Exponential concentration is what makes finite-sample guarantees possible.

For independent Bernoulli variables with means $p_i$ summing to $\mu$, combining Eq. (7.1) and Eq. (7.2) gives the clean exponential bound $M_X(s) \le \prod_i e^{p_i(e^s-1)} = e^{(e^s-1)\mu}$.

## 7.5 The Chernoff bound

**Theorem 7.7 (Chernoff bound, upper tail).** Let $X = \sum_{i=1}^{n} X_i$ be a sum of independent Bernoulli variables with $\mu = \mathbb{E}[X]$. For any $\delta > 0$,

$$\mathbb{P}\big(X \ge (1+\delta)\mu\big) \le \exp\Big(-\mu\big[(1+\delta)\ln(1+\delta) - \delta\big]\Big) \le e^{-\mu\delta^2/3}. \tag{7.3}$$

A symmetric argument gives $\mathbb{P}(X \le (1-\delta)\mu) \le e^{-\mu\delta^2/2}$, and the two combine into the two-sided bound $\mathbb{P}(|X-\mu| \ge \delta\mu) \le 2e^{-\mu\delta^2/3}$.

*Proof.* The idea is to apply Markov's inequality not to $X$ but to $e^{sX}$, which amplifies large deviations. For any $s > 0$, monotonicity of $e^{s(\cdot)}$ gives

$$\mathbb{P}\big(X \ge (1+\delta)\mu\big) = \mathbb{P}\big(e^{sX} \ge e^{s(1+\delta)\mu}\big) \le \frac{\mathbb{E}[e^{sX}]}{e^{s(1+\delta)\mu}} = \frac{M_X(s)}{e^{s(1+\delta)\mu}},$$

by Markov. Bounding the MGF by $M_X(s) \le e^{(e^s-1)\mu}$ from Section 7.4,

$$\mathbb{P}\big(X \ge (1+\delta)\mu\big) \le \exp\Big(\mu\big[(e^s-1) - s(1+\delta)\big]\Big).$$

This holds for every $s > 0$, so we minimize the exponent $f(s) = (e^s - 1) - s(1+\delta)$. Setting $f'(s) = e^s - (1+\delta) = 0$ gives the optimal $s = \ln(1+\delta) > 0$. Substituting, $e^s - 1 = \delta$ and $s(1+\delta) = (1+\delta)\ln(1+\delta)$, so the exponent becomes $\mu[\delta - (1+\delta)\ln(1+\delta)]$, which is Eq. (7.3). The looser form $e^{-\mu\delta^2/3}$ follows from the Taylor estimate $(1+\delta)\ln(1+\delta) - \delta \ge \delta^2/3$ for $\delta \in (0,1]$. The lower tail repeats the argument with $s < 0$. ∎

**Example 7.8 (The coin, revisited).** For $X \sim \mathrm{Binomial}(1000, \tfrac{1}{2})$ and the event $X \ge 600 = (1+0.2)\mu$, the exact Chernoff bound Eq. (7.3) gives $\exp(-500[(1.2)\ln 1.2 - 0.2]) \approx 8.3 \times 10^{-5}$. Compare Chebyshev's 0.025: the Chernoff bound is hundreds of times tighter, and it correctly reports a vanishingly small probability (the truth is $\approx 10^{-10}$). Exponential concentration is what makes learning from finite data possible.

For bounded variables there is a closely related additive form, *Hoeffding's inequality*: for $Z_1, \dots, Z_m$ independent with $Z_i \in [0,1]$ and sample mean $\bar{Z}$,

$$\mathbb{P}\big(|\bar{Z} - \mathbb{E}[\bar{Z}]| \ge \varepsilon\big) \le 2e^{-2m\varepsilon^2}. \tag{7.4}$$

It is proved by the same exponential-moment method and is the form we use next, because classification errors are exactly bounded $[0,1]$ averages.

**Example 7.9 (How many samples? Hoeffding versus Chebyshev).** We estimate a coin's bias $p$ by the empirical frequency $\hat{p}$ over $m$ flips and want to be 95% sure the estimate is within $\varepsilon = 0.05$, i.e. $\mathbb{P}(|\hat{p} - p| \ge 0.05) \le \delta = 0.05$. How many flips are enough?

| Inequality | Assumes | Bounds | Decay |
|---|---|---|---|
| Markov | $X \geq 0$, mean | $\mathbb{P}(X \geq a) \leq \mathbb{E}[X]/a$ | $1/a$ |
| Chebyshev | mean, variance | $\mathbb{P}(\lvert X-\mu\rvert \geq a) \leq \sigma^2/a^2$ | $1/a^2$ |
| Chernoff | sum of indep. Bernoulli | $\leq 2e^{-\mu\delta^2/3}$ | exponential |
| Hoeffding | indep., bounded in $[0,1]$ | $\leq 2e^{-2m\varepsilon^2}$ | exponential |

**Table 7.1.** The concentration inequalities, in increasing order of assumptions and sharpness. More structure (a variance, then independence and boundedness) buys faster decay. Chernoff and Hoeffding are the exponential bounds that learning theory runs on.

**Chebyshev.** The variance of a single flip is $p(1-p) \leq \frac{1}{4}$, so $\mathrm{Var}(\hat{p}) \leq \frac{1}{4m}$, and Chebyshev gives $\mathbb{P}(\lvert\hat{p}-p\rvert \geq \varepsilon) \leq \frac{1}{4m\varepsilon^2}$. Forcing this $\leq \delta$,

$$m \geq \frac{1}{4\delta\varepsilon^2} = \frac{1}{4(0.05)(0.05)^2} = 2000.$$

**Hoeffding.** The exponential bound Eq. (7.4) gives $2e^{-2m\varepsilon^2} \leq \delta$, i.e.

$$m \geq \frac{\ln(2/\delta)}{2\varepsilon^2} = \frac{\ln 40}{2(0.05)^2} = \frac{3.689}{0.005} \approx 738.$$

Hoeffding asks for 738 flips, Chebyshev for 2000: nearly a $3\times$ saving, and the gap only widens as $\delta$ shrinks (Hoeffding scales like $\ln(1/\delta)$, Chebyshev like $1/\delta$). This is the practical payoff of an exponential tail over a polynomial one, and the same counting, applied to a whole hypothesis class, yields the generalization bounds below.

## 7.6 Generalization: training error versus test error

Fix a hypothesis $h : \mathcal{X} \to \{0,1\}$ and draw a training set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ iid from the data distribution $\mathcal{D}$. The *training error* and *test error* are

$$\widehat{\mathrm{err}}(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}[h(\boldsymbol{x}_i) \neq y_i], \qquad \mathrm{err}(h) = \mathbb{P}_{(\boldsymbol{x},y)\sim\mathcal{D}}[h(\boldsymbol{x}) \neq y]. \tag{7.5}$$

The indicators $Z_i = \mathbf{1}[h(\boldsymbol{x}_i) \neq y_i]$ are iid Bernoulli($\mathrm{err}(h)$), so the training error is their sample mean and $\mathbb{E}[\widehat{\mathrm{err}}(h)] = \mathrm{err}(h)$: the training error is an *unbiased* estimate of the test error. Hoeffding's inequality Eq. (7.4) then controls the gap for this single, fixed $h$: $\mathbb{P}(\lvert\widehat{\mathrm{err}}(h) - \mathrm{err}(h)\rvert > \gamma) \leq 2e^{-2m\gamma^2}$.

### 7.6.1 From one hypothesis to a whole class

Learning chooses a hypothesis *after* seeing the data, by empirical risk minimization, $h_{\mathrm{ERM}} = \arg\min_{h\in\mathcal{H}} \widehat{\mathrm{err}}(h)$. Now the bound must hold *simultaneously* for every $h \in \mathcal{H}$, lest the search cherry-pick one that fits by chance. The union bound, $\mathbb{P}(\bigcup_i A_i) \leq \sum_i \mathbb{P}(A_i)$, handles this. For a finite class with $\lvert\mathcal{H}\rvert = K$,

$$\mathbb{P}\Big(\exists h \in \mathcal{H} :\ \lvert\widehat{\mathrm{err}}(h) - \mathrm{err}(h)\rvert > \gamma\Big) \leq K \cdot 2e^{-2m\gamma^2}. \tag{7.6}$$

Setting the right side to $\delta$ and solving for $\gamma$ gives the *uniform convergence* guarantee: with probability at least $1-\delta$,

$$\lvert\widehat{\mathrm{err}}(h) - \mathrm{err}(h)\rvert \leq \sqrt{\frac{\ln(2K/\delta)}{2m}} \quad \text{for all } h \in \mathcal{H}. \tag{7.7}$$

**Proposition 7.10 (Sample complexity and the ERM guarantee).** To make the gap at most $\gamma$ with confidence $1-\delta$, it suffices to take

$$m \geq \frac{\ln(2K/\delta)}{2\gamma^2}. \tag{7.8}$$

Under Eq. (7.7), the ERM hypothesis is near-optimal: $\mathrm{err}(h_{\mathrm{ERM}}) \leq \min_{h\in\mathcal{H}} \mathrm{err}(h) + 2\gamma$.

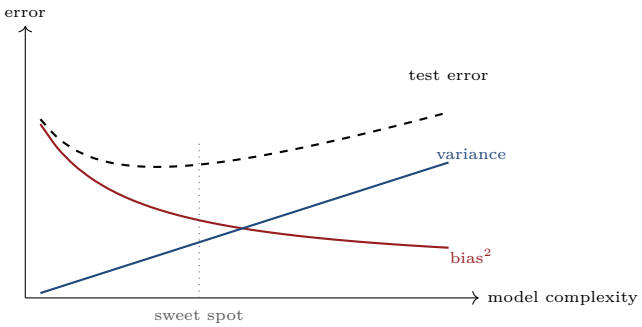



**Figure 7.2.** The bias–variance tradeoff. As the hypothesis class grows more complex, bias falls (it can fit the truth) but variance rises (it can fit the noise). Test error bottoms out at intermediate complexity and climbs toward either extreme.

The sample size grows only *logarithmically* in the number of hypotheses $K$ and in $1/\delta$, but quadratically in $1/\gamma$. For instance, with $K = 1000$, $\gamma = 0.05$, $\delta = 0.05$, Eq. (7.8) gives $m \geq \ln(40000)/(2 \cdot 0.0025) \approx 2120$ samples.

A guarantee of this shape, "with probability at least $1 - \delta$ (*probably*) the error is within $\gamma$ of the best (*approximately correct*)," is called *PAC* learning, for *probably approximately correct*. The two knobs are the confidence $\delta$ and the accuracy $\gamma$, and Eq. (7.8) is the number of samples that buys them. The whole edifice rests on one idea: a fixed hypothesis's training error concentrates around its test error (Hoeffding), and the union bound pays the small price of making that hold for the entire class at once.

## 7.7 The bias–variance tradeoff

The bound Eq. (7.7) says a richer class (larger $K$) costs more samples, hinting at a tension. Decomposing the expected squared error of a prediction $\hat{y}$ for target $y$ makes it precise:

$$\mathbb{E}\big[(y - \hat{y})^2\big] = \underbrace{\big(\mathbb{E}[\hat{y}] - y\big)^2}_{\text{bias}^2} + \underbrace{\mathbb{E}\big[(\hat{y} - \mathbb{E}[\hat{y}])^2\big]}_{\text{variance}} + \underbrace{\sigma^2}_{\text{noise}}. \tag{7.9}$$

A simple class has high bias (it underfits, unable to represent the truth) but low variance; a complex class has low bias but high variance (it overfits, chasing the noise in the particular sample). The union-bound term $\sqrt{\ln K/(2m)}$ in Eq. (7.7) is exactly a variance penalty that grows with capacity, so test error behaves like $\text{bias}^2 + \mathcal{O}(\ln K/m) + \text{noise}$, minimized at intermediate complexity (Fig. 7.2). More data lets you afford a richer class safely.

## 7.8 VC dimension

Infinite hypothesis classes, like all linear classifiers in $\mathbb{R}^d$, have $K = \infty$, so $\ln K$ is useless. The right measure is not how many hypotheses there are but how many points they can label in *every* possible way.

**Definition 7.11 (Shattering and VC dimension).** A class $\mathcal{H}$ *shatters* a set of points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ if for every one of the $2^n$ binary labelings there is some $h \in \mathcal{H}$ realizing it. The *Vapnik–Chervonenkis dimension* $d_{\text{VC}}(\mathcal{H})$ is the size of the largest set $\mathcal{H}$ can shatter.

**Example 7.12 (Linear classifiers).** For lines in $\mathbb{R}^2$, $h_{\boldsymbol{w},b}(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$, three points in general position can be split in all $2^3 = 8$ ways (Fig. 7.3), but four points cannot (the XOR labeling, diagonal pairs same-class, defeats every line). So $d_{\text{VC}} = 3$. In general, linear classifiers in $\mathbb{R}^d$ have $d_{\text{VC}} = d + 1$: the $d$ weights set the orientation and the bias sets the offset of the separating hyperplane.

**Example 7.13 (Intervals on the line: $d_{\text{VC}} = 2$).** The simplest shattering argument is for the class $\mathcal{H} = \{\mathbf{1}[x \in [a, b]]\}$ of intervals on $\mathbb{R}$ (label + inside, − outside). **Two points can be shattered.** For $x_1 < x_2$, all $2^2 = 4$ labelings are realizable: $(-,-)$ by an interval to the left of both, $(+,-)$ by $[x_1 - \epsilon, \frac{x_1+x_2}{2}]$, $(-,+)$ by an interval

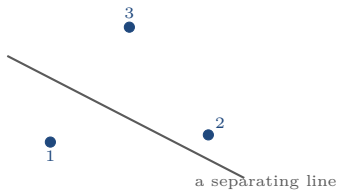



**Figure 7.3.** Three points in general position in $\mathbb{R}^2$. A linear classifier can realize all $2^3 = 8$ labelings (the line shown separates one labeling; tilting and shifting it produces the rest), so the points are shattered. No line can shatter four points, e.g. the XOR pattern. Hence $d_{\mathrm{VC}} = 3$ for lines in $\mathbb{R}^2$.

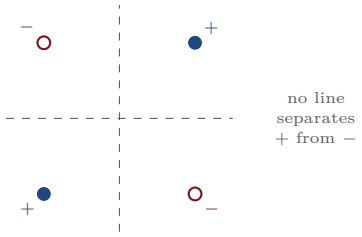



**Figure 7.4.** The XOR labeling defeats every line. The two filled + points sit on one diagonal, the two open − points on the other; no straight line can put both diagonals' endpoints on the correct sides. So four points cannot be shattered by lines, confirming $d_{\mathrm{VC}} = 3$ for linear classifiers in $\mathbb{R}^2$.

around $x_2$ only, and $(+,+)$ by $[x_1, x_2]$. So $d_{\mathrm{VC}} \geq 2$. **Three points cannot.** For $x_1 < x_2 < x_3$, the labeling $(+,-,+)$ would need an interval containing $x_1$ and $x_3$ but *not* the $x_2$ between them, which is impossible because an interval is connected. One labeling out of eight fails, so no 3-point set is shattered and $d_{\mathrm{VC}} = 2$. This matches the two free parameters $a, b$, and the same "a connected set can't skip a middle point" idea is why intervals generalize better than, say, arbitrary unions of intervals.

**Example 7.14 (Axis-aligned rectangles: $d_{\mathrm{VC}} = 4$).** A classifier that labels + inside an axis-aligned rectangle and − outside can shatter four points but not five. **Four points.** Place them at the compass tips, e.g. $(0,1), (0,-1), (1,0), (-1,0)$. For any subset to be labeled +, choose the rectangle to be the bounding box of just those points; because each point is the unique extreme one in its direction (top, bottom, right, left), the box can always include exactly the chosen ones and exclude the rest. All $2^4 = 16$ labelings are realizable, so $d_{\mathrm{VC}} \geq 4$. **Five points.** Given any five points, take the one that is not extreme in any of the four directions (some interior or non-extreme point always exists). Label the four extreme points + and that fifth point −. Any rectangle containing the four extremes must contain their bounding box, which contains the fifth point too, so it cannot exclude it. That labeling fails, so no five points are shattered and $d_{\mathrm{VC}} = 4$, matching the four free parameters (the rectangle's four sides).

The VC dimension replaces $\ln K$ in the generalization bound: for a class of VC dimension $d_{\mathrm{VC}}$, with high probability the gap between training and test error is $\mathcal{O}\big(\sqrt{d_{\mathrm{VC}}\,(\ln(m/d_{\mathrm{VC}}))/m}\big)$. A finite VC dimension makes even an infinite class learnable, and it is the honest measure of capacity in the bias–variance picture: a larger $d_{\mathrm{VC}}$ lowers bias but demands more data to keep variance in check.

| Hypothesis class | $d_{\mathrm{VC}}$ | Free parameters |
|---|---|---|
| Thresholds $\mathbf{1}[x \geq t]$ on $\mathbb{R}$ | 1 | 1 ($t$) |
| Intervals $\mathbf{1}[x \in [a, b]]$ on $\mathbb{R}$ | 2 | 2 ($a, b$) |
| Linear classifiers in $\mathbb{R}^d$ | $d + 1$ | $d + 1$ ($\boldsymbol{w}, b$) |
| Axis-aligned rectangles in $\mathbb{R}^2$ | 4 | 4 (sides) |

**Table 7.2.** VC dimension for common classes. It usually tracks the number of free parameters, formalizing the intuition that capacity is "degrees of freedom." The generalization bound replaces $\ln K$ with $d_{\mathrm{VC}}$, so a finite VC dimension makes even an infinite class learnable.

## 7.9 Sample complexity in practice

The bounds above are not just asymptotics; they give usable sample-size numbers. The worked examples here turn each inequality into a concrete count, and Fig. 7.5 shows the learning curve those bounds predict.

**Example 7.15 (How many samples for a finite hypothesis class).** Suppose a learner chooses among $K = 100$ candidate rules and the true rule is among them (the realizable case). How many labeled examples guarantee, with probability at least 0.95, that the empirically best rule has true error below $\varepsilon = 0.1$? The realizable PAC bound says it suffices that

$$m \geq \frac{1}{\varepsilon}\left(\ln K + \ln \tfrac{1}{\delta}\right) = \frac{1}{0.1}\big(\ln 100 + \ln 20\big) = 10(4.605 + 2.996) = 76.0,$$

so $m = 77$ examples suffice. The dependence is gentle: the class size $K$ and the confidence $\delta$ enter only through logarithms, so growing the hypothesis pool from 100 to 1000 adds just $\ln 10 \approx 2.3$ to the bracket, about 23 more samples. Accuracy is the expensive axis, entering as $1/\varepsilon$.

**Example 7.16 (The uniform generalization gap).** With $K = 1000$ hypotheses and $m = 1000$ samples, how tightly does training error track true error *simultaneously* for every hypothesis? Combining Hoeffding with a union bound over the $K$ rules gives, with probability $1 - \delta$,

$$\sup_h \big|\hat{R}(h) - R(h)\big| \leq \sqrt{\frac{\ln(2K/\delta)}{2m}} = \sqrt{\frac{\ln(2 \cdot 1000/0.05)}{2000}} = \sqrt{\frac{10.6}{2000}} \approx 0.073.$$

So no hypothesis's training error misleads by more than about 7.3 percentage points. The union bound is what makes the guarantee *uniform*: it pays a $\ln K$ price to protect against the worst of the $K$ rules being the one we pick, the multiple-comparisons tax of searching a large model space.

**Example 7.17 (The law of large numbers, quantified).** A quantity has standard deviation $\sigma = 10$. How many independent samples drive the standard error of the mean below 1? The sample mean has standard error $\sigma/\sqrt{m}$, so we need $10/\sqrt{m} \leq 1$, i.e.

$$\sqrt{m} \geq 10 \implies m \geq 100.$$

A hundred samples cut the spread tenfold, from 10 to 1. The $\sqrt{m}$ in the denominator is the law of large numbers made quantitative: to halve the error you must *quadruple* the data, the diminishing return that governs every Monte Carlo estimate and every opinion poll.

**Example 7.18 (A Hoeffding confidence interval).** A bounded loss in $[0, 1]$ is averaged over $m = 100$ held-out points. How wide is a 95% confidence interval for the true risk? Hoeffding gives a half-width

$$\varepsilon = \sqrt{\frac{\ln(2/\delta)}{2m}} = \sqrt{\frac{\ln(2/0.05)}{200}} = \sqrt{\frac{3.689}{200}} \approx 0.136.$$

So the test estimate pins the true risk to within about $\pm 0.136$ with 95% confidence, *without any distributional assumption* beyond boundedness. This is exactly the error bar one should report next to a held-out accuracy, and it shrinks as $1/\sqrt{m}$ just like the standard error.

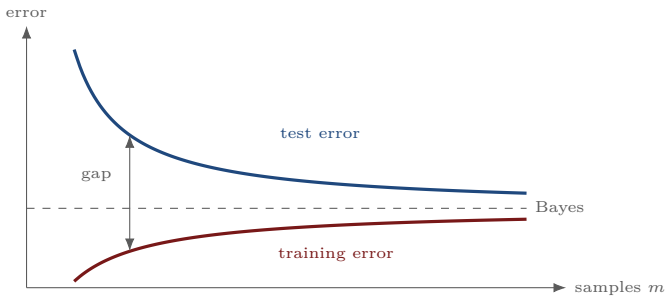



**Figure 7.5.** The learning curve the bounds predict. Training error rises from zero as the model can no longer fit every point, test error falls as more data constrains it, and the gap between them, bounded by the generalization results, closes at the $1/\sqrt{m}$ rate. Both approach the irreducible Bayes error from opposite sides.

## 7.10 Summary

- Markov: $\mathbb{P}(X \geq a) \leq \mathbb{E}[X]/a$ for $X \geq 0$. Chebyshev follows by applying it to $(X-\mu)^2$: $\mathbb{P}(|X-\mu| \geq a) \leq \sigma^2/a^2$.
- These decay polynomially. The Chernoff bound, via the MGF and Markov on $e^{sX}$ optimized over $s$, decays *exponentially*: $\mathbb{P}(|X-\mu| \geq \delta\mu) \leq 2e^{-\mu\delta^2/3}$. Hoeffding gives the additive form $2e^{-2m\varepsilon^2}$ for bounded averages.
- Training error is an unbiased estimate of test error; the union bound makes the estimate uniform over a hypothesis class, giving the gap $\sqrt{\ln(2K/\delta)/(2m)}$ and sample complexity $m \geq \ln(2K/\delta)/(2\gamma^2)$. ERM is then within $2\gamma$ of the best hypothesis.
- Test error decomposes into bias$^2$ + variance + noise; complexity trades bias against variance, minimized at intermediate capacity.
- For infinite classes, the VC dimension (largest shatterable set) replaces $\ln K$; linear classifiers in $\mathbb{R}^d$ have $d_{\mathrm{VC}} = d+1$.

## Exercises

**Exercise 7.1.** A nonnegative random variable has mean $\mu = 5$ and variance $\sigma^2 = 4$. (a) Bound $\mathbb{P}(X \geq 20)$ with Markov. (b) Bound $\mathbb{P}(|X-5| \geq 6)$ with Chebyshev. (c) Which used more information, and which is tighter here?

*Solution.* (a) Markov: $\mathbb{P}(X \geq 20) \leq \mu/20 = 5/20 = 0.25$. (b) Chebyshev: $\mathbb{P}(|X-5| \geq 6) \leq \sigma^2/6^2 = 4/36 = 1/9 \approx 0.111$. (c) Chebyshev uses the variance as well as the mean, and here it bounds a different (two-sided) event; on the comparable upper-tail event it is generally tighter because it exploits the extra information. Markov needs only nonnegativity and the mean, so it applies more broadly but bounds more loosely. □

**Exercise 7.2.** For $n = 1000$ fair coin flips, $X \sim \mathrm{Binomial}(1000, \frac{1}{2})$, compare the Chebyshev and Chernoff bounds on $\mathbb{P}(X \geq 600)$.

*Solution.* Here $\mu = 500$, $\sigma^2 = np(1-p) = 250$, and $600 = (1+\delta)\mu$ with $\delta = 0.2$. Chebyshev: $\mathbb{P}(|X-500| \geq 100) \leq 250/100^2 = 0.025$. Chernoff Eq. (7.3): $\mathbb{P}(X \geq 600) \leq \exp(-500[(1.2)\ln 1.2 - 0.2]) \approx 8.3 \times 10^{-5}$. The Chernoff bound is roughly 300 times smaller, and far closer to the true value ($\approx 10^{-10}$): exponential versus polynomial decay in the deviation. □

**Exercise 7.3.** A finite hypothesis class has $K = 1000$ members. How many iid samples guarantee that, with probability at least 0.95, every hypothesis has training error within $\gamma = 0.05$ of its true error?

*Solution.* Use Eq. (7.8) with $\delta = 0.05$:

$$m \geq \frac{\ln(2K/\delta)}{2\gamma^2} = \frac{\ln(2 \cdot 1000/0.05)}{2(0.05)^2} = \frac{\ln(40000)}{0.005} \approx \frac{10.60}{0.005} \approx 2120.$$

About 2120 samples suffice. Note the mild dependence on $K$ and $\delta$ (both logarithmic) against the strong dependence on accuracy ($1/\gamma^2$): halving the allowed gap quadruples the required sample size. □

**Exercise 7.4.** Let $\mathcal{H}$ be the set of threshold classifiers on the line, $h_t(x) = \mathbf{1}[x \geq t]$ for $t \in \mathbb{R}$. Find $d_{\mathrm{VC}}(\mathcal{H})$.

*Solution.* A single point $x_1$ is shattered: choose $t > x_1$ to label it 0 and $t \leq x_1$ to label it 1, realizing both labelings, so $d_{\mathrm{VC}} \geq 1$. Two points $x_1 < x_2$ cannot be shattered: a threshold labels everything to the right of $t$ as 1, so the labeling $(x_1 \mapsto 1,\ x_2 \mapsto 0)$ is impossible (it would need $x_1 \geq t$ but $x_2 < t$, contradicting $x_1 < x_2$). Hence no set of two points is shattered and $d_{\mathrm{VC}}(\mathcal{H}) = 1$. Thresholds are a one-parameter family, consistent with the rule of thumb that VC dimension tracks the number of free parameters. □

**Exercise 7.5.** Explain, using the decomposition Eq. (7.9) and the generalization bound Eq. (7.7), why fitting a degree-20 polynomial to 15 data points generalizes poorly, and what changes if you collect far more data.

*Solution.* A degree-20 polynomial class is very expressive, so its *bias* is small: it can pass near almost any target shape. But with only $m = 15$ points it has more than enough freedom to interpolate the sample, including its noise, so the *variance* term is large, the model swings wildly between data points, and test error is dominated by variance (overfitting). In the bound Eq. (7.7) the effective capacity (here the polynomial degree, a VC-dimension-like quantity) is large while $m$ is small, so the gap $\sqrt{(\text{capacity})/m}$ is large and the training error badly understates the test error. Collecting far more data shrinks that gap: with large $m$ the variance term falls, the expressive class can be fit reliably, and the low bias finally pays off. This is the quantitative form of "more data lets you safely use a more complex model." □

**Exercise 7.6.** You evaluate 10 candidate models on a validation set. Each individually has at most a 1% chance of looking good by pure luck when it is actually no better than random. Bound the probability that *at least one* of the ten looks good by luck.

*Solution.* Let $E_i$ be the event "model $i$ looks good by luck," with $\mathbb{P}(E_i) \leq 0.01$. The union bound (Boole's inequality) says the probability of *any* event in a collection is at most the sum of their probabilities, with no independence assumption needed:

$$\mathbb{P}\Big(\bigcup_{i=1}^{10} E_i\Big) \leq \sum_{i=1}^{10} \mathbb{P}(E_i) \leq 10(0.01) = 0.10.$$

So there is at most a 10% chance that some model fools you. This is exactly the "multiple comparisons" problem: testing more hypotheses inflates the chance that one passes by accident, which is why a finite hypothesis class of size $K$ pays a $\log K$ price in the generalization bound (Exercise 7.3), and why honest practice corrects the threshold (for instance Bonferroni, dividing the allowed error by the number of tests) before celebrating the best of many models. □

PART V

# Vector Calculus and Optimization

CHAPTER 8

# Vector Calculus, Backpropagation, and Gradient Descent

*The gradient is a compass needle pointing uphill. To learn, step the other way.*

*the one-line summary of optimization*

**Roadmap**

Every model in the course is trained by minimizing a loss, and minimization needs derivatives. This chapter builds the calculus of vectors and matrices: the gradient and Jacobian, the handful of identities used everywhere, and the chain rule that, run in reverse, becomes *backpropagation*. The Hessian and convexity then explain *why* following the negative gradient works, with a clean payoff: the linear-regression cost is convex, so gradient descent reaches its one global minimum. We finish with the optimizers that train real models, momentum, RMSProp, and Adam, and with logistic regression as a convex loss.

**Suggested reading:** Deisenroth et al. [5], Chapter 5; Goodfellow et al. [9], Chapters 6 and 8 for backpropagation and optimizers; Nocedal and Wright [16] for the optimization theory.

## 8.1 Gradients and Jacobians

**Definition 8.1 (Gradient and Jacobian).** For a scalar field $f : \mathbb{R}^d \to \mathbb{R}$, the *gradient* is the column vector of partial derivatives $\nabla f(\boldsymbol{x}) = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_d}\right)^\top \in \mathbb{R}^d$. For a vector field $F : \mathbb{R}^n \to \mathbb{R}^m$ with components $F_i$, the *Jacobian* is the $m \times n$ matrix of all first partials,

$$\boldsymbol{J}_{ij} = \frac{\partial F_i}{\partial x_j}, \qquad \boldsymbol{J} \in \mathbb{R}^{m \times n}.$$

The gradient is the special case $m = 1$ (transposed to a column).

The gradient points in the direction of steepest increase of $f$, and its magnitude is the rate of that increase. Both facts are made precise by the directional derivative in Section 8.4.

**Example 8.2 (A Jacobian and the chain rule on numbers).** Let $F : \mathbb{R}^2 \to \mathbb{R}^2$ be $F(x, y) = (x^2 + y,\ xy)$. Its Jacobian collects the four partial derivatives, $\partial F_1/\partial x = 2x$, $\partial F_1/\partial y = 1$, $\partial F_2/\partial x = y$, $\partial F_2/\partial y = x$, so

$$\boldsymbol{J}_F(x, y) = \begin{bmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial y} \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x & 1 \\ y & x \end{bmatrix}, \qquad \boldsymbol{J}_F(2, 3) = \begin{bmatrix} 4 & 1 \\ 3 & 2 \end{bmatrix}.$$

Now feed in a curve $g(t) = (t,\ t^2)$, so $x = t$ and $y = t^2$. The chain rule says the velocity of the output is the Jacobian times the velocity of the input, $\frac{d}{dt} F(g(t)) = \boldsymbol{J}_F(g(t))\, g'(t)$, with $g'(t) = (1,\ 2t)$ and $\boldsymbol{J}_F(t, t^2) = \left[\begin{smallmatrix} 2t & 1 \\ t^2 & t \end{smallmatrix}\right]$:

$$\frac{d}{dt} F(g(t)) = \begin{bmatrix} 2t & 1 \\ t^2 & t \end{bmatrix} \begin{bmatrix} 1 \\ 2t \end{bmatrix} = \begin{bmatrix} 2t + 2t \\ t^2 + 2t^2 \end{bmatrix} = \begin{bmatrix} 4t \\ 3t^2 \end{bmatrix}.$$

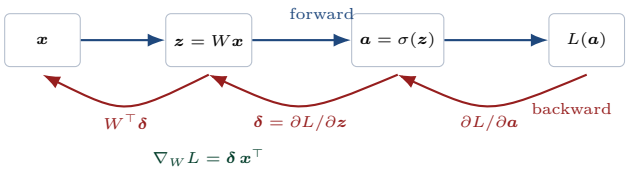



**Figure 8.1.** The computational graph of one layer. The forward pass (blue) computes $\boldsymbol{z}, \boldsymbol{a}, L$ left to right; the backward pass (red) propagates the error $\boldsymbol{\delta} = \partial L/\partial \boldsymbol{z}$ right to left, and the weight gradient $\nabla_W L = \boldsymbol{\delta}\boldsymbol{x}^\top$ (green) falls out at each node. Deep networks chain many such blocks.

Check it directly: $F(g(t)) = (t^2 + t^2,\ t \cdot t^2) = (2t^2,\ t^3)$, whose derivative is indeed $(4t,\ 3t^2)$. The matrix multiplication *is* the chain rule; reading it right to left, $g'$ pushes the input forward and $\boldsymbol{J}_F$ turns that into output motion, the same pattern backpropagation runs in reverse (Section 8.3).

## 8.2 Matrix calculus identities

A short table of identities covers almost every gradient in machine learning. Throughout, $\boldsymbol{a}, \boldsymbol{b}$ are constant vectors, $A$ a constant matrix, and $\boldsymbol{x}$ the variable.

**The identities to memorize**

$$\nabla_{\boldsymbol{x}} (\boldsymbol{a}^\top \boldsymbol{x}) = \boldsymbol{a}, \qquad \nabla_{\boldsymbol{x}} (\boldsymbol{x}^\top A \boldsymbol{x}) = (A + A^\top)\boldsymbol{x}, \tag{8.1}$$

$$\nabla_{\boldsymbol{x}} (\boldsymbol{x}^\top A \boldsymbol{x}) = 2A\boldsymbol{x} \ \ (A \text{ symmetric}), \qquad \nabla_{\boldsymbol{x}} \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 = 2A^\top (A\boldsymbol{x} - \boldsymbol{b}). \tag{8.2}$$

The last identity, the gradient of a squared residual, is the one behind least squares (Chapter 2) and is worth deriving once. Expanding $\|A\boldsymbol{x} - \boldsymbol{b}\|^2 = \boldsymbol{x}^\top A^\top A \boldsymbol{x} - 2\boldsymbol{b}^\top A\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{b}$ and applying the first two identities with the symmetric matrix $A^\top A$ gives $\nabla = 2A^\top A\boldsymbol{x} - 2A^\top \boldsymbol{b} = 2A^\top (A\boldsymbol{x} - \boldsymbol{b})$. Trace identities such as $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ and $\nabla_A \operatorname{tr}(A^\top B) = B$ extend the toolkit to derivatives with respect to matrices, used in Section 8.7.

## 8.3 The chain rule and backpropagation

Composite functions differentiate by the *chain rule*. If $\boldsymbol{z} = g(\boldsymbol{x})$ and $\boldsymbol{y} = f(\boldsymbol{z})$, the Jacobian of the composition is the product of Jacobians, $\boldsymbol{J}_{f\circ g}(\boldsymbol{x}) = \boldsymbol{J}_f(\boldsymbol{z})\, \boldsymbol{J}_g(\boldsymbol{x})$, the multivariate generalization of $(f \circ g)' = f' \cdot g'$. A neural network is a deep composition, and applying the chain rule from the output backward is exactly *backpropagation* [19].

*Intuition.* Backpropagation is the chain rule plus bookkeeping. A forward pass computes and stores each layer's output; the backward pass then multiplies Jacobians from the loss back toward the inputs, reusing those stored values. The reuse is the whole trick: computing the gradients separately for each of a million parameters would be hopeless, but sweeping backward once produces all of them together, at about the cost of a second forward pass (Fig. 8.1).

**Example 8.3 (Backpropagation through one layer).** Consider a single layer with weights $W$, input $\boldsymbol{x}$, pre-activation $\boldsymbol{z} = W\boldsymbol{x}$, activation $\boldsymbol{a} = \sigma(\boldsymbol{z})$ (applied elementwise), and scalar loss $L(\boldsymbol{a})$. To update $W$ we need $\nabla_W L$. Work backward:

$$\underbrace{\boldsymbol{\delta} := \frac{\partial L}{\partial \boldsymbol{z}} = \frac{\partial L}{\partial \boldsymbol{a}} \odot \sigma'(\boldsymbol{z})}_{\text{output error, propagated through } \sigma}, \qquad \nabla_W L = \boldsymbol{\delta}\, \boldsymbol{x}^\top, \qquad \frac{\partial L}{\partial \boldsymbol{x}} = W^\top \boldsymbol{\delta}.$$

The quantity $\boldsymbol{\delta}$ is the error signal at the layer's input; the weight gradient is its outer product with the layer's input, and $W^\top \boldsymbol{\delta}$ passes the error to the previous layer. Chaining these three formulas through every layer, from the loss back to the first weights, computes all gradients in one backward sweep, reusing each layer's stored forward values. This reuse is why training a network of millions of parameters costs only about twice a forward pass, and it is the algorithm that makes deep learning feasible.

**Example 8.4 (Backpropagation on real numbers).** Take a tiny two-layer scalar network on input $x = 2$. Layer one has weight $w_1 = 1$, bias $b_1 = 0$, and a ReLU; layer two has weight $w_2 = \frac{1}{2}$, bias $b_2 = -1$ and a linear output. The target is $y = 1$ and the loss is $L = \frac{1}{2}(\hat{y} - y)^2$.

**Forward pass** (left to right, storing each value):

$$z_1 = w_1 x + b_1 = 2, \quad a_1 = \mathrm{ReLU}(z_1) = 2, \quad z_2 = w_2 a_1 + b_2 = 0, \quad \hat{y} = z_2 = 0, \quad L = \tfrac{1}{2}(0-1)^2 = \tfrac{1}{2}.$$

**Backward pass** (right to left, chain rule). Start at the loss: $\frac{\partial L}{\partial \hat{y}} = \hat{y} - y = -1$. Through layer two,

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \hat{y}}\, a_1 = (-1)(2) = -2, \qquad \frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial \hat{y}} = -1, \qquad \frac{\partial L}{\partial a_1} = \frac{\partial L}{\partial \hat{y}}\, w_2 = (-1)\tfrac{1}{2} = -\tfrac{1}{2}.$$

Through the ReLU, $\mathrm{ReLU}'(z_1) = 1$ since $z_1 = 2 > 0$, so the error at $z_1$ is $\delta_1 = \frac{\partial L}{\partial a_1}\,\mathrm{ReLU}'(z_1) = -\frac{1}{2}$, and through layer one,

$$\frac{\partial L}{\partial w_1} = \delta_1\, x = (-\tfrac{1}{2})(2) = -1, \qquad \frac{\partial L}{\partial b_1} = \delta_1 = -\tfrac{1}{2}.$$

A gradient-descent step with rate $\alpha = 0.1$ would update $w_2 \leftarrow \frac{1}{2} - 0.1(-2) = 0.7$ and $w_1 \leftarrow 1 - 0.1(-1) = 1.1$, both moving to *raise* $\hat{y}$ toward the target $y = 1$, exactly as a single step should. Notice every backward quantity reused a value stored on the forward pass ($a_1$, $z_1$, $x$); that reuse is the whole efficiency of backpropagation.

## 8.4 Directional derivatives and steepest descent

**Definition 8.5 (Directional derivative).** The rate of change of $f$ at $\boldsymbol{x}$ along a unit vector $\boldsymbol{u}$ is $D_{\boldsymbol{u}} f(\boldsymbol{x}) = \nabla f(\boldsymbol{x})^\top \boldsymbol{u}$.

By the Cauchy–Schwarz inequality (Theorem 1.3), $D_{\boldsymbol{u}} f = \nabla f^\top \boldsymbol{u}$ is largest when $\boldsymbol{u}$ aligns with $\nabla f$ and smallest (most negative) when $\boldsymbol{u} = -\nabla f / \|\nabla f\|$. So the negative gradient is the direction of *steepest descent*: among all directions, stepping along $-\nabla f$ decreases $f$ fastest to first order. This single fact is the engine of gradient descent (Section 8.8).

## 8.5 The multivariate Taylor expansion

Near a point $\boldsymbol{x}$, a twice-differentiable $f$ is approximated by its second-order Taylor expansion,

$$f(\boldsymbol{x} + \boldsymbol{h}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top \boldsymbol{h} + \tfrac{1}{2}\, \boldsymbol{h}^\top \nabla^2 f(\boldsymbol{x})\, \boldsymbol{h} + o(\|\boldsymbol{h}\|^2), \tag{8.3}$$

where $\nabla^2 f$ is the Hessian (Section 8.6). The linear term is the gradient's first-order change; the quadratic term, governed by the Hessian, is the curvature. This expansion underlies both the optimality conditions below and second-order methods like Newton's.

*Newton's method* minimizes the quadratic model exactly at each step: setting the gradient of Eq. (8.3) to zero gives the step $\boldsymbol{h} = -[\nabla^2 f(\boldsymbol{x})]^{-1} \nabla f(\boldsymbol{x})$, so $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - [\nabla^2 f(\boldsymbol{x}_t)]^{-1} \nabla f(\boldsymbol{x}_t)$. Where gradient descent uses only the slope, Newton also uses the curvature, and near a minimum it converges *quadratically* (the error roughly squares each step).

**Example 8.6 (Newton's method, iteration by iteration).** Minimize $f(x) = \frac{1}{4}x^4 - x$, with $f'(x) = x^3 - 1$ and $f''(x) = 3x^2$; the minimum is at $x^\star = 1$ (where $f' = 0$). Newton's step is $x_{t+1} = x_t - \dfrac{x_t^3 - 1}{3x_t^2}$. From $x_0 = 2$:

| $t$ | $x_t$ | $f'(x_t) = x_t^3 - 1$ | $f''(x_t) = 3x_t^2$ |
|---|---|---|---|
| 0 | 2.00000 | 7.000 | 12.00 |
| 1 | 1.41667 | 1.843 | 6.021 |
| 2 | 1.11053 | 0.370 | 3.700 |
| 3 | 1.01064 | 0.032 | 3.064 |
| 4 | 1.00011 | 0.0003 | 3.001 |

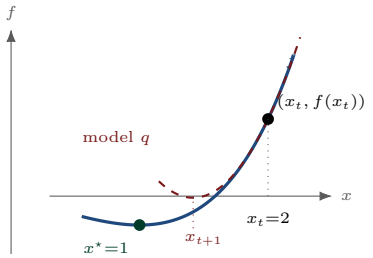



**Figure 8.2.** Newton's method as repeated quadratic fitting. At $x_t = 2$ the dashed parabola $q$ matches $f = \frac{1}{4}x^4 - x$ in value, slope, and curvature; its vertex sits at $x_{t+1} = 1.417$, the next iterate. Refitting at $x_{t+1}$ and stepping to the new vertex races toward the true minimum $x^\star = 1$, squaring the error each step.

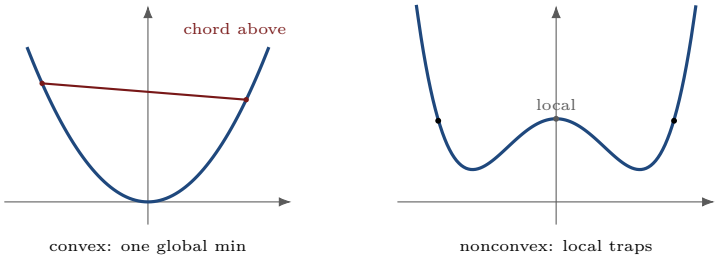



**Figure 8.3.** Convex versus nonconvex. For a convex function every chord lies above the graph and there is a single global minimum, so a zero-gradient point is the answer. A nonconvex function has local minima and saddle points that can trap a gradient method. The losses in this chapter are built convex to avoid exactly this.

The error $|x_t - 1|$ drops $1 \to 0.42 \to 0.11 \to 0.011 \to 0.0001$: each step roughly *squares* the previous error, the hallmark of quadratic convergence. Gradient descent on the same problem would creep linearly. The price is that Newton needs the Hessian (here $f''$) and its inverse every step, which is $\mathcal{O}(n^3)$ per iteration in $n$ dimensions, why large models use first-order methods or cheap Hessian approximations instead.

## 8.6 The Hessian and convexity

**Definition 8.7 (Hessian).** The *Hessian* of $f : \mathbb{R}^d \to \mathbb{R}$ is the symmetric matrix of second partials, $(\nabla^2 f)_{ij} = \frac{\partial^2 f}{\partial x_i \, \partial x_j}$ (symmetric when the mixed partials are continuous, by Clairaut's theorem). It measures local curvature.

Curvature decides whether a stationary point is a minimum, and convexity makes the local picture global.

**Definition 8.8 (Convex set and convex function).** A set $C \subseteq \mathbb{R}^d$ is *convex* if the segment between any two of its points stays inside: $\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in C$ for all $\boldsymbol{x}, \boldsymbol{y} \in C$ and $\lambda \in [0,1]$. A function $f$ is *convex* if its epigraph $\{(\boldsymbol{x}, t) : t \geq f(\boldsymbol{x})\}$ is a convex set, equivalently if

$$f\big(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\big) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}) \tag{8.4}$$

for all $\boldsymbol{x}, \boldsymbol{y}$ and $\lambda \in [0,1]$: the chord lies above the graph.

**Theorem 8.9 (The convexity triangle).** For a differentiable $f$ (twice differentiable for (iii)), the following are equivalent characterizations of convexity:

(i) *First-order:* $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{y}$ (the graph lies above every tangent plane);
(ii) *Global optimality:* if $f$ is convex and $\nabla f(\boldsymbol{x}^\star) = \boldsymbol{0}$, then $\boldsymbol{x}^\star$ is a *global* minimizer;
(iii) *Second-order:* $f$ is convex if and only if $\nabla^2 f(\boldsymbol{x}) \succeq 0$ everywhere.

*Proof.* (ii) follows from (i): putting $\boldsymbol{x} = \boldsymbol{x}^\star$ with $\nabla f(\boldsymbol{x}^\star) = \boldsymbol{0}$ in the first-order inequality gives $f(\boldsymbol{y}) \geq f(\boldsymbol{x}^\star)$ for all $\boldsymbol{y}$, so $\boldsymbol{x}^\star$ is globally optimal. (i) from convexity: rearranging Eq. (8.4) as $\frac{f(\boldsymbol{x}+\lambda(\boldsymbol{y}-\boldsymbol{x})) - f(\boldsymbol{x})}{\lambda} \leq f(\boldsymbol{y}) - f(\boldsymbol{x})$ and letting $\lambda \to 0^+$ turns the left side into the directional derivative $\nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$, giving (i). (iii): integrating the second-order Taylor remainder Eq. (8.3) along the segment from $\boldsymbol{x}$ to $\boldsymbol{y}$ shows $\nabla^2 f \succeq 0$ implies the first-order inequality; conversely a negative curvature direction would violate it. ∎

This is the reason zero gradient is enough for convex problems: there are no local minima to get trapped in, only the global one. Everything in this chapter's applications is built to be convex for exactly this guarantee.

**Example 8.10 (Finding and classifying critical points with the Hessian).** When a function is not convex, a zero gradient alone cannot tell a minimum from a maximum or a saddle; the Hessian's curvature decides. Find and classify every critical point of

$$f(x, y) = x^3 + y^3 - 3xy.$$

**Stationary points.** Set the gradient to zero: $\nabla f = (3x^2 - 3y,\ 3y^2 - 3x) = \boldsymbol{0}$ forces $y = x^2$ and $x = y^2$. Substituting the first into the second gives $x = x^4$, so $x(x^3 - 1) = 0$ and $x \in \{0, 1\}$. The two critical points are $(0,0)$ and $(1,1)$.
**Curvature.** The Hessian is

$$\nabla^2 f(x, y) = \begin{bmatrix} 6x & -3 \\ -3 & 6y \end{bmatrix}.$$

At $(0,0)$ it is $\left(\begin{smallmatrix} 0 & -3 \\ -3 & 0 \end{smallmatrix}\right)$ with $\det = 0 \cdot 0 - (-3)^2 = -9 < 0$. A negative determinant means eigenvalues of opposite sign (here $\pm 3$), so the surface rises in one direction and falls in the perpendicular one: a *saddle*. At $(1,1)$ the Hessian is $\left(\begin{smallmatrix} 6 & -3 \\ -3 & 6 \end{smallmatrix}\right)$ with $\det = 36 - 9 = 27 > 0$ and trace $12 > 0$, so both eigenvalues are positive (3 and 9): a *local minimum*, where $f(1,1) = 1 + 1 - 3 = -1$. In two dimensions the second-derivative test reads straight off the Hessian: $\det < 0$ is a saddle, $\det > 0$ with positive trace is a minimum, and $\det > 0$ with negative trace is a maximum. The saddle is exactly a mountain pass, where you climb along one axis while descending along the one across it, and that crossed curvature is what a negative-determinant Hessian encodes.

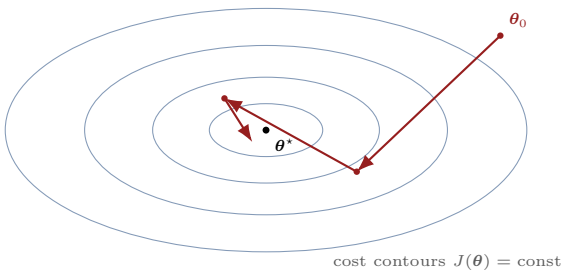



**Figure 8.4.** Gradient descent on a convex quadratic cost. Each step moves opposite the gradient, perpendicular to the contour through the current point, converging to the unique minimum $\boldsymbol{\theta}^\star$. A poorly conditioned cost (elongated contours) forces the zig-zag the optimizers of Section 8.9 are designed to damp.

## 8.7 Linear regression: gradient, Hessian, and a unique minimum

Take the regression cost $J(\boldsymbol{\theta}) = \frac{1}{2m}\|X\boldsymbol{\theta} - \boldsymbol{y}\|_2^2$. Writing the squared norm as a trace and using $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ expands it to

$$J(\boldsymbol{\theta}) = \tfrac{1}{2m}\big(\boldsymbol{\theta}^\top X^\top X\boldsymbol{\theta} - 2\boldsymbol{\theta}^\top X^\top \boldsymbol{y} + \boldsymbol{y}^\top \boldsymbol{y}\big).$$

The identities Eq. (8.1) with the symmetric $A = X^\top X$ then give the gradient and, differentiating again, the Hessian:

$$\nabla J(\boldsymbol{\theta}) = \frac{1}{m}X^\top(X\boldsymbol{\theta} - \boldsymbol{y}), \qquad \nabla^2 J(\boldsymbol{\theta}) = \frac{1}{m}X^\top X. \tag{8.5}$$

The Hessian is positive semidefinite, since $\boldsymbol{z}^\top X^\top X\boldsymbol{z} = \frac{1}{m}\|X\boldsymbol{z}\|^2 \geq 0$ for all $\boldsymbol{z}$, so $J$ is convex by Theorem 8.9(iii). Setting $\nabla J = \mathbf{0}$ recovers the normal equations $X^\top X\boldsymbol{\theta}^\star = X^\top \boldsymbol{y}$ of Chapter 2, and convexity certifies that this stationary point is the *global* minimum rather than merely a local one.

## 8.8 Gradient descent

When the closed-form solution is too expensive (Section 2.11), or absent, we minimize iteratively. *Gradient descent* repeatedly steps along the steepest-descent direction:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha\,\nabla J(\boldsymbol{\theta}_t), \tag{8.6}$$

with *learning rate* $\alpha > 0$. For the quadratic cost this is $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\alpha}{m}X^\top(X\boldsymbol{\theta}_t - \boldsymbol{y})$.

The learning rate must be tuned: too large and the iterates overshoot and diverge; too small and progress crawls. For a strongly convex quadratic with Hessian eigenvalues in $[\mu, L]$, gradient descent converges (linearly) for any fixed step $\alpha \in (0, 2/L)$, where $L = \lambda_{\max}(X^\top X)/m$ is the largest curvature. The ratio $L/\mu$, the condition number, sets the speed: ill-conditioned problems with elongated contours zig-zag slowly (Fig. 8.4).

**Example 8.11 (Gradient descent on a quadratic, iteration by iteration).** Minimize $f(\boldsymbol{\theta}) = \frac{1}{2}(3\theta_1^2 + \theta_2^2)$, a quadratic bowl with Hessian $A = \operatorname{diag}(3, 1)$ (so the curvatures are $L = 3$ and $\mu = 1$, condition number 3). The gradient is $\nabla f = A\boldsymbol{\theta} = (3\theta_1, \theta_2)$, so the step $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha A\boldsymbol{\theta}_t = (I - \alpha A)\boldsymbol{\theta}_t$ multiplies each coordinate by its own factor $1 - \alpha\lambda_i$. Take $\alpha = 0.2$ from $\boldsymbol{\theta}_0 = (1, 1)$, so the factors are $1 - 0.2(3) = 0.4$ and $1 - 0.2(1) = 0.8$:

| $t$ | $\boldsymbol{\theta}_t$ | $f(\boldsymbol{\theta}_t)$ | $\nabla f(\boldsymbol{\theta}_t)$ |
|---|---|---|---|
| 0 | (1, 1) | 2.000 | (3, 1) |
| 1 | (0.4, 0.8) | 0.560 | (1.2, 0.8) |
| 2 | (0.16, 0.64) | 0.243 | (0.48, 0.64) |
| 3 | (0.064, 0.512) | 0.137 | (0.192, 0.512) |
| 4 | (0.0256, 0.4096) | 0.085 | (. . . ) |

The steep coordinate $\theta_1$ collapses quickly (as $0.4^t$) while the shallow $\theta_2$ crawls (as $0.8^t$), so the *slow* direction sets the pace: after four steps $\theta_1$ is nearly zero but $\theta_2$ is still 0.41. This is ill-conditioning in miniature, and it is why momentum and preconditioning (Section 8.9) help. With a larger step $\alpha = 0.7 > 2/L = \frac{2}{3}$ the factor $1 - 0.7(3) = -1.1$ has magnitude $> 1$ and $\theta_1$ would *diverge*, oscillating with growing amplitude, the overshoot the bound $\alpha < 2/L$ rules out.

### 8.8.1 Stochastic gradient descent

The gradient $\nabla J(\boldsymbol{\theta}) = \frac{1}{m}\sum_i \nabla J_i(\boldsymbol{\theta})$ is an average over all $m$ training examples, so one step of full-batch gradient descent reads the entire dataset. With $m$ in the millions this is far too expensive per step. *Stochastic gradient descent* (SGD) replaces the full average by the gradient on a small *mini-batch* $B$:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \frac{1}{|B|}\sum_{i\in B} \nabla J_i(\boldsymbol{\theta}_t). \tag{8.7}$$

The mini-batch gradient is an unbiased estimate of the true gradient ($\mathbb{E}[\nabla J_i] = \nabla J$), so on average SGD heads downhill, but each step is noisy. That trade is overwhelmingly worth it: a mini-batch of 128 costs a tiny fraction of a full pass, so SGD takes thousands of cheap, approximate steps in the time full-batch descent takes one exact step. The noise even helps, jostling the iterate out of shallow nonconvex traps. Decaying the learning rate $\alpha_t \to 0$ tames the noise near the optimum. SGD and the variants below are how every large model is trained.

## 8.9 Practical optimizers: momentum, RMSProp, Adam

Plain gradient descent is rarely used as-is. Three refinements address its weaknesses, all maintaining running statistics of the gradients $g_t = \nabla J(\boldsymbol{\theta}_t)$.

- **Momentum** averages recent gradients into a velocity that damps zig-zags and accelerates along consistent directions:
$$\boldsymbol{v}_{t+1} = \beta \boldsymbol{v}_t + (1-\beta) g_t, \qquad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \boldsymbol{v}_{t+1},$$
with $\beta \approx 0.9$. Like a heavy ball rolling downhill, it carries through small bumps in noisy gradients.
- **RMSProp** divides each coordinate's step by a running root-mean-square of its gradient, so steep coordinates take small steps and flat ones take large:
$$\boldsymbol{s}_{t+1} = \rho \boldsymbol{s}_t + (1-\rho)(g_t \odot g_t), \qquad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\alpha}{\sqrt{\boldsymbol{s}_{t+1}} + \varepsilon} \odot g_t,$$
with $\rho \approx 0.9$ and a small $\varepsilon$ for numerical safety ($\odot$ is the elementwise product).
- **Adam** combines the two, tracking a first moment $\boldsymbol{m}_t$ (like momentum) and a second moment $\boldsymbol{v}_t$ (like RMSProp), with a bias correction for the early steps:
$$\boldsymbol{m}_{t+1} = \beta_1 \boldsymbol{m}_t + (1-\beta_1) g_t, \quad \boldsymbol{v}_{t+1} = \beta_2 \boldsymbol{v}_t + (1-\beta_2)(g_t \odot g_t),$$
$$\hat{\boldsymbol{m}} = \frac{\boldsymbol{m}_{t+1}}{1-\beta_1^{t+1}}, \quad \hat{\boldsymbol{v}} = \frac{\boldsymbol{v}_{t+1}}{1-\beta_2^{t+1}}, \qquad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha \frac{\hat{\boldsymbol{m}}}{\sqrt{\hat{\boldsymbol{v}}} + \varepsilon},$$
with typical $\beta_1 = 0.9$, $\beta_2 = 0.999$. Adam [15] is a default for training large models: it adapts the step size per coordinate and tolerates a poorly tuned learning rate.

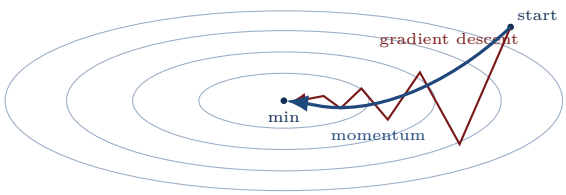



**Figure 8.5.** Momentum versus plain gradient descent on an ill-conditioned bowl. Plain descent zig-zags across the narrow valley, taking many small steps; momentum accumulates a velocity along the valley floor, damping the oscillation and reaching the minimum in far fewer strides. RMSProp and Adam add a per-coordinate rescaling on top of the same idea.

| Optimizer | Idea | Fixes |
|---|---|---|
| Gradient descent | step along $-g_t$ | baseline |
| Momentum | accumulate a velocity | zig-zag, slow valleys |
| RMSProp | per-coordinate step $\propto 1/\sqrt{\overline{g^2}}$ | badly scaled coordinates |
| Adam | momentum + RMSProp + bias fix | both, with little tuning |

**Table 8.1.** The optimizer family. Each refinement keeps running statistics of the gradients to repair a specific weakness of plain descent; Adam combines them and is the common default. All are first-order: they use only $g_t$, never the Hessian.

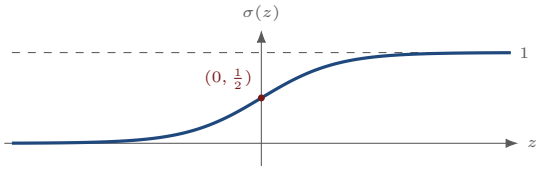



**Figure 8.6.** The logistic sigmoid $\sigma(z) = 1/(1+e^{-z})$ turns a real-valued score into a probability. It is monotone, symmetric about $(0, \frac{1}{2})$, and saturates toward 0 and 1; its derivative $\sigma'(z) = \sigma(z)(1-\sigma(z))$ is what the backprop error signal of Example 8.3 multiplies by.

## 8.10 Logistic regression: a convex loss

For binary labels $y_i \in \{0,1\}$, logistic regression passes the linear score $z_i = \boldsymbol{\theta}^\top \boldsymbol{x}_i$ through the *sigmoid* $\sigma(z) = 1/(1+e^{-z})$ to get a probability $p_i = \sigma(z_i)$. The sigmoid (Fig. 8.6) squashes the unbounded score into $(0,1)$: large positive scores map near 1, large negative near 0, and a score of zero to the undecided $\frac{1}{2}$. Fitting by maximum likelihood minimizes the *cross-entropy* loss

$$\ell(\boldsymbol{\theta}) = -\frac{1}{m}\sum_{i=1}^{m}\big[y_i \log p_i + (1-y_i)\log(1-p_i)\big], \tag{8.8}$$

whose gradient has the same clean form as linear regression, $\nabla\ell(\boldsymbol{\theta}) = \frac{1}{m}X^\top(\boldsymbol{p}-\boldsymbol{y})$ with $\boldsymbol{p} = \sigma(X\boldsymbol{\theta})$. The loss is convex (its Hessian $\frac{1}{m}X^\top \operatorname{diag}(p_i(1-p_i))X$ is PSD), so gradient descent finds the global optimum. The choice of loss matters: the squared loss $\frac{1}{2m}\sum_i (p_i - y_i)^2$ on the same model is *nonconvex* because $p_i$ is a nonlinear function of $\boldsymbol{\theta}$, which is why cross-entropy is the standard choice. Logistic regression returns in Chapter 10 as the probabilistic cousin of the support vector machine.

**Example 8.12 (Two steps of logistic-regression training).** Fit $\boldsymbol{\theta} = (\theta_1, \theta_2)$ (intercept and one feature) to three points with design matrix $X = \left[\begin{smallmatrix}1&0\\1&1\\1&2\end{smallmatrix}\right]$ and labels $\boldsymbol{y} = (0,0,1)$, starting at $\boldsymbol{\theta}_0 = \mathbf{0}$ with learning rate $\alpha = 1$.
**Step 0.** Every score is 0, so $\boldsymbol{p} = \sigma(\mathbf{0}) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ and the gradient is

$$\nabla\ell = \tfrac{1}{3}X^\top(\boldsymbol{p}-\boldsymbol{y}) = \tfrac{1}{3}\begin{bmatrix}1&1&1\\0&1&2\end{bmatrix}\begin{bmatrix}\frac{1}{2}\\ \frac{1}{2}\\ -\frac{1}{2}\end{bmatrix} = \tfrac{1}{3}\begin{bmatrix}\frac{1}{2}\\ -\frac{1}{2}\end{bmatrix} = (0.167,\ -0.167).$$

Updating, $\boldsymbol{\theta}_1 = \mathbf{0} - (0.167, -0.167) = (-0.167,\ 0.167)$. **Step 1.** Now the scores are $X\boldsymbol{\theta}_1 = (-0.167,\ 0,\ 0.167)$, giving $\boldsymbol{p} = (0.458,\ 0.500,\ 0.542)$ and $\nabla\ell = \frac{1}{3}X^\top(\boldsymbol{p}-\boldsymbol{y}) = (0.167,\ -0.139)$, so $\boldsymbol{\theta}_2 = (-0.333,\ 0.306)$. The intercept $\theta_1$ drifts negative and the slope $\theta_2$ positive, steadily raising the predicted probability on the lone positive point $x = 2$ ($p_3 : 0.5 \to 0.542 \to 0.569$) and lowering it on the negatives. Each step is the same residual-times-design-matrix formula as linear regression; only the nonlinear $\boldsymbol{p} = \sigma(X\boldsymbol{\theta})$ differs.

**Example 8.13 (The softmax gradient: why it is just $p - y$).** Multiclass classification replaces the sigmoid with the *softmax*, which turns a vector of scores $\boldsymbol{z} = (z_1, \dots, z_K)$ into a probability vector $p_k = \dfrac{e^{z_k}}{\sum_j e^{z_j}}$, and the loss is the cross-entropy $L = -\sum_k y_k \log p_k$ against a one-hot label $\boldsymbol{y}$. The gradient with respect to the scores collapses to

a remarkably clean form. First differentiate the softmax itself, which has two cases that combine into one using the Kronecker delta $\delta_{ki}$:

$$\frac{\partial p_k}{\partial z_i} = p_k(\delta_{ki} - p_i).$$

Now apply the chain rule to $L = -\sum_k y_k \log p_k$, using $\partial(\log p_k)/\partial z_i = (\delta_{ki} - p_i)$:

$$\frac{\partial L}{\partial z_i} = -\sum_k y_k(\delta_{ki} - p_i) = -y_i + p_i \sum_k y_k = p_i - y_i,$$

since a one-hot label sums to $\sum_k y_k = 1$. In vector form $\nabla_{\boldsymbol{z}} L = \boldsymbol{p} - \boldsymbol{y}$: the gradient is simply the predicted distribution minus the true one. As a check, scores $\boldsymbol{z} = (2, 1, 0)$ give $\boldsymbol{p} = (0.665, 0.245, 0.090)$, so with true class one the gradient is $(0.665 - 1,\ 0.245,\ 0.090) = (-0.335,\ 0.245,\ 0.090)$, which pushes $z_1$ up and the other two down, exactly the correction needed. This is the same "prediction minus target" shape as the linear-regression residual and the logistic gradient $X^\top(\boldsymbol{p} - \boldsymbol{y})$; all three are the same idea, and it is what makes the output layer of a neural network so simple to backpropagate through.

## 8.11 Summary

- The gradient ($\nabla f \in \mathbb{R}^d$) and Jacobian ($\boldsymbol{J} \in \mathbb{R}^{m \times n}$) package first derivatives; the identities Eq. (8.1)–Eq. (8.2) cover most ML gradients.
- The chain rule, run in reverse, is backpropagation: each layer computes an error signal $\boldsymbol{\delta}$, a weight gradient $\boldsymbol{\delta}\boldsymbol{x}^\top$, and passes $W^\top\boldsymbol{\delta}$ back, giving all gradients in one backward sweep.
- The negative gradient is the steepest-descent direction. The Hessian measures curvature; convexity (chord above graph, tangent below graph, PSD Hessian) makes a zero-gradient point a global minimum.
- Linear regression has $\nabla J = \frac{1}{m}X^\top(X\boldsymbol{\theta} - \boldsymbol{y})$ and PSD Hessian $\frac{1}{m}X^\top X$, hence a unique global minimum at the normal equations.
- Gradient descent $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha\nabla J$ converges for $\alpha \in (0, 2/L)$; stochastic gradient descent uses cheap mini-batch gradients to scale to huge datasets, and momentum, RMSProp, and Adam refine it (Table 8.1). Logistic regression's cross-entropy loss is convex with gradient $\frac{1}{m}X^\top(\boldsymbol{p} - \boldsymbol{y})$.

## Exercises

**Exercise 8.1.** Using the matrix-calculus identities, derive the gradient and Hessian of the ridge-regression cost $J(\boldsymbol{\theta}) = \frac{1}{2}\|X\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2$, and explain why the Hessian is positive definite for $\lambda > 0$ even when $X$ is rank deficient.

*Solution.* Write $J = \frac{1}{2}(X\boldsymbol{\theta} - \boldsymbol{y})^\top(X\boldsymbol{\theta} - \boldsymbol{y}) + \frac{\lambda}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta}$. By Eq. (8.2) and $\nabla(\frac{\lambda}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta}) = \lambda\boldsymbol{\theta}$,

$$\nabla J(\boldsymbol{\theta}) = X^\top(X\boldsymbol{\theta} - \boldsymbol{y}) + \lambda\boldsymbol{\theta} = (X^\top X + \lambda I)\boldsymbol{\theta} - X^\top\boldsymbol{y}, \qquad \nabla^2 J = X^\top X + \lambda I.$$

For any $\boldsymbol{z} \neq \boldsymbol{0}$, $\boldsymbol{z}^\top(X^\top X + \lambda I)\boldsymbol{z} = \|X\boldsymbol{z}\|^2 + \lambda\|\boldsymbol{z}\|^2 \geq \lambda\|\boldsymbol{z}\|^2 > 0$, so $\nabla^2 J \succ 0$ (positive *definite*) whenever $\lambda > 0$. The ridge term lifts every eigenvalue of $X^\top X$ by $\lambda$, curing the rank deficiency that made $X^\top X$ singular in Example 2.12; the minimizer $\boldsymbol{\theta}^\star = (X^\top X + \lambda I)^{-1}X^\top\boldsymbol{y}$ is then unique. □

**Exercise 8.2.** For logistic regression with data $X = \left[\begin{smallmatrix} 1 & 2 \\ 1 & -1 \\ 1 & 0 \end{smallmatrix}\right]$, labels $\boldsymbol{y} = (1, 0, 1)$, and $\boldsymbol{\theta}_0 = (0, 0)$, take one full-batch gradient-ascent step on the log-likelihood with learning rate $\alpha = 0.01$.

*Solution.* At $\boldsymbol{\theta}_0 = \boldsymbol{0}$ every score is $z_i = 0$, so $p_i = \sigma(0) = \frac{1}{2}$ for all $i$, giving $\boldsymbol{p} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. The log-likelihood gradient (for ascent) is $X^\top(\boldsymbol{y} - \boldsymbol{p})$:

$$\boldsymbol{y} - \boldsymbol{p} = \left(\tfrac{1}{2}, -\tfrac{1}{2}, \tfrac{1}{2}\right), \qquad X^\top(\boldsymbol{y} - \boldsymbol{p}) = \begin{bmatrix} 1 & 1 & 1 \\ 2 & -1 & 0 \end{bmatrix}\begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{3}{2} \end{bmatrix}.$$

One ascent step is $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_0 + \alpha\, X^\top(\boldsymbol{y} - \boldsymbol{p}) = (0, 0) + 0.01\,(0.5, 1.5) = (0.005,\ 0.015)$. The positive update to both coordinates nudges the model toward predicting the observed labels. □

**Exercise 8.3.** Prove that if $f : \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable and $\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$, then $\boldsymbol{x}^\star$ is a global minimizer.

*Solution.* By the first-order characterization of convexity (Theorem 8.9(i)), for every $\boldsymbol{y}$,

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}^\star) + \nabla f(\boldsymbol{x}^\star)^\top (\boldsymbol{y} - \boldsymbol{x}^\star).$$

Since $\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$, the inner-product term vanishes and $f(\boldsymbol{y}) \geq f(\boldsymbol{x}^\star)$ for all $\boldsymbol{y}$. Hence $\boldsymbol{x}^\star$ attains the global minimum. This is exactly why, for a convex loss, solving $\nabla f = \mathbf{0}$ (or running gradient descent to a stationary point) is guaranteed to find the best parameters, with no risk of a local trap. □

**Exercise 8.4.** Minimize $f(\theta) = \frac{1}{2}\theta^2$ by gradient descent from $\theta_0 = 4$ with learning rate $\alpha$. Write $\theta_t$ in closed form, and determine the range of $\alpha$ for which the iterates converge to 0.

*Solution.* Here $f'(\theta) = \theta$, so the update is $\theta_{t+1} = \theta_t - \alpha\theta_t = (1 - \alpha)\theta_t$. Iterating from $\theta_0 = 4$ gives $\theta_t = (1 - \alpha)^t \cdot 4$. This tends to 0 if and only if $|1 - \alpha| < 1$, i.e. $0 < \alpha < 2$. (The curvature is $L = 1$ here, matching the general bound $\alpha \in (0, 2/L)$.) For $\alpha = 1$ it reaches the minimum in one step; for $\alpha > 2$ the iterates grow and diverge; at $\alpha = 2$ they oscillate between 4 and $-4$ forever. □

**Exercise 8.5.** In Example 8.3, let $\boldsymbol{x} \in \mathbb{R}^n$, $W \in \mathbb{R}^{h \times n}$, so $\boldsymbol{z}, \boldsymbol{a} \in \mathbb{R}^h$ and $L \in \mathbb{R}$. Verify the shapes of $\boldsymbol{\delta}$, $\nabla_W L$, and $\partial L/\partial \boldsymbol{x}$, and confirm they are consistent with updating $W$ and propagating the error.

*Solution.* The error signal $\boldsymbol{\delta} = \partial L/\partial \boldsymbol{z}$ has the shape of $\boldsymbol{z}$, namely $h \times 1$. The weight gradient $\nabla_W L = \boldsymbol{\delta}\boldsymbol{x}^\top$ is $(h \times 1)(1 \times n) = h \times n$, the same shape as $W$, so the update $W \leftarrow W - \alpha \nabla_W L$ typechecks. The backward signal $\partial L/\partial \boldsymbol{x} = W^\top \boldsymbol{\delta}$ is $(n \times h)(h \times 1) = n \times 1$, the shape of $\boldsymbol{x}$, so it can serve as the error handed to the previous layer. The shapes line up exactly, which is the quickest sanity check on any backprop derivation. □

**Exercise 8.6.** For $f(x, y, z) = x^2 y + y z^2$, (a) find the gradient $\nabla f$, (b) evaluate it at the point $(1, 2, 3)$, and (c) give the directional derivative there in the direction of the unit vector $\boldsymbol{u} = (1, 0, 0)$.

*Solution.* (a) Differentiate with respect to each variable, holding the others fixed:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right) = (2xy,\ x^2 + z^2,\ 2yz).$$

(b) At $(1, 2, 3)$ this is $(2 \cdot 1 \cdot 2,\ 1^2 + 3^2,\ 2 \cdot 2 \cdot 3) = (4,\ 10,\ 12)$. (c) The directional derivative in a unit direction $\boldsymbol{u}$ is the projection of the gradient onto it, $D_{\boldsymbol{u}} f = \nabla f^\top \boldsymbol{u}$. With $\boldsymbol{u} = (1, 0, 0)$ this simply reads off the first component, $D_{\boldsymbol{u}} f = (4, 10, 12)^\top (1, 0, 0) = 4$, which is just $\partial f/\partial x$ as expected. The gradient $(4, 10, 12)$ also points in the direction of steepest ascent, and the function climbs fastest at the rate $\|\nabla f\| = \sqrt{16 + 100 + 144} = \sqrt{260} \approx 16.1$ per unit step in that best direction, four times steeper than along the $x$-axis alone. □

**Exercise 8.7.** Fit the through-origin model $y = \theta x$ to the points $(1, 1)$ and $(2, 3)$ by gradient descent on $J(\theta) = \sum_i (\theta x_i - y_i)^2$. (a) Find the closed-form least-squares $\theta^\star$. (b) Write the gradient-descent update with step $\alpha = 0.05$ and trace a few iterations from $\theta_0 = 0$, confirming it approaches $\theta^\star$.

*Solution.* (a) Setting $J'(\theta) = 0$ gives the one-variable normal equation. With $\sum x_i^2 = 1 + 4 = 5$ and $\sum x_i y_i = 1(1) + 2(3) = 7$,

$$J'(\theta) = 2\Big(\theta \sum x_i^2 - \sum x_i y_i\Big) = 2(5\theta - 7) = 0 \implies \theta^\star = \frac{7}{5} = 1.4.$$

(b) The update is $\theta_{t+1} = \theta_t - \alpha J'(\theta_t) = \theta_t - 0.05\,(10\theta_t - 14) = 0.5\,\theta_t + 0.7$. From $\theta_0 = 0$ this gives $0.7,\ 1.05,\ 1.225,\ 1.3125,\ 1.35625, \ldots$, halving the gap to 1.4 each step (the factor $0.5 = 1 - \alpha \cdot J''$ with $J'' = 10$). The iteration is contracting because $|1 - \alpha J''| = |1 - 0.5| < 1$, and its fixed point $\theta = 0.5\theta + 0.7$ solves to $\theta = 1.4$, exactly the closed-form answer. Gradient descent and the normal equations reach the same least-squares solution; the closed form is one matrix solve, while descent crawls there iteratively but scales to problems too large to solve in closed form. □

CHAPTER 9

# Constrained Optimization: Linear Programming, Duality, and KKT

*The dual problem assigns a price to every constraint, and at the optimum the buyer and the seller agree.*

*the bargaining picture of duality*

**Roadmap**

Chapter 8 minimized smooth functions with no constraints. Real problems have them: probabilities sum to one, margins must be respected, budgets cannot be exceeded. This chapter handles constraints. We begin with linear programming and the simplex method, then build the general theory: the *Lagrangian* folds constraints into the objective with multipliers, the *dual problem* produces lower bounds, and *weak* and *strong* duality relate the two. The *Karush–Kuhn–Tucker conditions* characterize a constrained optimum in four lines, and they are the exact tool that derives the support vector machine in Chapter 10.

**Suggested reading:** Boyd and Vandenberghe [3], Chapters 4 and 5; Nocedal and Wright [16] for the algorithmic side.

## 9.1 The constrained problem

The general constrained optimization problem minimizes an objective subject to inequality and equality constraints:

$$\min_{\boldsymbol{w}\in\mathbb{R}^n} f(\boldsymbol{w}) \quad \text{subject to} \quad g_i(\boldsymbol{w}) \le 0 \ (i=1,\dots,K), \quad h_j(\boldsymbol{w}) = 0 \ (j=1,\dots,L). \tag{9.1}$$

The set of $\boldsymbol{w}$ satisfying every constraint is the *feasible set*. A constraint is *active* at a point if it holds with equality there, and *inactive* if there is slack. The problem is *convex* when $f$ and the $g_i$ are convex and the $h_j$ are affine, the case where the theory is cleanest and the case that covers least squares, logistic regression, and the SVM.

## 9.2 Linear programming

When the objective and all constraints are linear, Eq. (9.1) is a *linear program* (LP). A standard form is

$$\max_{\boldsymbol{x}\ge\mathbf{0}} \ \boldsymbol{c}^{\top}\boldsymbol{x} \quad \text{subject to} \quad A\boldsymbol{x} \le \boldsymbol{b}, \tag{9.2}$$

which any LP can be rewritten to match: negate $\boldsymbol{c}$ to flip a minimization, negate a row to reverse a $\ge$ constraint, and split an equality into two inequalities. Introducing *slack variables* $s_i \ge 0$ then turns each inequality $\boldsymbol{a}_i^{\top}\boldsymbol{x} \le b_i$ into an equality $\boldsymbol{a}_i^{\top}\boldsymbol{x} + s_i = b_i$, the equality form the simplex method pivots on. The feasible set is a *polytope*, the intersection of half-spaces, and because the objective is linear it is optimized at a *vertex* (corner) of that polytope.

**Example 9.1 (A two-variable LP).** Maximize $x_1 + 2x_2$ subject to $x_1 + x_2 \le 4$, $x_1 + 3x_2 \le 6$, $x_1, x_2 \ge 0$. The feasible region is the quadrilateral with vertices $(0,0), (4,0), (3,1), (0,2)$ (Fig. 9.1), where the last is the intersection of the two constraint lines ($x_1 + x_2 = 4$ and $x_1 + 3x_2 = 6$ give $x_2 = 1, x_1 = 3$). Evaluating the objective at the vertices gives $0, 4, 5, 4$, so the maximum is 5 at $(3,1)$. The optimum sits at a corner, as the theory promises, and there both constraints are active.

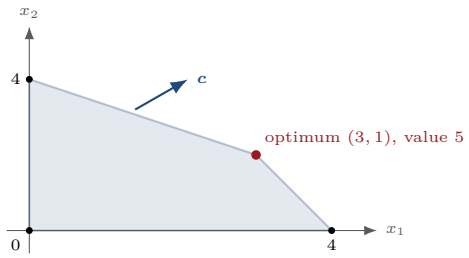



**Figure 9.1.** A linear program in two variables. The feasible polytope is shaded; the number at each vertex is the objective $x_1 + 2x_2$. The objective increases in the direction $\boldsymbol{c} = (1, 2)$, so the maximum is attained at the vertex $(3, 1)$ where both constraints are active. The simplex method walks vertex-to-vertex toward it.

## 9.3 The simplex method

Checking every vertex is hopeless once there are many variables, since the number of vertices grows combinatorially. The *simplex method* instead starts at one vertex and repeatedly moves to an adjacent vertex that improves the objective, stopping when no neighbor is better, which for a convex polytope means the global optimum has been reached. Mechanically, one introduces slack variables, writes the system as a tableau, and *pivots*: a nonbasic variable with a favorable objective coefficient enters the basis while a basic variable leaves (chosen by a ratio test that keeps all variables nonnegative). In Example 9.1 the method would start at the origin $(0, 0)$, pivot to $(4, 0)$ (raising the objective from 0 to 4), and pivot again to $(3, 1)$ (objective 5), at which point no adjacent vertex improves and the algorithm halts. Each pivot is one step of Gaussian elimination on the tableau; the art is only in choosing which variable enters and leaves.

**Example 9.2 (A full simplex run, tableau by tableau).** Maximize $z = 10 + 20x_1 + 16x_2 + 5x_3$ subject to $x_1 \le 4$, $2x_1 + x_2 + x_3 \le 10$, $2x_1 + 2x_2 + x_3 \le 16$, and $x_1, x_2, x_3 \ge 0$. Add slacks $x_4, x_5, x_6 \ge 0$ to turn the inequalities into equalities $x_1 + x_4 = 4$, $2x_1 + x_2 + x_3 + x_5 = 10$, $2x_1 + 2x_2 + x_3 + x_6 = 16$. Starting with the slacks basic ($x_1 = x_2 = x_3 = 0$) gives the feasible corner $x_4 = 4, x_5 = 10, x_6 = 16$ with $z = 10$; the bottom row holds the reduced costs $c_j - z_j$, initially the objective coefficients:

| basis | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | RHS |
|---|---|---|---|---|---|---|---|
| $x_4$ | 1 | 0 | 0 | 1 | 0 | 0 | 4 |
| $x_5$ | 2 | 1 | 1 | 0 | 1 | 0 | 10 |
| $x_6$ | 2 | 2 | 1 | 0 | 0 | 1 | 16 |
| $c_j - z_j$ | 20 | 16 | 5 | 0 | 0 | 0 | $z$=10 |

**Pivot 1.** The largest reduced cost is 20 ($x_1$ enters). The ratio test $\frac{4}{1}, \frac{10}{2}, \frac{16}{2} = 4, 5, 8$ is smallest in the $x_4$ row, so $x_4$ leaves. Pivoting on that 1:

| basis | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | RHS |
|---|---|---|---|---|---|---|---|
| $x_1$ | 1 | 0 | 0 | 1 | 0 | 0 | 4 |
| $x_5$ | 0 | 1 | 1 | −2 | 1 | 0 | 2 |
| $x_6$ | 0 | 2 | 1 | −2 | 0 | 1 | 8 |
| $c_j - z_j$ | 0 | 16 | 5 | −20 | 0 | 0 | $z$=90 |

because $x_1 = 4$ raises $z$ by $20 \cdot 4$ to 90. **Pivot 2.** Now $x_2$ enters (reduced cost 16); the ratio test $\frac{2}{1}, \frac{8}{2} = 2, 4$ picks the $x_5$ row, so $x_5$ leaves:

| basis | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | RHS |
|---|---|---|---|---|---|---|---|
| $x_1$ | 1 | 0 | 0 | 1 | 0 | 0 | 4 |
| $x_2$ | 0 | 1 | 1 | −2 | 1 | 0 | 2 |
| $x_6$ | 0 | 0 | −1 | 2 | −2 | 1 | 4 |
| $c_j - z_j$ | 0 | 0 | −11 | 12 | −16 | 0 | $z$=122 |

**Pivot 3.** A positive reduced cost 12 remains (the slack $x_4$ can still help once feasibility is restored). $x_4$ enters; the ratio test $\frac{4}{1}, \frac{4}{2} = 4, 2$ picks the $x_6$ row, so $x_6$ leaves:

| basis | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | RHS |
|---|---|---|---|---|---|---|---|
| $x_1$ | 1 | 0 | $\frac{1}{2}$ | 0 | 1 | $-\frac{1}{2}$ | 2 |
| $x_2$ | 0 | 1 | 0 | 0 | $-1$ | 1 | 6 |
| $x_4$ | 0 | 0 | $-\frac{1}{2}$ | 1 | $-1$ | $\frac{1}{2}$ | 2 |
| $c_j - z_j$ | 0 | 0 | $-5$ | 0 | $-4$ | $-6$ | $z$=146 |

Every reduced cost is now $\le 0$, so no neighbor improves and we stop. The optimum is $x_1 = 2,\ x_2 = 6,\ x_3 = 0$ with $z = 10 + 20(2) + 16(6) = 146$, which a direct LP solver confirms. Each pivot was one Gauss–Jordan elimination, and the objective climbed $10 \to 90 \to 122 \to 146$ as the search walked corner to corner.

Every LP comes paired with a *dual* LP, an instance of the general duality below. The primal $\max\{\boldsymbol{c}^\top \boldsymbol{x} : A\boldsymbol{x} \le \boldsymbol{b},\ \boldsymbol{x} \ge \boldsymbol{0}\}$ has the dual $\min\{\boldsymbol{b}^\top \boldsymbol{y} : A^\top \boldsymbol{y} \ge \boldsymbol{c},\ \boldsymbol{y} \ge \boldsymbol{0}\}$, with one dual variable $y_i$ per primal constraint. Weak duality reads $\boldsymbol{c}^\top \boldsymbol{x} \le \boldsymbol{b}^\top \boldsymbol{y}$ for any feasible pair, and for LPs strong duality holds whenever either problem is feasible, so the two optima coincide exactly. The dual variable $y_i$ is the shadow price of primal constraint $i$ (Remark 9.9).

**Example 9.3 (The dual of the two-variable LP).** Take the primal of Example 9.1: max $x_1 + 2x_2$ subject to $x_1 + x_2 \le 4$, $x_1 + 3x_2 \le 6$, $\boldsymbol{x} \ge \boldsymbol{0}$, whose optimum was $(3, 1)$ with value 5. Here $\boldsymbol{c} = (1, 2)$, $\boldsymbol{b} = (4, 6)$, $A = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 3 \end{smallmatrix}\right]$, so the dual is

$$\min\ 4y_1 + 6y_2 \quad \text{s.t.} \quad y_1 + y_2 \ge 1, \quad y_1 + 3y_2 \ge 2, \quad y_1, y_2 \ge 0.$$

Both primal constraints were active at $(3, 1)$, so both dual variables are positive, which by complementary slackness makes both dual constraints tight: $y_1 + y_2 = 1$ and $y_1 + 3y_2 = 2$. Subtracting gives $2y_2 = 1$, so $y_2 = \frac{1}{2}$ and $y_1 = \frac{1}{2}$. The dual objective is $4(\frac{1}{2}) + 6(\frac{1}{2}) = 2 + 3 = 5$, equal to the primal optimum, confirming strong duality. The shadow prices $y_1 = y_2 = \frac{1}{2}$ say that relaxing either resource bound by one unit would raise the primal objective by about $\frac{1}{2}$.

## 9.4 The Lagrangian

For nonlinear constraints we need a more general device. The *Lagrangian* attaches a multiplier to each constraint and adds it to the objective:

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\boldsymbol{w}) + \sum_{i=1}^{K} \alpha_i\, g_i(\boldsymbol{w}) + \sum_{j=1}^{L} \beta_j\, h_j(\boldsymbol{w}), \tag{9.3}$$

with *inequality multipliers* $\alpha_i \ge 0$ and *equality multipliers* $\beta_j \in \mathbb{R}$. The multiplier is a price: it penalizes violating its constraint. The sign restriction $\alpha_i \ge 0$ reflects that an inequality $g_i \le 0$ may only be pushed back on from one side. Minimizing $\mathcal{L}$ over $\boldsymbol{w}$ (with the multipliers fixed) trades the objective against the constraints and is the gateway to duality.

*Intuition.* The geometry is cleanest with a single equality constraint $h(\boldsymbol{w}) = 0$. At a constrained optimum the objective cannot decrease along the constraint surface, so its gradient $\nabla f$ can have no component *along* the surface: it must point straight across it, parallel to the constraint's normal $\nabla h$. That is the multiplier equation $\nabla f = -\beta\, \nabla h$, i.e. stationarity of the Lagrangian (Fig. 9.2). The constraint contour and an objective contour are tangent at the optimum; the multiplier $\beta$ is the ratio of the gradient lengths.

**Example 9.4 (Lagrange multipliers: the nearest point on a line).** Find the point of the line $2x + y = 10$ closest to the origin, i.e. minimize $f(x, y) = x^2 + y^2$ subject to $h(x, y) = 2x + y - 10 = 0$. Form the Lagrangian

$$\mathcal{L}(x, y, \lambda) = x^2 + y^2 + \lambda(2x + y - 10).$$

Setting the partials to zero,

$$\frac{\partial \mathcal{L}}{\partial x} = 2x + 2\lambda = 0 \ \Rightarrow\ x = -\lambda, \qquad \frac{\partial \mathcal{L}}{\partial y} = 2y + \lambda = 0 \ \Rightarrow\ y = -\tfrac{\lambda}{2}.$$

Substituting into the constraint $2x + y = 10$ gives $2(-\lambda) + (-\frac{\lambda}{2}) = -\frac{5\lambda}{2} = 10$, so $\lambda = -4$, and therefore $x = 4$, $y = 2$. The minimum value is $f(4, 2) = 16 + 4 = 20$, and indeed the distance from the origin to the line $2x + y = 10$

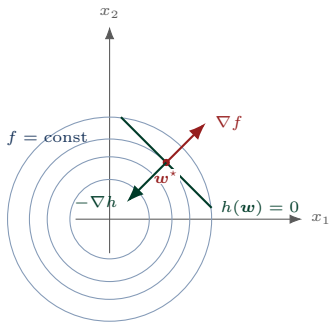



**Figure 9.2.** Lagrange multipliers, the tangency picture. Minimizing $f = x_1^2 + x_2^2$ on the line $x_1 + x_2 = 1$: the optimum $\boldsymbol{w}^\star = (\frac{1}{2}, \frac{1}{2})$ is where an objective contour just touches the constraint line. There the gradients are parallel, $\nabla f = -\beta\,\nabla h$, which is the stationarity condition $\nabla_{\boldsymbol{w}} \mathcal{L} = \mathbf{0}$.

is $\frac{|10|}{\sqrt{2^2+1^2}} = \frac{10}{\sqrt{5}}$, whose square is 20, matching. Geometrically, at the optimum the objective's gradient $\nabla f = (2x, 2y) = (8, 4)$ is parallel to the constraint normal $\nabla h = (2, 1)$, since $(8, 4) = 4(2, 1)$: the level circle of $f$ just kisses the line. Think of the shortest leash from a post at the origin to a straight fence; the taut leash meets the fence at a right angle, and the multiplier $\lambda$ measures how hard it pulls.

## 9.5 The dual function and the dual problem

**Definition 9.5 (Dual function and dual problem).** The *dual function* is the infimum of the Lagrangian over the primal variables,

$$\theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \inf_{\boldsymbol{w} \in \mathbb{R}^n} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}). \tag{9.4}$$

The *dual problem* maximizes it over the admissible multipliers, $\max_{\boldsymbol{\alpha} \geq \mathbf{0},\ \boldsymbol{\beta}} \theta(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

**Example 9.6 (Computing a dual function).** Minimize $x_1^2 + x_2^2$ subject to $x_1 + x_2 = 1$. The Lagrangian is $\mathcal{L} = x_1^2 + x_2^2 + \beta(x_1 + x_2 - 1)$. Setting $\partial \mathcal{L}/\partial x_i = 2x_i + \beta = 0$ gives $x_1 = x_2 = -\beta/2$, and substituting back,

$$\theta(\beta) = 2 \cdot \tfrac{\beta^2}{4} + \beta\big(-\beta - 1\big) = -\tfrac{\beta^2}{2} - \beta.$$

Maximizing the dual, $\theta'(\beta) = -\beta - 1 = 0$ gives $\beta^\star = -1$ and $\theta(-1) = \frac{1}{2}$. The primal optimum is $x_1 = x_2 = \frac{1}{2}$ with value $\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$, so the dual maximum equals the primal minimum here.

## 9.6 Weak and strong duality

**Theorem 9.7 (Weak duality).** For any $\boldsymbol{\alpha} \geq \mathbf{0}$ and any $\boldsymbol{\beta}$, the dual function is a lower bound on the primal optimum: $\theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) \leq f(\boldsymbol{w}^\star)$, where $\boldsymbol{w}^\star$ is an optimal feasible point of Eq. (9.1).

*Proof.* At any feasible $\boldsymbol{w}^\star$ we have $g_i(\boldsymbol{w}^\star) \leq 0$ and $h_j(\boldsymbol{w}^\star) = 0$, so with $\alpha_i \geq 0$ every term $\alpha_i g_i(\boldsymbol{w}^\star) \leq 0$ and every $\beta_j h_j(\boldsymbol{w}^\star) = 0$. Hence $\mathcal{L}(\boldsymbol{w}^\star, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\boldsymbol{w}^\star) + \sum_i \alpha_i g_i(\boldsymbol{w}^\star) + \sum_j \beta_j h_j(\boldsymbol{w}^\star) \leq f(\boldsymbol{w}^\star)$. Taking the infimum over $\boldsymbol{w}$ on the left only decreases it, $\theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \leq \mathcal{L}(\boldsymbol{w}^\star, \boldsymbol{\alpha}, \boldsymbol{\beta}) \leq f(\boldsymbol{w}^\star)$. ∎

The gap $f(\boldsymbol{w}^\star) - \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \theta$ is the *duality gap*. It is always nonnegative, and for nice problems it is zero.

**Theorem 9.8 (Strong duality).** If the problem is convex ($f$ and $g_i$ convex, $h_j$ affine) and *Slater's condition* holds, namely there is a strictly feasible point with $g_i(\tilde{\boldsymbol{w}}) < 0$ for all $i$, then the duality gap is zero:

$$\max_{\boldsymbol{\alpha} \geq \mathbf{0},\ \boldsymbol{\beta}} \theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \min_{\boldsymbol{w}} f(\boldsymbol{w}).$$

When strong duality holds we may solve whichever of the two problems is easier, and the dual is often simpler: for the SVM it has fewer variables and exposes the support vectors (Chapter 10).

| Condition | Equation | Meaning |
|---|---|---|
| Stationarity | $\nabla_{\boldsymbol{w}} \mathcal{L} = \mathbf{0}$ | objective gradient balanced by constraint prices |
| Primal feasibility | $g_i \le 0,\ h_j = 0$ | the point obeys the constraints |
| Dual feasibility | $\alpha_i \ge 0$ | inequality prices are nonnegative |
| Compl. slackness | $\alpha_i\, g_i = 0$ | only active constraints carry a price |

**Table 9.1.** The four KKT conditions. Together they are necessary for any optimum and, for convex problems, sufficient: a point satisfying all four is globally optimal. Complementary slackness is the workhorse, pairing each constraint with its multiplier.

*Remark* 9.9 *(Multipliers are shadow prices).* The multiplier measures how much the optimal value would improve if a constraint were relaxed. If the constraint $g_i(\boldsymbol{w}) \le u_i$ is loosened by raising $u_i$ from zero, the optimal value changes at rate $\partial f^\star / \partial u_i = -\alpha_i^\star$. So $\alpha_i^\star$ is the *shadow price* of constraint $i$: the marginal value of one more unit of that resource. A binding constraint with a large multiplier is expensive to satisfy and worth relaxing; an inactive one has price zero and can be ignored. This is the precise sense in which the epigraph's "buyer and seller" agree at the optimum, and it is why the dual variables carry genuine economic meaning rather than mere algebraic bookkeeping.

**Example 9.10 (Strong duality on an inequality).** Minimize $x^2$ subject to $x \ge 1$, i.e. $g(x) = 1 - x \le 0$. The Lagrangian is $\mathcal{L} = x^2 + \alpha(1 - x)$; minimizing over $x$ gives $x = \alpha/2$ and $\theta(\alpha) = -\frac{\alpha^2}{4} + \alpha$. The dual maximum is at $\alpha^\star = 2$, where $\theta(2) = 1$, matching the primal optimum $x^\star = 1$, $f = 1$. The multiplier $\alpha^\star = 2 > 0$ signals that the constraint is active at the optimum.

## 9.7 The Karush–Kuhn–Tucker conditions

Duality crystallizes into a checklist that any optimum must satisfy, and that *certifies* an optimum for convex problems.

**Theorem 9.11 (KKT conditions).** Under mild regularity, if $(\boldsymbol{w}^\star, \boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star)$ solves Eq. (9.1) then it satisfies the four *KKT conditions*:

$$\begin{aligned} &\text{(stationarity)} && \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}^\star, \boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star) = \mathbf{0}, \\ &\text{(primal feasibility)} && g_i(\boldsymbol{w}^\star) \le 0, \quad h_j(\boldsymbol{w}^\star) = 0, \\ &\text{(dual feasibility)} && \alpha_i^\star \ge 0, \\ &\text{(complementary slackness)} && \alpha_i^\star\, g_i(\boldsymbol{w}^\star) = 0. \end{aligned} \tag{9.5}$$

For a convex problem these conditions are also *sufficient*: any point meeting them is a global optimum.

*Stationarity* balances the objective's gradient against the constraints' gradients, each weighted by its price. *Complementary slackness* is the most useful in practice: it says $\alpha_i^\star g_i(\boldsymbol{w}^\star) = 0$, so a constraint that is inactive ($g_i < 0$) must have zero multiplier, and only *active* constraints ($g_i = 0$) can carry a positive price. Geometrically, only the fences you are touching push back; distant fences exert no force. In the SVM this is precisely why only the *support vectors*, the points on the margin, have nonzero multipliers.

**Example 9.12 (KKT, worked).** Minimize $f(w) = (w - 3)^2$ subject to $g(w) = 2 - w \le 0$ (that is, $w \ge 2$). The Lagrangian is $\mathcal{L} = (w - 3)^2 + \alpha(2 - w)$ with $\alpha \ge 0$. Stationarity: $\mathcal{L}'(w) = 2(w - 3) - \alpha = 0$, so $w = 3 + \frac{\alpha}{2}$. Complementary slackness: $\alpha(2 - w) = 0$, so either $\alpha = 0$ or $w = 2$. *Case $\alpha = 0$:* stationarity gives $w = 3$, which is feasible ($3 \ge 2$), with $f(3) = 0$. *Case $w = 2$:* stationarity gives $\alpha = 2(2 - 3) = -2 < 0$, violating dual feasibility, so it is rejected. Hence $w^\star = 3$, $\alpha^\star = 0$: the unconstrained minimum already lies inside the feasible region, so the constraint is inactive and carries no price. One checks all four KKT conditions hold.

**Example 9.13 (KKT with an *active* constraint, in 2-D).** Now a case where the constraint bites. Minimize $f(x, y) = (x - 2)^2 + (y - 2)^2$ subject to $g(x, y) = x + y - 2 \le 0$. The unconstrained minimum $(2, 2)$ has $x + y = 4 > 2$, so it is infeasible and the constraint must be *active*. Form $\mathcal{L} = (x - 2)^2 + (y - 2)^2 + \alpha(x + y - 2)$, $\alpha \ge 0$. **Stationarity:** $\partial_x \mathcal{L} = 2(x - 2) + \alpha = 0$ and $\partial_y \mathcal{L} = 2(y - 2) + \alpha = 0$, so $x = y$ (both equal $2 - \frac{\alpha}{2}$). **Complementary slackness:**

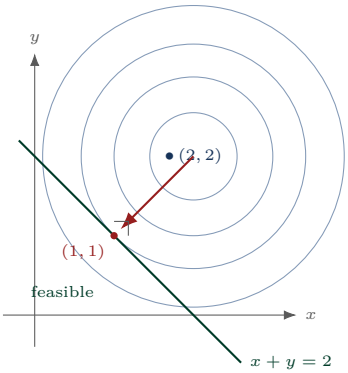



**Figure 9.3.** The active-constraint KKT example. The objective's circular contours are centered at the unconstrained optimum $(2, 2)$, which is infeasible. The constrained optimum $(1, 1)$ is where the smallest feasible contour touches the boundary line $x + y = 2$, i.e. the foot of the perpendicular from $(2, 2)$ to the line. The gradient of $f$ there points along the constraint normal, the stationarity condition.

$\alpha(x+y-2) = 0$. Since we argued $\alpha > 0$, the constraint is tight: $x+y = 2$, hence $x = y = 1$. **Recover the multiplier:** stationarity gives $\alpha = 2(2 - x) = 2(2 - 1) = 2 \geq 0$, satisfying dual feasibility. So $(x^\star, y^\star) = (1, 1)$ with $f = 2$ and $\alpha^\star = 2$. Geometrically $(1, 1)$ is the point of the half-plane $x + y \leq 2$ nearest the unconstrained optimum $(2, 2)$, the foot of the perpendicular onto the boundary line. The shadow price $\alpha^\star = 2$ says relaxing the budget to $x + y \leq 2 + \epsilon$ would lower the objective by about $2\epsilon$, the mirror image of Example 9.12, where an inactive constraint had $\alpha^\star = 0$.

## 9.8 Why this matters: the SVM ahead

The support vector machine of Chapter 10 is a convex quadratic program: minimize $\frac{1}{2}\|\boldsymbol{w}\|^2$ subject to the margin constraints $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$. Forming its Lagrangian, taking the dual, and reading off the KKT conditions turns it into a cleaner problem in the multipliers $\alpha_i$, where complementary slackness reveals that almost all $\alpha_i = 0$ and only the support vectors remain. Every tool of this chapter, the Lagrangian, the dual, strong duality, and complementary slackness, is used there in sequence. The constrained optimization theory is not an aside; it is the derivation engine for the most important classifier in the course.

## 9.9 Worked constrained problems

The examples here exercise the machinery end to end: an inequality-constrained quadratic program solved through KKT, a multivariable equality problem, and a numerical check of weak duality.

**Example 9.14 (A quadratic program with an active constraint).** Minimize $f(x, y) = (x - 1)^2 + (y - 1)^2$ subject to $2x + y \leq 2$. First test the unconstrained minimum $(1, 1)$: it gives $2(1) + 1 = 3 > 2$, violating the constraint, so the constraint is *active* and the optimum lies on the line $2x + y = 2$. Geometrically the solution is the point of that line closest to $(1, 1)$, the foot of the perpendicular:

$$(x, y) = (1, 1) - \frac{2(1) + 1 - 2}{2^2 + 1^2}(2, 1) = (1, 1) - \tfrac{1}{5}(2, 1) = (0.6,\ 0.8).$$

The objective there is $(0.6 - 1)^2 + (0.8 - 1)^2 = 0.16 + 0.04 = 0.2$. Check KKT with the constraint written $g(x, y) = 2x + y - 2 \leq 0$: stationarity $\nabla f + \mu \nabla g = \mathbf{0}$ reads $2(-0.4, -0.2) + \mu(2, 1) = \mathbf{0}$, giving $\mu = 0.4 \geq 0$, and the constraint is tight, so complementary slackness holds. All KKT conditions are met, certifying the global minimum of this convex program (Fig. 9.4).

**Example 9.15 (A multivariable equality constraint).** Minimize $x^2+2y^2+z^2$ subject to $x+y+z = 6$. Stationarity of the Lagrangian gives $\nabla(x^2 + 2y^2 + z^2) = \lambda \nabla(x + y + z)$, i.e. $(2x, 4y, 2z) = \lambda(1, 1, 1)$. So $x = z = \frac{\lambda}{2}$ and $y = \frac{\lambda}{4}$. Substituting into the constraint,

$$\tfrac{\lambda}{2} + \tfrac{\lambda}{4} + \tfrac{\lambda}{2} = \tfrac{5\lambda}{4} = 6 \implies \lambda = \tfrac{24}{5} = 4.8,$$

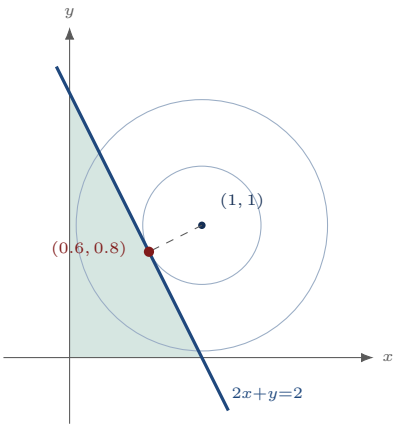



**Figure 9.4.** The quadratic program. Objective contours are circles about the unconstrained minimum $(1, 1)$; the feasible region is the shaded half-plane $2x + y \le 2$. The constrained optimum $(0.6, 0.8)$ is where a contour just touches the constraint line, the foot of the perpendicular from $(1, 1)$.

so $x = z = 2.4$, $y = 1.2$, with minimum value $2.4^2 + 2(1.2^2) + 2.4^2 = 5.76 + 2.88 + 5.76 = 14.4$. Notice $y$ is smallest: its heavier coefficient ($2y^2$) makes spending the budget on $y$ costly, so the optimizer leans on the cheaper $x$ and $z$. The multiplier $\lambda = 4.8$ is the rate at which the optimum would rise if the budget 6 were increased.

**Example 9.16 (Weak duality, checked on numbers).** Take the primal max $x_1 + x_2$ subject to $x_1 \le 3$, $x_2 \le 2$, $\boldsymbol{x} \ge \mathbf{0}$, whose optimum is plainly $(3, 2)$ with value 5. Its dual is min $3y_1 + 2y_2$ subject to $y_1 \ge 1$, $y_2 \ge 1$, $\boldsymbol{y} \ge \mathbf{0}$, with optimum $(1, 1)$ and value $3 + 2 = 5$. Weak duality says every feasible primal value is at most every feasible dual value. Test a non-optimal pair: the primal point $(2, 1)$ gives 3, and the dual point $(2, 2)$ gives $3(2) + 2(2) = 10$, and indeed $3 \le 10$. As both are pushed to their optima the values squeeze together to the common 5, the strong-duality equality. The dual optimum $(1, 1)$ are the shadow prices: relaxing $x_1 \le 3$ to $x_1 \le 4$ would raise the primal optimum by $y_1 = 1$.

**Example 9.17 (A linear program by vertex enumeration).** Maximize $2x + 3y$ subject to $x + y \le 5$, $x \le 3$, $y \le 4$, $x, y \ge 0$. The feasible region is a polygon, and a linear objective is maximized at a corner, so evaluate the objective at each vertex:

$$(0, 0) : 0, \quad (3, 0) : 6, \quad (3, 2) : 12, \quad (1, 4) : 14, \quad (0, 4) : 12,$$

where $(3, 2)$ is the meet of $x = 3$ and $x + y = 5$, and $(1, 4)$ the meet of $y = 4$ and $x + y = 5$. The maximum is 14 at $(1, 4)$, where the constraints $y \le 4$ and $x + y \le 5$ are both active. The objective increases along $\boldsymbol{c} = (2, 3)$, and the corner farthest in that direction wins (Fig. 9.5).

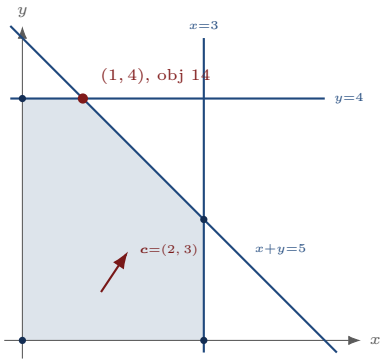



**Figure 9.5.** The feasible polygon (shaded) and the objective direction $\boldsymbol{c} = (2, 3)$. The maximum sits at the vertex $(1, 4)$, the corner farthest along $\boldsymbol{c}$, where the constraints $y \le 4$ and $x + y \le 5$ are both active.

**Example 9.18 (The dual and complementary slackness).** The LP of Example 9.17 has one dual variable per constraint. With $\boldsymbol{c} = (2, 3)$, $\boldsymbol{b} = (5, 3, 4)$, and constraint matrix rows $(1, 1), (1, 0), (0, 1)$, the dual is

$$\min\ 5y_1 + 3y_2 + 4y_3 \quad \text{s.t.} \quad y_1 + y_2 \geq 2, \quad y_1 + y_3 \geq 3, \quad \boldsymbol{y} \geq \boldsymbol{0}.$$

At the primal optimum $(1, 4)$, constraints 1 ($x + y \leq 5$) and 3 ($y \leq 4$) are active while constraint 2 ($x \leq 3$) is slack, so complementary slackness forces $y_2 = 0$. Since both primal variables are positive, both dual constraints are tight: $y_1 + y_2 = 2$ gives $y_1 = 2$, and $y_1 + y_3 = 3$ gives $y_3 = 1$. The dual objective is $5(2) + 3(0) + 4(1) = 14$, equal to the primal, confirming strong duality. The shadow prices $y_1 = 2$ and $y_3 = 1$ say loosening $x + y \leq 5$ or $y \leq 4$ by one unit would raise the optimum by 2 or 1 respectively, while the slack constraint $x \leq 3$ has price 0.

**Example 9.19 (An inactive constraint).** Minimize $(x-1)^2 + (y-1)^2$ subject to $x + y \leq 5$. Test the unconstrained minimum $(1, 1)$: it gives $1 + 1 = 2 \leq 5$, satisfied with slack, so the constraint is *inactive*. By complementary slackness an inactive constraint forces its multiplier to zero, and the problem reduces to the unconstrained one. The minimum is $(1, 1)$ with value 0. Always check whether the unconstrained optimum is already feasible; only a binding constraint bends the solution, and here it does not.

**Example 9.20 (Minimum norm to a hyperplane).** Minimize $x^2 + y^2$ subject to $3x + 4y = 10$. Lagrange gives $(2x, 2y) = \lambda(3, 4)$, so $(x, y) = \frac{\lambda}{2}(3, 4)$. The constraint $3x + 4y = 10$ becomes $\frac{\lambda}{2}(9 + 16) = \frac{25\lambda}{2} = 10$, so $\lambda = \frac{4}{5}$ and $(x, y) = \frac{2}{5}(3, 4) = (1.2, 1.6)$. The minimum value is $1.2^2 + 1.6^2 = 1.44 + 2.56 = 4$, the squared distance from the origin to the line, which the closed form $b^2/\|\boldsymbol{a}\|^2 = 10^2/25 = 4$ confirms. The optimal point lies along the line's normal $(3, 4)$, the foot of the perpendicular from the origin, exactly the geometry the pseudoinverse computes for the minimum-norm solution of Chapter 4.

## 9.10 Summary

- Constrained problems minimize $f$ subject to $g_i \leq 0$, $h_j = 0$. Linear programs have a polytope feasible set and an optimum at a vertex; the simplex method walks improving vertices to it.
- The Lagrangian $\mathcal{L} = f + \sum_i \alpha_i g_i + \sum_j \beta_j h_j$ prices the constraints. The dual function $\theta(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \inf_{\boldsymbol{w}} \mathcal{L}$ and the dual problem maximizes it.
- Weak duality always holds: $\theta \leq f(\boldsymbol{w}^\star)$. For convex problems with Slater's condition, strong duality holds and the gap is zero, so dual and primal optima coincide.
- The KKT conditions, stationarity, primal feasibility, dual feasibility, and complementary slackness, are necessary for optimality and sufficient when convex. Complementary slackness means only active constraints carry positive multipliers.
- This machinery derives the SVM in Chapter 10, where the support vectors are exactly the points with nonzero multipliers.

## Exercises

**Exercise 9.1.** Maximize $2x_1 + 3x_2$ subject to $x_1 + x_2 \leq 4$, $x_1 + 2x_2 \leq 6$, and $x_1, x_2 \geq 0$. Find the optimum by identifying the feasible vertices.

*Solution.* The feasible region is bounded by the axes and the two constraint lines. Its vertices are $(0, 0)$, $(4, 0)$ (where $x_1 + x_2 = 4$ meets $x_2 = 0$), $(0, 3)$ (where $x_1 + 2x_2 = 6$ meets $x_1 = 0$), and the intersection of $x_1 + x_2 = 4$ with $x_1 + 2x_2 = 6$: subtracting gives $x_2 = 2$, then $x_1 = 2$, so $(2, 2)$. The objective $2x_1 + 3x_2$ at these vertices is 0, 8, 9, 10. The maximum is 10 at $(2, 2)$, where both constraints are active. □

**Exercise 9.2.** For $\min x_1^2 + x_2^2$ subject to $x_1 + x_2 = 1$, form the Lagrangian, compute the dual function $\theta(\beta)$, maximize it, and verify strong duality against the primal optimum.

*Solution.* The Lagrangian is $\mathcal{L} = x_1^2 + x_2^2 + \beta(x_1 + x_2 - 1)$. Stationarity $2x_i + \beta = 0$ gives $x_1 = x_2 = -\beta/2$; substituting, $\theta(\beta) = 2(\beta^2/4) + \beta(-\beta - 1) = -\beta^2/2 - \beta$. Then $\theta'(\beta) = -\beta - 1 = 0$ gives $\beta^\star = -1$ and $\theta(-1) = \frac{1}{2}$. The primal optimum is $x_1 = x_2 = \frac{1}{2}$ (the closest point on the line to the origin), with value $\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$. Dual maximum equals primal minimum, so strong duality holds, as guaranteed by convexity with an affine constraint. □

**Exercise 9.3.** Solve $\min x^2 + y^2$ subject to $x + y \geq 2$ using the KKT conditions.

*Solution.* Write the constraint as $g(x, y) = 2 - x - y \leq 0$ and form $\mathcal{L} = x^2 + y^2 + \alpha(2 - x - y)$, $\alpha \geq 0$. Stationarity: $2x - \alpha = 0$ and $2y - \alpha = 0$, so $x = y = \alpha/2$. Complementary slackness: $\alpha(2 - x - y) = 0$. If $\alpha = 0$ then $x = y = 0$, but that violates $x + y \geq 2$ (primal feasibility), so it is rejected. Hence the constraint is active, $x + y = 2$, giving $2 \cdot (\alpha/2) = 2$, so $\alpha^\star = 2$ and $x^\star = y^\star = 1$. All KKT conditions hold ($\alpha = 2 \geq 0$, $x + y = 2$, slackness $2 \cdot 0 = 0$), and since the problem is convex this is the global minimum, with value $1 + 1 = 2$. Geometrically, $(1, 1)$ is the point of the half-plane $x + y \geq 2$ closest to the origin. □

**Exercise 9.4.** State the four KKT conditions for the problem $\min f(\boldsymbol{w})$ subject to $g_i(\boldsymbol{w}) \leq 0$, and explain what complementary slackness implies about inactive constraints.

*Solution.* With Lagrangian $\mathcal{L}(\boldsymbol{w}, \boldsymbol{\alpha}) = f(\boldsymbol{w}) + \sum_i \alpha_i g_i(\boldsymbol{w})$, the KKT conditions at an optimum $(\boldsymbol{w}^\star, \boldsymbol{\alpha}^\star)$ are: **stationarity** $\nabla_{\boldsymbol{w}} \mathcal{L} = \mathbf{0}$; **primal feasibility** $g_i(\boldsymbol{w}^\star) \leq 0$; **dual feasibility** $\alpha_i^\star \geq 0$; and **complementary slackness** $\alpha_i^\star g_i(\boldsymbol{w}^\star) = 0$. Complementary slackness forces a product to be zero, so for each $i$ at least one factor vanishes. If a constraint is *inactive*, $g_i(\boldsymbol{w}^\star) < 0$, then its multiplier must be zero, $\alpha_i^\star = 0$: it exerts no force on the solution and could be removed without changing the optimum. Only *active* constraints ($g_i = 0$) can carry a positive multiplier. This is exactly why, in the SVM, only the support vectors (the points exactly on the margin) have nonzero multipliers. □

PART VI

# Machine Learning Applications

CHAPTER 10

# Supervised Learning: Classification, SVMs, and Kernels

*Of all the lines that separate the data, take the one that stays farthest from it.*

*the max-margin principle*

**Roadmap**

This chapter is where the machinery pays off. We build classifiers in increasing sophistication: the perceptron, logistic regression, and then the *support vector machine*, which chooses the separating hyperplane of widest margin. The SVM is a constrained quadratic program, so its derivation runs straight through Chapter 9: form the Lagrangian, take the dual, and read off the KKT conditions. Complementary slackness reveals that only a few points, the *support vectors*, matter. Because the dual depends on the data only through inner products, the *kernel trick* then buys nonlinear boundaries for free.

**Suggested reading:** Deisenroth et al. [5], Chapter 12; Bishop [2], Chapter 7; the original Cortes and Vapnik [4].

## 10.1 Binary classification

A binary classifier maps a feature vector $\boldsymbol{x} \in \mathbb{R}^d$ to a label. For the SVM it is convenient to take labels in $\{-1, +1\}$ rather than $\{0, 1\}$. A *linear* classifier scores a point by $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ and predicts $\operatorname{sign}(f(\boldsymbol{x}))$; the *decision boundary* is the hyperplane $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$. The questions are how to choose $(\boldsymbol{w}, b)$ and, when no line separates the classes, what to do instead.

## 10.2 The perceptron

The oldest answer is the *perceptron*. It sweeps through the data and, on each misclassified point (one with $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \le 0$), nudges the weights toward it:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + y_i \boldsymbol{x}_i, \qquad b \leftarrow b + y_i. \tag{10.1}$$

If the data are linearly separable, this converges to *some* separating hyperplane in finitely many steps. But it stops at the first boundary that works, which may pass arbitrarily close to the data and generalize poorly, and it never converges on non-separable data. Both defects motivate better criteria.

**Example 10.1 (A perceptron run, every update).** Take three points $\boldsymbol{x}_1 = (2, 1), y_1 = +1$; $\boldsymbol{x}_2 = (-1, 1), y_2 = -1$; $\boldsymbol{x}_3 = (0, -1), y_3 = -1$, and start from $\boldsymbol{w} = (0, 0)$, $b = 0$. A point is misclassified when $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \le 0$; we sweep in order.

- $\boldsymbol{x}_1$**:** $y_1(\boldsymbol{w}^\top \boldsymbol{x}_1 + b) = 1 \cdot 0 = 0 \le 0$, update. $\boldsymbol{w} \leftarrow (0, 0) + 1 \cdot (2, 1) = (2, 1)$, $b \leftarrow 0 + 1 = 1$.
- $\boldsymbol{x}_2$**:** $\boldsymbol{w}^\top \boldsymbol{x}_2 + b = 2(-1) + 1(1) + 1 = 0$, so $y_2 \cdot 0 = 0 \le 0$, update. $\boldsymbol{w} \leftarrow (2, 1) + (-1)(-1, 1) = (3, 0)$, $b \leftarrow 1 - 1 = 0$.
- $\boldsymbol{x}_3$**:** $\boldsymbol{w}^\top \boldsymbol{x}_3 + b = 3(0) + 0(-1) + 0 = 0 \le 0$, update. $\boldsymbol{w} \leftarrow (3, 0) + (-1)(0, -1) = (3, 1)$, $b \leftarrow 0 - 1 = -1$.

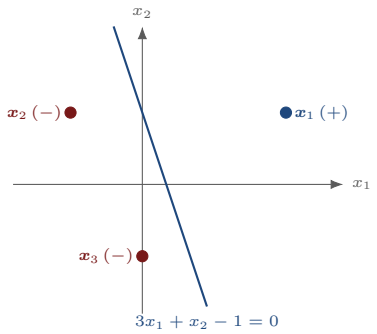



**Figure 10.1.** The separator found in Example 10.1. After three updates the perceptron settles on $3x_1 + x_2 = 1$, with the single positive point on one side and both negatives on the other. The line is some valid separator, generally not the maximum-margin one.

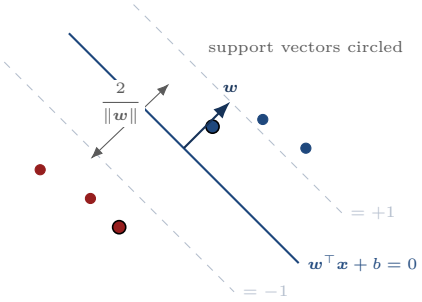



**Figure 10.2.** The maximum-margin hyperplane. The solid line is the decision boundary; the dashed gutters $\boldsymbol{w}^\top \boldsymbol{x} + b = \pm 1$ bound the margin of width $2/\|\boldsymbol{w}\|$. The circled points lying on the gutters are the support vectors; they alone determine the boundary. Maximizing the margin means minimizing $\|\boldsymbol{w}\|$.

**Second sweep (check for convergence):** with $\boldsymbol{w} = (3, 1)$, $b = -1$,

$$\boldsymbol{x}_1:\ 3(2) + 1(1) - 1 = 6,\quad y_1 \cdot 6 = +6 > 0; \qquad \boldsymbol{x}_2:\ -3,\quad y_2(-3) = +3 > 0; \qquad \boldsymbol{x}_3:\ -2,\quad y_3(-2) = +2 > 0.$$

All three are now classified correctly, so the perceptron **converges** with boundary $3x_1 + x_2 - 1 = 0$ (Fig. 10.1). A different visiting order, or a fourth point, could yield a different separating line: the perceptron finds *a* separator rather than the best one, which is precisely the gap the maximum-margin SVM closes.

## 10.3 Logistic regression as a classifier

Chapter 8 fit logistic regression by minimizing the convex cross-entropy loss; as a classifier it predicts class 1 when $\sigma(\boldsymbol{w}^\top \boldsymbol{x} + b) > \frac{1}{2}$, i.e. when $\boldsymbol{w}^\top \boldsymbol{x} + b > 0$, so its boundary is also a hyperplane. Its strength is a calibrated *probability* $\sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$, and every point contributes to the fit. The SVM takes the opposite stance: it ignores points far from the boundary and cares only about the hardest, closest ones. We contrast the two in Section 10.9.

## 10.4 Maximum-margin classification and the geometric margin

When many hyperplanes separate the data, the SVM picks the one with the largest *margin*, the widest empty corridor around the boundary. Place two parallel "gutter" hyperplanes $\boldsymbol{w}^\top \boldsymbol{x} + b = +1$ and $\boldsymbol{w}^\top \boldsymbol{x} + b = -1$ on either side, and scale $(\boldsymbol{w}, b)$ so that every point sits on the correct side of its gutter:

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 \quad \text{for all } i. \tag{10.2}$$

**Proposition 10.2 (Geometric margin).** The distance between the two gutters is $2/\|\boldsymbol{w}\|$.

*Proof.* Let $\boldsymbol{x}_+$ lie on the $+1$ gutter and $\boldsymbol{x}_-$ on the $-1$ gutter. The margin width is the projection of $\boldsymbol{x}_+ - \boldsymbol{x}_-$ onto the unit normal $\boldsymbol{w}/\|\boldsymbol{w}\|$:

$$\text{width} = \frac{\boldsymbol{w}^\top(\boldsymbol{x}_+ - \boldsymbol{x}_-)}{\|\boldsymbol{w}\|} = \frac{(\boldsymbol{w}^\top \boldsymbol{x}_+ + b) - (\boldsymbol{w}^\top \boldsymbol{x}_- + b)}{\|\boldsymbol{w}\|} = \frac{1-(-1)}{\|\boldsymbol{w}\|} = \frac{2}{\|\boldsymbol{w}\|}.$$

■

Maximizing $2/\|\boldsymbol{w}\|$ is the same as minimizing $\frac{1}{2}\|\boldsymbol{w}\|^2$, which gives the hard-margin SVM.

## 10.5 The hard-margin SVM and its dual

**Hard-margin SVM (primal)**

$$\min_{\boldsymbol{w},b} \ \tfrac{1}{2}\|\boldsymbol{w}\|^2 \qquad \text{subject to} \qquad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \quad i = 1, \dots, m.$$

This is a convex quadratic program, exactly the setting of Chapter 9. Writing the constraints as $g_i = 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \leq 0$ with multipliers $\alpha_i \geq 0$, the Lagrangian is

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \tfrac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{m} \alpha_i \big(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)\big). \tag{10.3}$$

Stationarity (the first KKT condition) gives the two key identities

$$\nabla_{\boldsymbol{w}} \mathcal{L} = \boldsymbol{0} \ \Rightarrow \ \boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i, \qquad \frac{\partial \mathcal{L}}{\partial b} = 0 \ \Rightarrow \ \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{10.4}$$

The first says the weight vector is a combination of the training points, weighted by $\alpha_i y_i$. Substituting both back into $\mathcal{L}$ eliminates $\boldsymbol{w}$ and $b$ and leaves a problem in $\boldsymbol{\alpha}$ alone.

**Theorem 10.3 (SVM dual).** The hard-margin SVM is equivalent to the dual quadratic program

$$\max_{\boldsymbol{\alpha} \geq \boldsymbol{0}} \ \sum_{i=1}^{m} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \, (\boldsymbol{x}_i^\top \boldsymbol{x}_j) \quad \text{subject to} \quad \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{10.5}$$

By complementary slackness, $\alpha_i \big[1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)\big] = 0$, so $\alpha_i > 0$ only for points on the margin (where $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = 1$): these are the *support vectors*. The solution is $\boldsymbol{w}^\star = \sum_{i \in \mathrm{SV}} \alpha_i^\star y_i \boldsymbol{x}_i$, and $b$ is recovered from any support vector via $b = y_i - \sum_j \alpha_j^\star y_j (\boldsymbol{x}_j^\top \boldsymbol{x}_i)$.

The dual is the payoff of Chapter 9: it is again convex, it depends on the data only through the inner products $\boldsymbol{x}_i^\top \boldsymbol{x}_j$ (which Section 10.7 exploits), and complementary slackness makes the solution *sparse*, supported on a handful of points.

**Example 10.4 (A two-point SVM, solved through the dual).** Take $\boldsymbol{x}_1 = (0, 0)$ with $y_1 = -1$ and $\boldsymbol{x}_2 = (2, 0)$ with $y_2 = +1$. The constraint $\sum_i \alpha_i y_i = 0$ forces $\alpha_2 - \alpha_1 = 0$, so $\alpha_1 = \alpha_2 = \alpha$. Both points must lie on their gutters (with only two points, both are support vectors), and $\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i = \alpha(2, 0) - \alpha(0, 0) = (2\alpha, 0)$. Requiring $y_2(\boldsymbol{w}^\top \boldsymbol{x}_2 + b) = 1$ with the boundary halfway between the points gives $\boldsymbol{w} = (1, 0)$, hence $2\alpha = 1$ and $\alpha = \frac{1}{2}$. Then $b = -1$, and the margin is $2/\|\boldsymbol{w}\| = 2$. The boundary is the vertical line $x_1 = 1$, equidistant from the two points, exactly where intuition puts it.

**Example 10.5 (A three-point SVM with a non-support-vector).** Now add a third point far from the boundary to see complementary slackness in action. Take

$$\boldsymbol{x}_1 = (3, 0),\ y_1 = +1; \qquad \boldsymbol{x}_2 = (1, 0),\ y_2 = -1; \qquad \boldsymbol{x}_3 = (-2, 0),\ y_3 = -1.$$

The points are collinear (all on the $x_1$-axis), so the needed inner products are just the products of first coordinates:

$$\boldsymbol{x}_1^\top \boldsymbol{x}_1 = 9,\ \boldsymbol{x}_2^\top \boldsymbol{x}_2 = 1,\ \boldsymbol{x}_3^\top \boldsymbol{x}_3 = 4, \quad \boldsymbol{x}_1^\top \boldsymbol{x}_2 = 3,\ \boldsymbol{x}_1^\top \boldsymbol{x}_3 = -6,\ \boldsymbol{x}_2^\top \boldsymbol{x}_3 = -2.$$

**Guess the far point is slack.** Geometrically $\boldsymbol{x}_3 = (-2, 0)$ sits well to the negative side, far behind $\boldsymbol{x}_2$, so we expect it to be strictly inside its margin and carry $\alpha_3 = 0$; we will verify this at the end. Dropping it leaves the two-point dual on $\boldsymbol{x}_1, \boldsymbol{x}_2$, whose equality constraint $\sum_i \alpha_i y_i = \alpha_1 - \alpha_2 = 0$ forces $\alpha_1 = \alpha_2 = \alpha$. **Maximize the dual.** With $y_1 y_2 = -1$ and $\boldsymbol{x}_1^\top \boldsymbol{x}_2 = 3$,

$$W(\alpha) = \alpha_1 + \alpha_2 - \tfrac{1}{2}\left[\alpha_1^2(9) + \alpha_2^2(1) + 2\alpha_1\alpha_2(-1)(3)\right] = 2\alpha - \tfrac{1}{2}\left(9\alpha^2 + \alpha^2 - 6\alpha^2\right) = 2\alpha - 2\alpha^2.$$

Setting $W'(\alpha) = 2 - 4\alpha = 0$ gives $\alpha = \tfrac{1}{2}$, so $\alpha_1 = \alpha_2 = \tfrac{1}{2}$. **Recover the classifier.**

$$\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i = \tfrac{1}{2}(3, 0) - \tfrac{1}{2}(1, 0) = (1, 0),$$

and from the support vector $\boldsymbol{x}_1$, $y_1(\boldsymbol{w}^\top \boldsymbol{x}_1 + b) = 1 \Rightarrow 3 + b = 1 \Rightarrow b = -2$. The boundary is $x_1 = 2$, the margin is $2/\|\boldsymbol{w}\| = 2$, and the gutters are $x_1 = 3$ (through $\boldsymbol{x}_1$) and $x_1 = 1$ (through $\boldsymbol{x}_2$). **Verify the slack point.** For $\boldsymbol{x}_3$, $y_3(\boldsymbol{w}^\top \boldsymbol{x}_3 + b) = (-1)(-2 - 2) = 4 \geq 1$, so its constraint holds with slack; complementary slackness $\alpha_3\left[1 - y_3(\boldsymbol{w}^\top \boldsymbol{x}_3 + b)\right] = 0$ then forces $\alpha_3 = 0$, confirming the guess. Only $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are support vectors; the distant $\boldsymbol{x}_3$ could be deleted without moving the boundary at all. This is the SVM's sparsity made concrete.

**Example 10.6 (Classifying a new point, two equivalent ways).** Once trained, the machine labels a test point $\boldsymbol{x}$ by the sign of its score. Use the two-point machine above: support vectors $\boldsymbol{x}_1 = (0, 0)$ with $y_1 = -1$ and $\boldsymbol{x}_2 = (2, 0)$ with $y_2 = +1$, multipliers $\alpha_1 = \alpha_2 = \tfrac{1}{2}$, weight $\boldsymbol{w} = (1, 0)$, bias $b = -1$. The score can be written with the weight vector,

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b = x_1 - 1,$$

or, equivalently, as the *support-vector expansion* $f(\boldsymbol{x}) = \sum_i \alpha_i y_i (\boldsymbol{x}_i^\top \boldsymbol{x}) + b$, which is the form a kernel needs. Classify $\boldsymbol{x} = (1.5,\ 0.7)$ both ways. With the weight: $f = 1.5 - 1 = 0.5 > 0$, so class $+1$. With the expansion,

$$f = \tfrac{1}{2}(-1)\underbrace{(\boldsymbol{x}_1^\top \boldsymbol{x})}_{0} + \tfrac{1}{2}(+1)\underbrace{(\boldsymbol{x}_2^\top \boldsymbol{x})}_{2(1.5)+0(0.7)=3} + (-1) = \tfrac{1}{2}(3) - 1 = 0.5,$$

the same value. A second point $\boldsymbol{x} = (0.5,\ 3)$ scores $f = 0.5 - 1 = -0.5 < 0$, class $-1$; notice its large $x_2 = 3$ is irrelevant, because $\boldsymbol{w} = (1, 0)$ ignores that coordinate (the data only ever varied in $x_1$). The expansion view is the one that generalizes: replace each inner product $\boldsymbol{x}_i^\top \boldsymbol{x}$ by a kernel $k(\boldsymbol{x}_i, \boldsymbol{x})$ and the same formula gives a nonlinear classifier. The machine has, in effect, memorized only its support vectors and predicts by a weighted vote of how similar $\boldsymbol{x}$ is to each.

## 10.6 The soft margin

Real data is rarely separable, and one outlier can make the hard-margin constraints contradictory. The *soft margin* permits violations through slack variables $\xi_i \geq 0$, penalized in the objective:

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \ \tfrac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \xi_i \quad \text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \tag{10.6}$$

The hyperparameter $C > 0$ sets the price of violations: large $C$ insists on classifying every point (narrow margin, risk of overfitting an outlier), small $C$ tolerates violations for a wider, smoother margin. Remarkably, the dual is almost unchanged, only the multipliers gain an upper bound:

$$\max_{\boldsymbol{\alpha}} \ \sum_i \alpha_i - \tfrac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i^\top \boldsymbol{x}_j) \quad \text{s.t.} \quad 0 \leq \alpha_i \leq C, \ \ \sum_i \alpha_i y_i = 0. \tag{10.7}$$

The cap $\alpha_i \leq C$ classifies the points into three kinds: $\alpha_i = 0$ are the easy points outside the margin (irrelevant to the boundary); $0 < \alpha_i < C$ are the support vectors exactly on the margin; and $\alpha_i = C$ are the points inside the margin or misclassified. $C$ plays the role of an inverse regularization strength, tuning the bias–variance balance of Chapter 7.

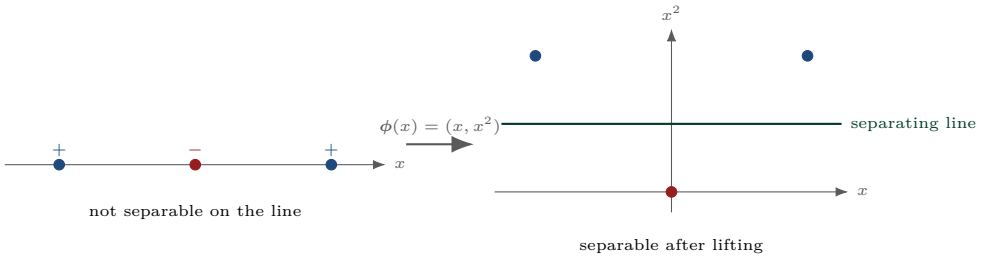



**Figure 10.3.** The kernel trick, lifting the data. On the line, the − point sits between two + points and no threshold separates them. The feature map $\phi(x) = (x, x^2)$ sends each point onto a parabola in the plane, where the horizontal line $x^2 = 1$ separates the classes. A kernel computes the needed inner products in the lifted space without ever building $\phi$.

## 10.7 The kernel trick

The dual Eq. (10.5) touches the data only through inner products $\boldsymbol{x}_i^\top \boldsymbol{x}_j$, never the raw vectors. This is the opening for nonlinearity. Map the data through a feature map $\boldsymbol{\phi} : \mathbb{R}^d \to \mathcal{H}$ into a higher- (possibly infinite-) dimensional space, and the dual needs only the inner products $\boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_j)$.

**Definition 10.7 (Kernel).** A *kernel* is a function $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}')$ that computes the inner product in feature space directly. A symmetric $k$ is a valid kernel (corresponds to some $\boldsymbol{\phi}$) if and only if its Gram matrix $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is positive semidefinite for every dataset (*Mercer's condition*).

The PSD requirement is forced rather than arbitrary. If $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}')$ for some feature map, then for any points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$ and any coefficients $\boldsymbol{c} \in \mathbb{R}^m$,

$$\boldsymbol{c}^\top K \boldsymbol{c} = \sum_{i,j} c_i c_j \, \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_j) = \Big\| \sum_{i=1}^{m} c_i \, \boldsymbol{\phi}(\boldsymbol{x}_i) \Big\|^2 \geq 0, \tag{10.8}$$

so the Gram matrix is automatically PSD. Mercer's theorem is the converse: any continuous symmetric PSD kernel arises from *some* (possibly infinite-dimensional) feature map, so we may design kernels directly and trust a feature space exists behind them.

*Intuition.* A pattern that is hopelessly tangled in the original space can fall apart cleanly once lifted into a higher-dimensional one. Figure 10.3 shows the canonical case: three points on a line, − flanked by two +, that no threshold separates; the map $\phi(x) = (x, x^2)$ raises them onto a parabola where a single horizontal line does the job. The kernel trick performs this lift implicitly, paying only for inner products, never for the (possibly infinite-dimensional) coordinates.

The trick is that we never form $\boldsymbol{\phi}$. Replacing $\boldsymbol{x}_i^\top \boldsymbol{x}_j$ by $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ everywhere in the dual gives a nonlinear classifier $f(\boldsymbol{x}) = \sum_{i \in \mathrm{SV}} \alpha_i y_i \, k(\boldsymbol{x}_i, \boldsymbol{x}) + b$ at the cost of evaluating $k$. A small but telling example: the polynomial kernel $k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^\top \boldsymbol{x}')^2$ on $\mathbb{R}^2$ expands to $(x_1 x_1' + x_2 x_2')^2 = x_1^2 x_1'^2 + 2 x_1 x_2 x_1' x_2' + x_2^2 x_2'^2$, which is the inner product $\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}')$ of the explicit map $\boldsymbol{\phi}(\boldsymbol{x}) = (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$. The kernel evaluates in $\mathbb{R}^2$ what would otherwise be a dot product in $\mathbb{R}^3$; for higher degrees the saving is enormous, and for the RBF kernel the lifted space is infinite-dimensional yet the kernel is a one-line formula.

**Example 10.8 (The kernel trick on numbers).** Take $\boldsymbol{x} = (1, 2)$ and $\boldsymbol{x}' = (3, 1)$ with the quadratic kernel $k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^\top \boldsymbol{x}')^2$. The cheap route works in $\mathbb{R}^2$: $\boldsymbol{x}^\top \boldsymbol{x}' = 1 \cdot 3 + 2 \cdot 1 = 5$, so $k = 5^2 = 25$. The expensive route lifts to $\mathbb{R}^3$ via $\boldsymbol{\phi}(\boldsymbol{x}) = (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$:

$$\boldsymbol{\phi}(\boldsymbol{x}) = (1,\, 2\sqrt{2},\, 4), \qquad \boldsymbol{\phi}(\boldsymbol{x}') = (9,\, 3\sqrt{2},\, 1),$$

$$\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}') = 1 \cdot 9 + (2\sqrt{2})(3\sqrt{2}) + 4 \cdot 1 = 9 + 12 + 4 = 25.$$

Both give 25: the kernel computed the lifted inner product without ever forming the three-dimensional vectors. For the homogeneous degree-$p$ kernel $(\boldsymbol{x}^\top \boldsymbol{x}')^p$ on $\mathbb{R}^d$ the feature space has $\binom{d+p-1}{p}$ coordinates ($\binom{3}{2} = 3$ here), while the inhomogeneous $(\boldsymbol{x}^\top \boldsymbol{x}' + c)^p$ keeps the lower-degree terms too for $\binom{d+p}{p}$; either way the saving grows fast, and for the RBF kernel (infinite-dimensional $\boldsymbol{\phi}$) it is the difference between possible and impossible.

| **Kernel** | $k(\boldsymbol{x}, \boldsymbol{x}')$ | **Feature space** |
|---|---|---|
| Linear | $\boldsymbol{x}^\top \boldsymbol{x}'$ | the input space itself |
| Polynomial | $(\boldsymbol{x}^\top \boldsymbol{x}' + c)^d$ | monomials up to degree $d$ |
| RBF / Gaussian | $\exp(-\gamma \lVert \boldsymbol{x} - \boldsymbol{x}' \rVert^2)$ | infinite-dimensional |
| Sigmoid | $\tanh(\kappa\, \boldsymbol{x}^\top \boldsymbol{x}' + \theta)$ | (not always valid) |

**Table 10.1.** Common kernels. Each replaces the dot product in the SVM dual, buying a nonlinear boundary at the cost of one kernel evaluation per pair. The RBF kernel, with its infinite-dimensional feature space and single width parameter $\gamma$, is the usual default.

The most common choice is the *radial basis function* (Gaussian) kernel

$$k_\gamma(\boldsymbol{x}, \boldsymbol{x}') = \exp\!\big(-\gamma \lVert \boldsymbol{x} - \boldsymbol{x}' \rVert_2^2\big), \tag{10.9}$$

whose feature space is infinite-dimensional: each support vector contributes a Gaussian bump, and $\gamma$ sets its width (small $\gamma$, smooth boundary; large $\gamma$, wiggly boundary that can overfit). A straight cut in the lifted space is a curved boundary back in the original space, the "inflate the balloon" picture: a pattern that tangles in $\mathbb{R}^d$ can come apart cleanly in higher dimensions. The Gram matrix is computed efficiently from $\lVert \boldsymbol{x}_i - \boldsymbol{x}_j \rVert^2 = \lVert \boldsymbol{x}_i \rVert^2 + \lVert \boldsymbol{x}_j \rVert^2 - 2\boldsymbol{x}_i^\top \boldsymbol{x}_j$, vectorizing the pairwise distances through $XX^\top$. Table 10.1 lists the kernels used in practice.

That the RBF feature space is infinite-dimensional is not a slogan; it falls out of a series expansion. Write

$$k_\gamma(\boldsymbol{x}, \boldsymbol{x}') = e^{-\gamma \lVert \boldsymbol{x} \rVert^2}\, e^{-\gamma \lVert \boldsymbol{x}' \rVert^2}\, e^{2\gamma\, \boldsymbol{x}^\top \boldsymbol{x}'}, \qquad e^{2\gamma\, \boldsymbol{x}^\top \boldsymbol{x}'} = \sum_{k=0}^{\infty} \frac{(2\gamma)^k}{k!}\, (\boldsymbol{x}^\top \boldsymbol{x}')^k,$$

using $\lVert \boldsymbol{x} - \boldsymbol{x}' \rVert^2 = \lVert \boldsymbol{x} \rVert^2 + \lVert \boldsymbol{x}' \rVert^2 - 2\boldsymbol{x}^\top \boldsymbol{x}'$. Each term $(\boldsymbol{x}^\top \boldsymbol{x}')^k$ is itself a polynomial kernel of degree $k$ (whose feature map is the degree-$k$ monomials), so the RBF kernel is an infinite weighted sum of polynomial kernels: its feature map stacks monomials of *every* degree, scaled by the Gaussian envelope $e^{-\gamma \lVert \boldsymbol{x} \rVert^2}$. No finite-dimensional $\boldsymbol{\phi}$ could reproduce it, yet the kernel is one exponential.

**Example 10.9 (Computing an RBF Gram matrix by hand).** Take three points $\boldsymbol{x}_1 = (0, 0)$, $\boldsymbol{x}_2 = (1, 0)$, $\boldsymbol{x}_3 = (0, 1)$ with labels $y = (+1, -1, +1)$ and $\gamma = 1$. The squared distances are $\lVert \boldsymbol{x}_1 - \boldsymbol{x}_2 \rVert^2 = 1$, $\lVert \boldsymbol{x}_1 - \boldsymbol{x}_3 \rVert^2 = 1$, $\lVert \boldsymbol{x}_2 - \boldsymbol{x}_3 \rVert^2 = 2$, and 0 on the diagonal, so

$$K = \begin{bmatrix} e^{0} & e^{-1} & e^{-1} \\ e^{-1} & e^{0} & e^{-2} \\ e^{-1} & e^{-2} & e^{0} \end{bmatrix} = \begin{bmatrix} 1 & 0.368 & 0.368 \\ 0.368 & 1 & 0.135 \\ 0.368 & 0.135 & 1 \end{bmatrix}.$$

Every diagonal entry is 1 (a point is maximally similar to itself), and the off-diagonals decay with distance: the nearer pairs (distance$^2 = 1$) share 0.368, the farther pair (distance$^2 = 2$) only 0.135. The matrix the SVM dual actually uses is $P_{ij} = y_i y_j K_{ij}$, the label-signed version,

$$P = \operatorname{diag}(\boldsymbol{y})\, K\, \operatorname{diag}(\boldsymbol{y}) = \begin{bmatrix} 1 & -0.368 & 0.368 \\ -0.368 & 1 & -0.135 \\ 0.368 & -0.135 & 1 \end{bmatrix},$$

which is symmetric PSD by Eq. (10.8); feeding it to a QP solver returns the multipliers $\boldsymbol{\alpha}$. One checks $K$ is PSD directly: its eigenvalues are 1.59, 0.86, 0.54, all positive.

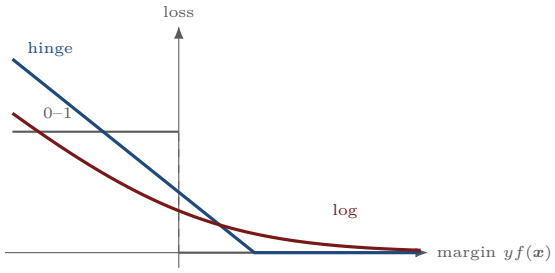



**Figure 10.4.** Surrogate losses as functions of the margin $yf(\boldsymbol{x})$. The true 0–1 loss (grey step) is not differentiable, so both methods minimize a convex surrogate above it: the SVM's hinge loss $\max(0, 1 - yf)$ and logistic regression's log loss $\ln(1 + e^{-yf})$. The hinge is *exactly zero* past the margin, which kills the gradient of easy points and makes the SVM sparse; the log loss is always positive, so every point contributes.

## 10.8 Solving it: the quadratic program and multiclass

The dual Eq. (10.7) is a *quadratic program*: maximize a quadratic in $\boldsymbol{\alpha}$ subject to one linear equality and box constraints $0 \le \alpha_i \le C$. Writing $Q_{ij} = y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, it is $\max_{\boldsymbol{\alpha}} \ \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^\top Q \boldsymbol{\alpha}$ over the box and the equality, which any QP solver handles. The specialized *sequential minimal optimization* (SMO) algorithm [18] exploits the structure by optimizing two multipliers at a time (the smallest update that respects $\sum_i \alpha_i y_i = 0$), which is what the coding assignments implement.

The SVM is binary by construction; for $K$ classes one wraps it. *One-vs-rest* trains $K$ classifiers, each separating one class from all others, and predicts the highest-scoring. *One-vs-one* trains a classifier for each of the $\binom{K}{2}$ pairs and predicts by majority vote. Both reduce multiclass classification to the binary machine above.

## 10.9 SVM versus logistic regression

Both draw a hyperplane, but they optimize different losses. Logistic regression minimizes the log loss, which every point influences, and returns a probability. The SVM minimizes (in its equivalent unconstrained form) the *hinge loss* $\max(0, 1 - y_i f(\boldsymbol{x}_i))$ plus $\frac{1}{2C}\|\boldsymbol{w}\|^2$, which is exactly zero for points safely beyond the margin, so only the support vectors enter the solution. The SVM gives a sparse, margin-maximizing boundary with no probabilities; logistic regression gives calibrated probabilities using all the data. With a kernel the SVM draws nonlinear boundaries cheaply, which is its signature advantage. Figure 10.4 compares the two losses against the true 0–1 loss they approximate.

**Example 10.10 (Hinge loss, point by point).** Put numbers on the hinge to see where sparsity comes from. Reuse the trained machine $\boldsymbol{w} = (1, 0)$, $b = -1$, whose score is $f(\boldsymbol{x}) = x_1 - 1$, and evaluate the per-point loss $\ell = \max\big(0,\, 1 - y\, f(\boldsymbol{x})\big)$. The quantity $y\, f(\boldsymbol{x})$ is the *margin*: positive means correctly classified, and greater than 1 means safely past the margin.

| point $\boldsymbol{x}$ | label $y$ | score $f(\boldsymbol{x})$ | margin $yf(\boldsymbol{x})$ | hinge $\ell$ |
|---|---|---|---|---|
| $(2, 0)$ | $+1$ | $+1.0$ | $+1.0$ | $0$ |
| $(3, 0)$ | $+1$ | $+2.0$ | $+2.0$ | $0$ |
| $(1.5, 0)$ | $+1$ | $+0.5$ | $+0.5$ | $0.5$ |
| $(0.5, 0)$ | $+1$ | $-0.5$ | $-0.5$ | $1.5$ |

The point at $(3, 0)$ is deep in its own region with margin 2, so it pays nothing and, crucially, exerts no force on the boundary: move it farther away and the loss stays 0. The point exactly on the margin at $(2, 0)$ also pays 0 but sits at the kink, where it can still be a support vector. The interesting cases are the last two: $(1.5, 0)$ is correctly classified yet inside the margin, so it pays 0.5, and $(0.5, 0)$ is outright misclassified ($f < 0$ for a $+1$ point), paying 1.5. Only points with margin below 1 have a nonzero loss and a nonzero gradient, which is exactly why the SVM's solution depends on the support vectors alone and ignores the easy majority. The hinge is a hard bouncer: clear the velvet rope at margin 1 and you are waved through free, fall short and you pay in linear proportion to how far short you fell.

| Classifier | Loss | Probabilities? | Sparse? | Nonlinear? |
|---|---|---|---|---|
| Perceptron | misclassification | no | no | no |
| Logistic regression | log loss | yes | no | via features |
| Hard-margin SVM | none (separable) | no | yes (SVs) | via kernel |
| Soft-margin SVM | hinge $+\,\ell_2$ | no | yes (SVs) | via kernel |

**Table 10.2.** The linear classifiers of this chapter. The SVM's hinge loss buys sparsity (only support vectors matter) and the kernel trick buys nonlinearity; logistic regression gives probabilities at the cost of using every point.

## 10.10 Evaluating a classifier: the confusion matrix

A trained classifier is only as good as the test-set tally of where it is right and wrong. Sort the predictions into the four cells of the *confusion matrix* (Fig. 10.5): true positives $TP$ and true negatives $TN$ on the diagonal (the correct calls), false positives $FP$ and false negatives $FN$ off it (the two kinds of error). From these four counts come the standard scores,

$$\text{accuracy} = \frac{TP+TN}{\text{total}}, \quad \text{precision} = \frac{TP}{TP+FP}, \quad \text{recall} = \frac{TP}{TP+FN}, \quad F_1 = \frac{2\,\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}}.$$

Precision asks "of the points I flagged positive, how many really were?"; recall asks "of the truly positive points, how many did I catch?" The two trade off: a cautious classifier that flags only sure cases has high precision but low recall, while an eager one has the reverse. The $F_1$ score, their harmonic mean, rewards balancing the two and collapses toward whichever is smaller.

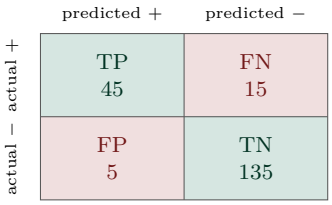



**Figure 10.5.** The confusion matrix for the spam filter of Example 10.11. Correct calls sit on the green diagonal ($TP$, $TN$); the two error types sit off it ($FN$, a missed positive; $FP$, a false alarm). Every metric is a ratio of these four counts.

**Example 10.11 (Reading a confusion matrix).** A spam filter is tested on 200 messages, of which 60 are spam. It catches 45 spam ($TP = 45$), misses 15 ($FN = 15$), wrongly flags 5 good messages ($FP = 5$), and clears the other 135 ($TN = 135$), as tallied in Fig. 10.5. Then

$$\text{accuracy} = \frac{45+135}{200} = 0.90, \quad \text{precision} = \frac{45}{45+5} = 0.90, \quad \text{recall} = \frac{45}{45+15} = 0.75,$$

and $F_1 = \dfrac{2(0.90)(0.75)}{0.90+0.75} = \dfrac{1.35}{1.65} \approx 0.82$. The filter is precise (it rarely junks a good message) but its recall of 0.75 means it lets a quarter of the spam through. Accuracy alone flatters it: a lazy filter that passed *everything* would still score $140/200 = 0.70$ just by being right on the majority class. That is the "accuracy paradox," and it is why precision, recall, and $F_1$ are the honest summaries once the classes are imbalanced.

A classifier with an adjustable threshold, like logistic regression's probability or the SVM's score, does not give a single (precision, recall) pair but a whole curve. Lowering the threshold flags more points, which raises recall but usually lowers precision; raising it does the reverse. Sweeping the threshold and plotting precision against recall (Fig. 10.6) summarizes the classifier at every operating point, and the area under that curve is a single threshold-free quality score. A worthless classifier sits on the flat baseline at precision equal to the positive rate; the further the curve bows above that line, the better.

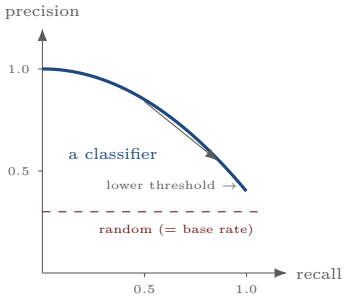



**Figure 10.6.** The precision–recall tradeoff. Sweeping a classifier's threshold traces this curve: at a high threshold it flags only sure cases (high precision, low recall, left); lowering it catches more positives at the cost of false alarms (right). The dashed line is a random classifier, whose precision is just the base rate; the area under the solid curve measures quality independent of any single threshold.

## 10.11 Summary

- Linear classifiers score by $\boldsymbol{w}^\top \boldsymbol{x} + b$. The perceptron finds *a* separator; logistic regression finds a probabilistic one; the SVM finds the *maximum-margin* one.
- The margin is $2/\|\boldsymbol{w}\|$, so the hard-margin SVM minimizes $\frac{1}{2}\|\boldsymbol{w}\|^2$ subject to $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$.
- Its dual (Chapter 9 in action) is $\max_{\boldsymbol{\alpha} \geq \mathbf{0}} \sum_i \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j$ with $\sum_i \alpha_i y_i = 0$. Complementary slackness makes it sparse: only support vectors have $\alpha_i > 0$, and $\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i$.
- The soft margin adds slack and the cap $0 \leq \alpha_i \leq C$, tuning the bias–variance tradeoff.
- The dual uses only inner products, so the kernel trick $\boldsymbol{x}_i^\top \boldsymbol{x}_j \to k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ buys nonlinear boundaries; the RBF kernel $e^{-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2}$ is the default. Multiclass is handled by one-vs-rest or one-vs-one wrappers.

## Exercises

**Exercise 10.1.** Two points are given: $\boldsymbol{x}_1 = (0,0)$ with $y_1 = -1$ and $\boldsymbol{x}_2 = (2,0)$ with $y_2 = +1$. Solve the hard-margin SVM: find the dual variables $\alpha_i$, the weight vector $\boldsymbol{w}$, the bias $b$, the margin width, and the decision boundary.

*Solution.* The equality constraint $\sum_i \alpha_i y_i = -\alpha_1 + \alpha_2 = 0$ gives $\alpha_1 = \alpha_2 = \alpha$. With only two points, both are support vectors. From stationarity, $\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i = \alpha(+1)(2,0) + \alpha(-1)(0,0) = (2\alpha, 0)$. The boundary lies halfway between the points (by symmetry), at $x_1 = 1$, where the unit-margin scaling forces $\boldsymbol{w} = (1,0)$ and $b = -1$ (so $\boldsymbol{w}^\top \boldsymbol{x} + b = x_1 - 1$, which is $-1$ at $\boldsymbol{x}_1$ and $+1$ at $\boldsymbol{x}_2$). Hence $2\alpha = 1$, $\alpha = \frac{1}{2}$. The margin width is $2/\|\boldsymbol{w}\| = 2$, and the decision boundary is the line $x_1 = 1$. □

**Exercise 10.2.** (a) Sketch a linearly separable dataset in $\mathbb{R}^2$ and mark which points would be support vectors. (b) Sketch a dataset that is not linearly separable and explain how the soft margin or a kernel handles it.

*Solution.* (a) For two well-separated clouds, the support vectors are the points of each class *closest* to the gap, the ones lying on the margin gutters; all other points, being farther away, have $\alpha_i = 0$ and could be deleted without changing the boundary. Typically only two or three points per side are support vectors. (b) If the classes overlap, no hyperplane separates them. The *soft margin* (Eq. (10.6)) allows the offending points to violate the margin, paying a penalty $C\xi_i$, and still produces a sensible linear boundary. Alternatively a *kernel* (Section 10.7) lifts the data into a higher-dimensional space where a separating hyperplane exists; for instance, concentric rings in $\mathbb{R}^2$ are separable by a circle, which is a hyperplane in the RBF feature space. □

**Exercise 10.3.** Compute the $2 \times 2$ RBF Gram matrix $K_{ij} = \exp(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$ with $\gamma = 1$ for the points $\boldsymbol{x}_1 = (0,0)$ and $\boldsymbol{x}_2 = (1,1)$, and confirm it is a valid kernel matrix.

*Solution.* The squared distance is $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 = 1^2 + 1^2 = 2$, while $\|\boldsymbol{x}_i - \boldsymbol{x}_i\|^2 = 0$. So

$$K = \begin{bmatrix} e^{0} & e^{-2} \\ e^{-2} & e^{0} \end{bmatrix} = \begin{bmatrix} 1 & e^{-2} \\ e^{-2} & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & 0.135 \\ 0.135 & 1 \end{bmatrix}.$$

It is symmetric, and its eigenvalues are $1 \pm e^{-2}$, both positive, so it is positive definite, hence a valid (Mercer) kernel matrix. Diagonal entries are 1 because a point is maximally similar to itself, and off-diagonal entries lie in $(0,1)$, decaying with distance. □

**Exercise 10.4.** In the soft-margin dual, the multipliers satisfy $0 \le \alpha_i \le C$. Explain what each of the three cases $\alpha_i = 0$, $0 < \alpha_i < C$, and $\alpha_i = C$ says about the $i$th point, and what happens to the boundary as $C \to \infty$.

*Solution.* By the KKT conditions for Eq. (10.6): $\alpha_i = 0$ means the point is correctly classified and strictly outside the margin (it has no influence on $\boldsymbol{w} = \sum_j \alpha_j y_j \boldsymbol{x}_j$); $0 < \alpha_i < C$ means the point sits exactly on its margin gutter (a "free" support vector, $\xi_i = 0$); and $\alpha_i = C$ means the point is inside the margin or misclassified ($\xi_i > 0$, a "bound" support vector). As $C \to \infty$ the penalty on any violation becomes infinite, so the soft margin reverts to the hard margin: the optimizer refuses to let any point cross its gutter, which recovers Eq. (10.5) and, on noisy data, can contort the boundary to fit a single outlier, the overfitting regime. □

**Exercise 10.5.** Both logistic regression and the soft-margin SVM fit a hyperplane. State the loss each minimizes and explain why the SVM solution is sparse in the data while logistic regression is not.

*Solution.* Logistic regression minimizes the *log loss* $-\sum_i [y_i \log p_i + (1 - y_i)\log(1 - p_i)]$, which is strictly positive for every point (no finite score makes it exactly zero), so every training point exerts some pull on the boundary; the fit is dense. The soft-margin SVM minimizes the *hinge loss* $\sum_i \max(0, 1 - y_i f(\boldsymbol{x}_i))$ plus $\frac{1}{2C}\|\boldsymbol{w}\|^2$. The hinge loss is *exactly zero* for any point safely beyond its margin ($y_i f(\boldsymbol{x}_i) \ge 1$), so those points contribute nothing to the gradient and have $\alpha_i = 0$; only the points on or inside the margin (the support vectors) enter the solution. This flat region of the hinge loss is the source of the SVM's sparsity, and it is why an SVM can be stored and evaluated using only its support vectors. □

**Exercise 10.6.** A trained linear classifier has weight vector $\boldsymbol{w} = (3, 4)$ and bias $b = -1$, defining the boundary $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$. (a) What is the width of the SVM margin? (b) How far is the decision boundary from the origin?

*Solution.* Both answers are pure geometry of the hyperplane, and both divide by $\|\boldsymbol{w}\| = \sqrt{3^2 + 4^2} = 5$. (a) The SVM margin runs between the planes $\boldsymbol{w}^\top \boldsymbol{x} + b = \pm 1$, and the distance between two parallel planes $\boldsymbol{w}^\top \boldsymbol{x} = c_1$ and $\boldsymbol{w}^\top \boldsymbol{x} = c_2$ is $|c_1 - c_2|/\|\boldsymbol{w}\|$. Here that gap is

$$\text{margin width} = \frac{2}{\|\boldsymbol{w}\|} = \frac{2}{5} = 0.4,$$

which is exactly the quantity the SVM maximizes when it minimizes $\frac{1}{2}\|\boldsymbol{w}\|^2$. (b) The distance from a point $\boldsymbol{x}_0$ to the plane $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$ is $|\boldsymbol{w}^\top \boldsymbol{x}_0 + b|/\|\boldsymbol{w}\|$; at the origin $\boldsymbol{x}_0 = \mathbf{0}$ this is

$$\frac{|b|}{\|\boldsymbol{w}\|} = \frac{1}{5} = 0.2.$$

So the boundary passes 0.2 units from the origin, and the margin extends 0.2 further on either side. Shrinking $\|\boldsymbol{w}\|$ widens the margin, which is the geometric content of the SVM objective: a smaller weight vector buys a fatter safety corridor around the boundary. □

CHAPTER 11

# Unsupervised Learning: Clustering, GMMs, PCA, and ICA

*Even with no labels at all, the data has a shape. Unsupervised learning is the search for that shape.*

*the goal of learning without labels*

**Roadmap**

The classifiers of Chapter 10 needed labeled data. This final chapter asks what we can learn from $\{\boldsymbol{x}_i\}$ alone. Four objectives organize the field: *clustering* (group similar points), *density estimation* (model where the data lives), *dimensionality reduction* (find the few directions that matter), and *source separation* (unmix combined signals). We develop $k$-means and Lloyd's algorithm, Gaussian mixtures trained by expectation–maximization, principal component analysis from both the variance-maximization and SVD viewpoints (the SVD of Chapter 4 and the spectral theorem of Chapter 3 returning in force), and independent component analysis. Every method is carried out on a small dataset with all the arithmetic shown. A brief look at reinforcement learning closes the course.

**Suggested reading:** Deisenroth et al. [5], Chapters 10 and 11; Bishop [2], Chapter 9 for EM and Chapter 12 for PCA; Hyvärinen and Oja [13] for ICA.

## 11.1 $k$-means clustering

Clustering partitions $m$ points into $k$ groups so that points in a group are close to its center. The $k$-means objective is the total *within-cluster distortion*, minimized over both the centroids $\boldsymbol{\mu}_j$ and the hard assignments $z_{ij} \in \{0,1\}$ ($\sum_j z_{ij} = 1$, each point in one cluster):

$$\min_{\{\boldsymbol{\mu}_j\},\{z_{ij}\}} \sum_{i=1}^{m}\sum_{j=1}^{k} z_{ij}\,\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|_2^2. \tag{11.1}$$

The objective is hard to minimize jointly (the $z_{ij}$ are discrete), but it is easy in each block of variables with the other fixed, which suggests alternating minimization.

**Proposition 11.1 (The two optimal steps).** *(Centroids.)* With assignments fixed, the best centroid for cluster $j$ is the mean of its members. Writing $C_j = \{i : z_{ij} = 1\}$ and $n_j = |C_j|$, set the gradient of $\sum_{i\in C_j}\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2$ to zero:

$$\nabla_{\boldsymbol{\mu}_j}\sum_{i\in C_j}\|\boldsymbol{x}_i-\boldsymbol{\mu}_j\|^2 = \sum_{i\in C_j} 2(\boldsymbol{\mu}_j-\boldsymbol{x}_i) = 2\Big(n_j\boldsymbol{\mu}_j - \sum_{i\in C_j}\boldsymbol{x}_i\Big) = \boldsymbol{0} \;\Rightarrow\; \boldsymbol{\mu}_j^{\star} = \frac{1}{n_j}\sum_{i\in C_j}\boldsymbol{x}_i.$$

*(Assignments.)* With centroids fixed, the best assignment sends each point to its nearest centroid, $z_{ij}^{\star} = 1$ iff $j = \arg\min_{\ell}\|\boldsymbol{x}_i - \boldsymbol{\mu}_\ell\|^2$, because the objective separates over points and the $k$ terms for a fixed $i$ are nonnegative.

*Lloyd's algorithm* alternates these two steps: assign every point to the nearest centroid, recompute each centroid as the mean of its members, and repeat. Each step can only decrease Eq. (11.1), and there are at most $k^m$ distinct

| Objective | Question | Method | Output |
|---|---|---|---|
| Clustering | who is similar to whom? | $k$-means, GMM | group label per point |
| Density estimation | where do points occur? | GMM | a density $p(\boldsymbol{x})$ |
| Dimensionality reduction | what is the essence? | PCA | low-dim coordinates |
| Source separation | what are the hidden causes? | ICA | independent signals |

**Table 11.1.** The four unsupervised objectives. Each takes the same unlabeled data $\{\boldsymbol{x}_i\}_{i=1}^m$ and optimizes a different criterion.

assignments, so the algorithm reaches a fixed point in finite time. But Eq. (11.1) is *nonconvex*, so the fixed point is only a *local* minimum, and the result depends on the initialization.

### 11.1.1 A worked $k$-means run, every iteration

**Example 11.2 (Lloyd's algorithm on six points).** Take the six points

$$\boldsymbol{x}_1 = (1,1),\ \boldsymbol{x}_2 = (2,1),\ \boldsymbol{x}_3 = (1,2),\quad \boldsymbol{x}_4 = (5,5),\ \boldsymbol{x}_5 = (6,5),\ \boldsymbol{x}_6 = (5,6),$$

two visually obvious clusters, and deliberately seed both centroids inside the lower-left group to test whether Lloyd's algorithm recovers: $\boldsymbol{\mu}_1^{(0)} = (1,1)$, $\boldsymbol{\mu}_2^{(0)} = (1,2)$.

**Iteration 0, assignment.** Compare each point's squared distance to the two centroids. For $\boldsymbol{x}_3 = (1,2)$: $\|\boldsymbol{x}_3 - \boldsymbol{\mu}_1\|^2 = 0 + 1 = 1$ and $\|\boldsymbol{x}_3 - \boldsymbol{\mu}_2\|^2 = 0$, so $\boldsymbol{x}_3 \to$ cluster 2. For $\boldsymbol{x}_4 = (5,5)$: distances $16 + 16 = 32$ to $\boldsymbol{\mu}_1$ and $16 + 9 = 25$ to $\boldsymbol{\mu}_2$, so $\boldsymbol{x}_4 \to$ cluster 2; likewise $\boldsymbol{x}_5, \boldsymbol{x}_6 \to$ cluster 2. Points $\boldsymbol{x}_1, \boldsymbol{x}_2 \to$ cluster 1. The assignment is

$$C_1 = \{\boldsymbol{x}_1, \boldsymbol{x}_2\}, \qquad C_2 = \{\boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6\}.$$

**Iteration 0, update.**

$$\boldsymbol{\mu}_1^{(1)} = \tfrac{1}{2}\big[(1,1) + (2,1)\big] = (1.5,\ 1), \qquad \boldsymbol{\mu}_2^{(1)} = \tfrac{1}{4}\big[(1,2) + (5,5) + (6,5) + (5,6)\big] = (4.25,\ 4.5).$$

**Iteration 1, assignment.** Now $\boldsymbol{x}_3 = (1,2)$ has $\|\boldsymbol{x}_3 - \boldsymbol{\mu}_1^{(1)}\|^2 = 0.25 + 1 = 1.25$ versus $\|\boldsymbol{x}_3 - \boldsymbol{\mu}_2^{(1)}\|^2 = 10.56 + 6.25 = 16.81$, so $\boldsymbol{x}_3$ jumps back to cluster 1. The other points keep their clusters, giving the natural split

$$C_1 = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, \qquad C_2 = \{\boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6\}.$$

**Iteration 1, update.**

$$\boldsymbol{\mu}_1^{(2)} = \tfrac{1}{3}\big[(1,1) + (2,1) + (1,2)\big] = (\tfrac{4}{3}, \tfrac{4}{3}), \qquad \boldsymbol{\mu}_2^{(2)} = \tfrac{1}{3}\big[(5,5) + (6,5) + (5,6)\big] = (\tfrac{16}{3}, \tfrac{16}{3}).$$

**Iteration 2.** Re-running the assignment leaves the clusters unchanged, so the centroids no longer move: the algorithm has **converged** in two reassignment rounds (Fig. 11.1). The final within-cluster distortion is

$$J = \sum_{i \in C_1} \|\boldsymbol{x}_i - \boldsymbol{\mu}_1\|^2 + \sum_{i \in C_2} \|\boldsymbol{x}_i - \boldsymbol{\mu}_2\|^2 = \tfrac{8}{3} \approx 2.67,$$

each cluster being three points at squared distances $\tfrac{2}{9}, \tfrac{5}{9}, \tfrac{5}{9}$ summing to $\tfrac{4}{3}$. A poor initialization was corrected because the badly placed point $(1,2)$ sat closer to the lower-left mean once the second centroid was pulled toward the upper-right group.

### 11.1.2 Why it works, and what it sees

The objective is exactly the *within-cluster scatter* $S_W$. Comparing it to the total scatter about the global mean $\bar{\boldsymbol{x}}$ gives a decomposition that explains what $k$-means optimizes. With $S_T = \sum_i \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}\|^2$ the total scatter and $S_B = \sum_j n_j \|\boldsymbol{\mu}_j - \bar{\boldsymbol{x}}\|^2$ the *between-cluster* scatter,

$$S_T = S_W + S_B. \tag{11.2}$$

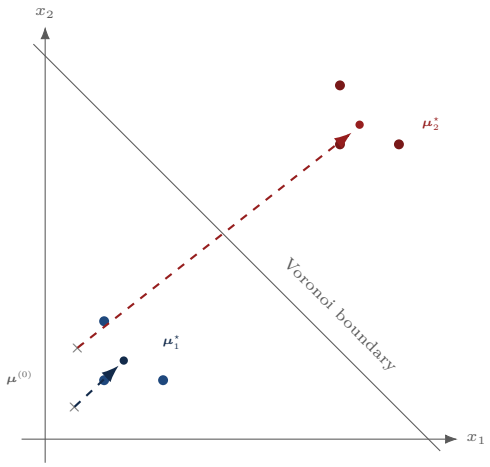



**Figure 11.1.** The run of Example 11.2. Both centroids start in the lower-left group (×); the dashed paths trace how the update step pulls $\boldsymbol{\mu}_2$ toward the upper-right group over two iterations, after which the perpendicular-bisector (Voronoi) boundary separates the clusters cleanly. Blue and red show the final assignment.

Since $S_T$ is fixed by the data, minimizing the within-cluster scatter $S_W$ is the same as *maximizing* the between-cluster separation $S_B$: $k$-means seeks clusters that are compact inside and far apart outside. The squared-Euclidean metric makes the cells convex polytopes (Voronoi regions) and biases the method toward round, equal-size clusters; crescent or nested shapes defeat it, which is where the Gaussian mixture of Section 11.2 and other models take over.

**$k$-means in practice**

- **Standardize** features first, so that a coordinate measured in centimeters does not dominate one measured in kilograms (distances scale with units).
- **Seed well.** Random seeds give erratic local minima. *$k$-means++* picks the first centroid uniformly at random, then each subsequent one with probability proportional to its squared distance from the nearest chosen centroid, spreading the seeds out and giving an $\mathcal{O}(\log k)$ expected-approximation guarantee.
- **Choose** $k$ by the *elbow* in $J(k)$ (the distortion as a function of $k$) or the *silhouette* score; there is no single correct $k$.
- **Restart** from several seeds and keep the lowest-distortion run.

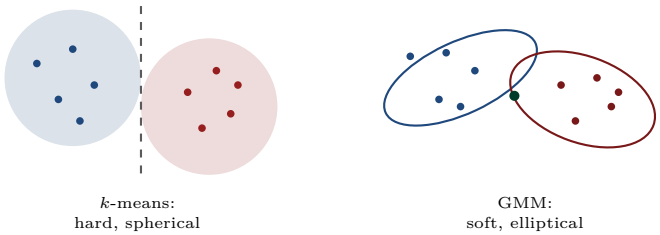



**Figure 11.2.** $k$-means versus the Gaussian mixture. $k$-means draws a hard boundary and assumes round, equal-size clusters; the GMM learns each cluster's elliptical shape through $\boldsymbol{\Sigma}_j$ and assigns *soft* memberships, so a point in the overlap (green) belongs partly to each. $k$-means is the GMM's zero-variance, hard-assignment limit.

## 11.2 Gaussian mixture models and EM

$k$-means makes hard, all-or-nothing assignments and learns only cluster centers, not shapes. The *Gaussian mixture model* (GMM) fixes both by modeling the data as drawn from a weighted blend of Gaussians,

$$p(\boldsymbol{x}) = \sum_{j=1}^{k} \pi_j \, \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \qquad \pi_j \geq 0, \ \sum_j \pi_j = 1, \tag{11.3}$$

with mixing weights $\pi_j$, means $\boldsymbol{\mu}_j$, and covariances $\boldsymbol{\Sigma}_j$. The covariances let each cluster be an ellipsoid (Section 6.4), and the model returns *soft* memberships (Fig. 11.2).

Fitting the mixture by maximum likelihood is hard directly because of the latent "which component generated $\boldsymbol{x}_i$," so we use *expectation–maximization*.

**Expectation–maximization for a Gaussian mixture**

Alternate until the log-likelihood converges:

- **E-step.** Compute the *responsibility* of component $j$ for point $i$, the posterior probability it generated $\boldsymbol{x}_i$:

$$\gamma_{ij} = \frac{\pi_j \, \mathcal{N}(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{\ell} \pi_\ell \, \mathcal{N}(\boldsymbol{x}_i \mid \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)}.$$

- **M-step.** Re-estimate each component as a responsibility-weighted fit, with $N_j = \sum_i \gamma_{ij}$:

$$\pi_j = \frac{N_j}{m}, \quad \boldsymbol{\mu}_j = \frac{1}{N_j} \sum_i \gamma_{ij} \boldsymbol{x}_i, \quad \boldsymbol{\Sigma}_j = \frac{1}{N_j} \sum_i \gamma_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\top.$$

The E-step is a Bayes' theorem computation (Chapter 5); the M-step is a weighted version of the Gaussian MLE (Section 6.4.1).

*Intuition.* EM solves a chicken-and-egg problem. If we knew which component generated each point, fitting the Gaussians would be easy (just the MLE per group); if we knew the Gaussians, assigning points would be easy (just Bayes' rule). We know neither, so EM alternates: guess the assignments (E-step), refit the Gaussians (M-step), repeat. Formally, each iteration maximizes a lower bound on the log-likelihood that touches it at the current parameters, so the true log-likelihood can only rise; EM therefore climbs to a local optimum [6]. The price, as with $k$-means, is sensitivity to initialization.

**Example 11.3 (A soft assignment in two dimensions).** Two equal-weight spherical Gaussians have $\boldsymbol{\mu}_1 = (0, 0)$ and $\boldsymbol{\mu}_2 = (3, 0)$, each with covariance $I$. What responsibility does component 1 take for the point $\boldsymbol{x} = (1, 0)$, which sits a third of the way across? With equal weights and equal covariances the normalizers cancel, leaving

$$\gamma_1(\boldsymbol{x}) = \frac{\exp(-\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{\mu}_1\|^2)}{\exp(-\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{\mu}_1\|^2) + \exp(-\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{\mu}_2\|^2)} = \frac{e^{-1/2}}{e^{-1/2} + e^{-2}} = \frac{0.607}{0.607 + 0.135} \approx 0.82,$$

since $\|\boldsymbol{x} - \boldsymbol{\mu}_1\|^2 = 1$ and $\|\boldsymbol{x} - \boldsymbol{\mu}_2\|^2 = 4$. So $\boldsymbol{x}$ is about 82% owned by the nearer cluster and 18% by the farther one, a graded membership. $k$-means would instead assign it 100% to cluster 1; the GMM's softness is what lets points near a boundary hedge, and it is exactly these fractional $\gamma_{ij}$ that the M-step uses as weights.

### 11.2.1 A worked EM iteration, every responsibility

**Example 11.4 (One EM iteration on the line).** Take six one-dimensional points $\{-1, 0, 1, 3, 4, 5\}$ and a two-component mixture with equal weights $\pi_1 = \pi_2 = \frac{1}{2}$, unit variances $\sigma_1^2 = \sigma_2^2 = 1$, and initial means $\mu_1 = 0$, $\mu_2 = 4$.

**E-step.** The responsibility of component 1 for a point $x$ is

$$\gamma_1(x) = \frac{\frac{1}{2}\mathcal{N}(x \mid 0, 1)}{\frac{1}{2}\mathcal{N}(x \mid 0, 1) + \frac{1}{2}\mathcal{N}(x \mid 4, 1)} = \frac{e^{-x^2/2}}{e^{-x^2/2} + e^{-(x-4)^2/2}},$$

the normalizing $\frac{1}{\sqrt{2\pi}}$ and the equal weights cancelling. For $x = 1$, the two exponents are $e^{-1/2} = 0.6065$ and $e^{-9/2} = 0.0111$, so $\gamma_1(1) = \frac{0.6065}{0.6065+0.0111} = 0.982$. Computing all six:

| $x$ | $-1$ | $0$ | $1$ | $3$ | $4$ | $5$ |
|---|---|---|---|---|---|---|
| $\gamma_1(x)$ | 1.000 | 0.9997 | 0.982 | 0.018 | 0.0003 | 0.000 |
| $\gamma_2(x)$ | 0.000 | 0.0003 | 0.018 | 0.982 | 0.9997 | 1.000 |

Points far from the midpoint $x = 2$ are assigned almost surely (responsibilities near 0 or 1); points near it, like $x = 1$ and $x = 3$, are *soft*, split a little between components. This softness is exactly what $k$-means throws away.

**M-step.** The effective counts are $N_1 = \sum_i \gamma_1(x_i) = 3.0$ and $N_2 = 3.0$ (by the symmetry of the data about $x = 2$), so the weights stay $\pi_1 = \pi_2 = \frac{1}{2}$. The new means are responsibility-weighted averages,

$$\mu_1 = \frac{1}{N_1}\sum_i \gamma_1(x_i)\, x_i = 0.0125, \qquad \mu_2 = \frac{1}{N_2}\sum_i \gamma_2(x_i)\, x_i = 3.9875,$$

barely moved because the assignment was already nearly hard. The new variances,

$$\sigma_1^2 = \frac{1}{N_1}\sum_i \gamma_1(x_i)(x_i - \mu_1)^2 = 0.716, \qquad \sigma_2^2 = 0.716,$$

*shrink* from 1 toward the true spread of each tight group of three. One full EM iteration has sharpened the model. Iterating further drives the variances down and the responsibilities fully hard, and the log-likelihood increases at every step.

## 11.3 Principal component analysis

Dimensionality reduction seeks a few directions that capture most of the data's variation. Center the data ($\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}$) and ask for the unit direction $\boldsymbol{u}$ along which the projections $\boldsymbol{u}^\top \tilde{\boldsymbol{x}}_i$ have the largest variance:

$$\max_{\|\boldsymbol{u}\|=1} \frac{1}{m}\sum_{i=1}^{m} (\boldsymbol{u}^\top \tilde{\boldsymbol{x}}_i)^2 = \max_{\|\boldsymbol{u}\|=1} \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u}, \qquad \boldsymbol{\Sigma} = \frac{1}{m}\sum_i \tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^\top, \tag{11.4}$$

where $\boldsymbol{\Sigma}$ is the empirical covariance matrix and we used the projection identity $(\boldsymbol{u}^\top \tilde{\boldsymbol{x}}_i)^2 = \boldsymbol{u}^\top (\tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^\top)\boldsymbol{u}$. The quantity $\boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u}$ is the *Rayleigh quotient* (Section 3.8), and maximizing it over unit vectors is an eigenvalue problem.

**Theorem 11.5 (Principal directions).** Let $\boldsymbol{\Sigma} = Q \Lambda Q^\top$ be the spectral decomposition with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$ and orthonormal eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$. The variance-maximizing direction is $\boldsymbol{u}^\star = \boldsymbol{v}_1$, and the maximum variance is $\lambda_1$. The top $k$ eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ are the $k$ *principal components*.

*Proof.* Write $\boldsymbol{q} = Q^\top \boldsymbol{u}$; since $Q$ is orthogonal, $\|\boldsymbol{q}\| = \|\boldsymbol{u}\| = 1$. Then

$$\boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} = \boldsymbol{u}^\top Q \Lambda Q^\top \boldsymbol{u} = \boldsymbol{q}^\top \Lambda \boldsymbol{q} = \sum_{i=1}^{n} \lambda_i q_i^2.$$

This is a weighted average of the eigenvalues with weights $q_i^2 \geq 0$ summing to 1, so it is largest when all the weight sits on the biggest eigenvalue: $\boldsymbol{q} = \boldsymbol{e}_1$, i.e. $\boldsymbol{u} = Q\boldsymbol{e}_1 = \boldsymbol{v}_1$, with value $\lambda_1$. The Lagrangian route agrees: stationarity of $\boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} - \alpha(\boldsymbol{u}^\top \boldsymbol{u} - 1)$ is $\boldsymbol{\Sigma}\boldsymbol{u} = \alpha \boldsymbol{u}$, so stationary points are eigenvectors and the objective there equals the eigenvalue $\alpha$; the largest wins. ∎

### 11.3.1 A worked PCA, from data to components

**Example 11.6 (PCA on four points, every step).** Take four already-centered points in $\mathbb{R}^2$ (their mean is $\mathbf{0}$, as one checks by summing):

$$\tilde{\boldsymbol{x}}_1 = (2,1), \quad \tilde{\boldsymbol{x}}_2 = (1,2), \quad \tilde{\boldsymbol{x}}_3 = (-1,-2), \quad \tilde{\boldsymbol{x}}_4 = (-2,-1).$$

**Step 1: covariance matrix.** With $m = 4$,

$$\boldsymbol{\Sigma} = \frac{1}{4}\sum_{i=1}^{4} \tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^\top = \frac{1}{4}\left( \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} + \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} \right) = \frac{1}{4}\begin{bmatrix} 10 & 8 \\ 8 & 10 \end{bmatrix} = \begin{bmatrix} 2.5 & 2 \\ 2 & 2.5 \end{bmatrix}.$$

**Step 2: eigenvalues.** The characteristic polynomial is $(2.5-\lambda)^2 - 2^2 = 0$, so $2.5 - \lambda = \pm 2$, giving $\lambda_1 = 4.5$ and $\lambda_2 = 0.5$. **Step 3: eigenvectors.** For $\lambda_1 = 4.5$, $(\boldsymbol{\Sigma} - 4.5I)\boldsymbol{v} = \mathbf{0}$ is $\left[\begin{smallmatrix} -2 & 2 \\ 2 & -2 \end{smallmatrix}\right]\boldsymbol{v} = \mathbf{0}$, i.e. $v_1 = v_2$, so $\boldsymbol{v}_1 = \frac{1}{\sqrt{2}}(1,1)$. For $\lambda_2 = 0.5$, $v_1 = -v_2$, so $\boldsymbol{v}_2 = \frac{1}{\sqrt{2}}(-1,1)$. They are orthogonal, as the spectral theorem promises. **Step 4: explained variance.** The first component carries

$$\mathrm{EVR}(1) = \frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{4.5}{5} = 90\%$$

of the total variance, so one coordinate keeps nine tenths of the spread. **Step 5: project.** The first principal coordinate of each point is $y_i = \boldsymbol{v}_1^\top \tilde{\boldsymbol{x}}_i$:

$$y_1 = \tfrac{2+1}{\sqrt{2}} = \tfrac{3}{\sqrt{2}}, \quad y_2 = \tfrac{3}{\sqrt{2}}, \quad y_3 = -\tfrac{3}{\sqrt{2}}, \quad y_4 = -\tfrac{3}{\sqrt{2}}.$$

The four points collapse onto the 45° line $\mathrm{span}\{\boldsymbol{v}_1\}$ with almost no loss (Fig. 11.3), since they lay nearly along it to begin with. The reconstruction $\hat{\boldsymbol{x}}_i = \bar{\boldsymbol{x}} + y_i \boldsymbol{v}_1$ recovers each point up to the small $\boldsymbol{v}_2$-component discarded.

**Example 11.7 (Reconstruction error equals the discarded eigenvalue).** Keep the centered data of the previous example, with $\boldsymbol{v}_1 = \frac{1}{\sqrt{2}}(1,1)$, $\boldsymbol{v}_2 = \frac{1}{\sqrt{2}}(-1,1)$, and eigenvalues $\lambda_1 = 4.5$, $\lambda_2 = 0.5$. Take the single point $\boldsymbol{x} = (2,1)$ and compress it to one number by projecting onto the top component:

$$y_1 = \boldsymbol{v}_1^\top \boldsymbol{x} = \tfrac{1}{\sqrt{2}}(2+1) = \tfrac{3}{\sqrt{2}}.$$

Reconstruct from that one coordinate, $\hat{\boldsymbol{x}} = y_1 \boldsymbol{v}_1 = \frac{3}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}}(1,1) = \frac{3}{2}(1,1) = (1.5,\, 1.5)$. The reconstruction misses by

$$\boldsymbol{x} - \hat{\boldsymbol{x}} = (2 - 1.5,\; 1 - 1.5) = (0.5,\; -0.5), \qquad \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 = 0.25 + 0.25 = 0.5.$$

That error is not arbitrary. The discarded second coordinate is $y_2 = \boldsymbol{v}_2^\top \boldsymbol{x} = \frac{1}{\sqrt{2}}(-2+1) = -\frac{1}{\sqrt{2}}$, and $y_2^2 = \frac{1}{2} = 0.5$ matches the squared error exactly, because the leftover vector $\boldsymbol{x} - \hat{\boldsymbol{x}}$ is precisely $y_2 \boldsymbol{v}_2$. Every point here has $y_2^2 = 0.5$, so the dataset's mean squared reconstruction error is $0.5 = \lambda_2$: the variance you throw away by dropping a component equals its eigenvalue. This is the working content of the Eckart–Young theorem for PCA, and it is why a scree plot of the eigenvalues reads directly as "error left on the table" for each truncation. Think of flattening a slightly tilted constellation of points onto a single line: the squared distance each point falls from the line is exactly the variance in the direction you collapsed.

### 11.3.2 The SVD viewpoint

There is no need to form $\boldsymbol{\Sigma}$ at all. With the centered data as columns of $\tilde{X}$ and its SVD $\tilde{X} = U\Sigma_X V^\top$ (Chapter 4),

$$\boldsymbol{\Sigma} = \frac{1}{m}\tilde{X}\tilde{X}^\top = \frac{1}{m} U \Sigma_X^2 U^\top,$$

so the *left singular vectors of the centered data are the principal components*, and the eigenvalues are $\lambda_j = \sigma_j^2/m$. PCA is the SVD of the centered data; the eigenfaces of Section 4.7 were this computation applied to images. (With points as *rows*, the principal components are instead the right singular vectors.) Computing the SVD of $\tilde{X}$ directly is more stable than forming and diagonalizing $\tilde{X}\tilde{X}^\top$, which squares the condition number (Section 2.11). Centering is essential, and when features have different units one also scales them to unit variance (*whitening*); see Jolliffe [14] for the full treatment.

How many components to keep is read from a *scree plot* of the eigenvalues (Fig. 11.4): the "elbow" where they flatten marks the point of diminishing returns, and one keeps enough to reach a target explained-variance ratio such as 95%.

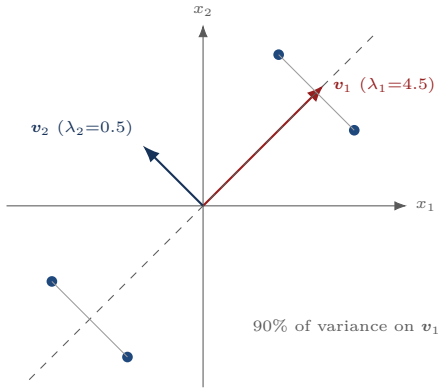



**Figure 11.3.** PCA on the four points of Example 11.6. The principal components are the orthonormal eigenvectors of the covariance, ordered by variance. The first, $\boldsymbol{v}_1$ at 45°, holds $\lambda_1/(\lambda_1+\lambda_2) = 90\%$ of the variance; projecting onto it (grey feet) loses almost nothing.

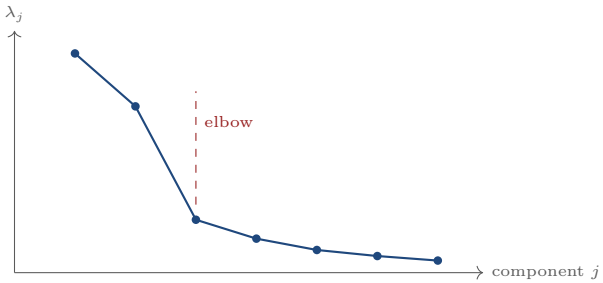



**Figure 11.4.** A scree plot. Eigenvalues drop sharply and then flatten; the elbow (here after the second component) suggests how many principal components to keep. The first two here would already capture most of the variance.

**Example 11.8 (Synthesis: correlation, a singular covariance, and a zero eigenvalue).** This one example threads together most of the book. Five students are recorded as (hours studied, exam score):

$$(2,50),\ (4,60),\ (6,70),\ (8,80),\ (10,90),$$

which happen to satisfy score $= 40 + 5\cdot$ hours exactly. The means are $\bar{h} = 6$ and $\bar{s} = 70$, so the centered data is $(-4,-20),(-2,-10),(0,0),(2,10),(4,20)$. The covariance matrix (dividing by $n = 5$, the PCA convention) is

$$\boldsymbol{\Sigma} = \frac{1}{n}\sum_i (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^\top = \begin{bmatrix} 8 & 40 \\ 40 & 200 \end{bmatrix},$$

since, for instance, $\mathrm{Var}(h) = \frac{16+4+0+4+16}{5} = 8$ and $\mathrm{Cov}(h,s) = \frac{80+20+0+20+80}{5} = 40$. The correlation coefficient is

$$\rho = \frac{40}{\sqrt{8}\sqrt{200}} = \frac{40}{\sqrt{1600}} = \frac{40}{40} = 1,$$

a perfect positive correlation, exactly as a perfectly linear relationship demands. Watch what that does to the spectrum. The determinant is $\det\boldsymbol{\Sigma} = 8(200) - 40^2 = 1600 - 1600 = 0$, so $\boldsymbol{\Sigma}$ is *singular*, and with $\operatorname{tr}\boldsymbol{\Sigma} = 208$ its eigenvalues are $\lambda_1 = 208$ and $\lambda_2 = 0$. The top eigenvector solves $(\boldsymbol{\Sigma} - 208I)\boldsymbol{v} = \boldsymbol{0}$, giving $\boldsymbol{v}_1 \propto (1,5)$, normalized $\frac{1}{\sqrt{26}}(1,5) \approx (0.196, 0.981)$, and the variance explained by the first principal component is $208/208 = 100\%$. PCA has discovered that this two-feature data is secretly one-dimensional: it lies exactly on a line, so a single coordinate reconstructs every point with zero error. The chain of equivalences is the payoff, because each link lives in a different chapter. Perfect correlation $\rho = 1$ (Chapter 6) forces a singular covariance $\det\boldsymbol{\Sigma} = 0$ (Chapter 1), which forces a zero eigenvalue (Chapter 3), which means the data has rank one and lies on a line, which is precisely what PCA reports as 100% variance on one component. The same singularity is *multicollinearity*: if these were predictors in a regression, $X^\top X$ would be singular and ordinary least squares would have no unique solution (Chapter 2), the exact failure that the ridge penalty and the pseudoinverse were built to repair. Real data never correlates this perfectly, so $\lambda_2$ is small but nonzero; the lesson is that near-collinear features produce a near-singular covariance, an unstable fit, and a scree plot that begs you to drop a component.

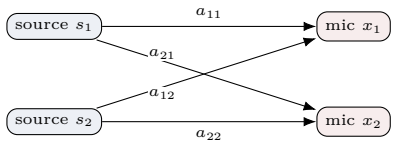



**Figure 11.5.** The cocktail-party problem. Two sources reach two microphones through an unknown mixing matrix $A$; ICA estimates an unmixing matrix $W$ that recovers the independent sources from the mixtures alone.

| | **PCA** | **ICA** |
|---|---|---|
| Finds directions that are | uncorrelated | statistically independent |
| Uses statistics of order | two (covariance) | higher (non-Gaussianity) |
| Assumes | nothing about shape | sources non-Gaussian |
| Ordered? | yes, by variance | no natural order |
| Recovers mixed sources? | no (rotation ambiguous) | yes |

**Table 11.2.** PCA versus ICA. PCA decorrelates using second-order statistics and orders directions by variance; ICA goes further, using non-Gaussianity to recover genuinely independent sources, which is what the cocktail party demands.

## 11.4 Independent component analysis

PCA finds *uncorrelated* directions; independent component analysis (ICA) finds *independent* ones, a stronger and different goal. The motivating problem is the *cocktail party*: $n$ microphones record $n$ speakers talking at once, and each recording is a linear mixture $\boldsymbol{x} = A\boldsymbol{s}$ of the unknown source signals $\boldsymbol{s}$ (Fig. 11.5). We want to recover $\boldsymbol{s}$, equivalently an unmixing matrix $W \approx A^{-1}$ with $\hat{\boldsymbol{s}} = W\boldsymbol{x}$, from $\boldsymbol{x}$ alone (*blind source separation*). ICA recovers the sources only up to scaling and permutation: we cannot know a source's loudness or its number, only its waveform.

PCA cannot do this. Decorrelation pins the sources down only up to an arbitrary rotation, because jointly Gaussian uncorrelated variables stay uncorrelated and Gaussian under any rotation, so second-order statistics carry no further information. ICA breaks the tie by assuming the sources are *non-Gaussian* and statistically independent. By the central limit theorem a mixture of independent signals looks "more Gaussian" than any single source, so recovering the sources means rotating (after whitening) to make each output as *non-Gaussian* as possible. Two contrast functions measure non-Gaussianity:

- **Kurtosis** $\kappa(y) = \mathbb{E}[y^4] - 3(\mathbb{E}[y^2])^2$, zero for a Gaussian, positive for heavy-tailed (super-Gaussian) signals, negative for light-tailed ones;
- **Negentropy** $J(y) = \mathrm{H}(y_{\text{gauss}}) - \mathrm{H}(y)$, the entropy gap to a Gaussian of equal variance, nonnegative and zero only for a Gaussian.

**The ICA pipeline**

1. **Center** the data, subtracting the mean.
2. **Whiten** with PCA: $Z = \Lambda^{-1/2} Q^\top \tilde{X}$ has identity covariance, so all that remains is to find an *orthogonal* rotation.
3. **Rotate** to maximize non-Gaussianity. *FastICA* iterates the fixed-point update, for a contrast $g$ (e.g. $g(u) = \tanh u$),
$$\boldsymbol{w} \leftarrow \mathbb{E}\big[Z\, g(\boldsymbol{w}^\top Z)\big] - \mathbb{E}\big[g'(\boldsymbol{w}^\top Z)\big]\,\boldsymbol{w}, \qquad \boldsymbol{w} \leftarrow \boldsymbol{w}/\|\boldsymbol{w}\|,$$
orthogonalizing against previously found directions when extracting several [13].

The result separates the voices that PCA could only decorrelate. The contrast between the two methods is collected in Table 11.2.

## 11.5 A glimpse of reinforcement learning

The course closes by pointing beyond supervised and unsupervised learning to *reinforcement learning*, where an agent learns by acting and receiving rewards. The setting is a *Markov decision process*: states $s$, actions $a$, transition probabilities $P(s' \mid s, a)$, rewards $R(s, a)$, and a discount $\gamma \in [0, 1)$. A *policy* $\pi$ chooses actions, and its *value function* $V^\pi(s)$ is the expected discounted return from $s$. The value satisfies the *Bellman equation*

$$V^\pi(s) = \sum_a \pi(a \mid s) \Big[ R(s, a) + \gamma \sum_{s'} P(s' \mid s, a)\, V^\pi(s') \Big], \tag{11.5}$$

a fixed-point relation that *value iteration* and *policy iteration* solve to find the optimal policy [23]. A two-state example shows the flavor: if a single action moves between states $A, B$ with rewards $R(A) = 1$, $R(B) = 0$ and deterministic transitions $A \to B \to A$, then with discount $\gamma$ the values solve $V(A) = 1 + \gamma V(B)$ and $V(B) = 0 + \gamma V(A)$, giving $V(A) = \frac{1}{1-\gamma^2}$ and $V(B) = \frac{\gamma}{1-\gamma^2}$. The same mathematics threads through: expectations and probability (Chapter 5), fixed-point iteration like the optimizers of Chapter 8, and linear algebra for the value updates. Reinforcement learning is where the foundations of this book are put to work on sequential decisions, and it is the natural next course.

## 11.6 Summary

- Unsupervised learning finds structure in unlabeled data: clustering, density estimation, dimensionality reduction, source separation (Table 11.1).
- $k$-means minimizes within-cluster distortion by alternating nearest-centroid assignment and mean-update (Lloyd's algorithm); it converges to a local minimum, so $k$-means++ seeding and restarts matter. The decomposition $S_T = S_W + S_B$ shows it simultaneously tightens clusters and separates them.
- A Gaussian mixture models the data as a blend of Gaussians and is trained by EM (responsibilities, then weighted re-estimation); the worked iteration shows responsibilities going soft near cluster boundaries. $k$-means is its zero-variance, hard-assignment limit.
- PCA's variance-maximizing directions are the top eigenvectors of the covariance matrix (Rayleigh quotient), equivalently the left singular vectors of the centered data; the eigenvalues are the variances explained, read off a scree plot.
- ICA recovers *independent*, non-Gaussian sources from linear mixtures by whitening then rotating to maximize non-Gaussianity, succeeding where PCA's mere decorrelation cannot. Reinforcement learning, built on the Bellman equation, is the natural sequel.

## Exercises

**Exercise 11.1.** Run $k$-means with $k = 2$ on the one-dimensional points $\{1, 2, 9, 10\}$, starting from centroids $\mu_1 = 0$, $\mu_2 = 8$. Carry out the assignment and update steps to convergence, and report the final clusters, centroids, and objective value.

*Solution.* **Iteration 0, assignment.** Distances to $\mu_1 = 0$: $1, 2, 9, 10$; to $\mu_2 = 8$: $7, 6, 1, 2$. Each point joins the nearer centroid: 1 and 2 are closer to 0 (distances $1, 2 < 7, 6$), while 9 and 10 are closer to 8 ($1, 2 < 9, 10$). Clusters $C_1 = \{1, 2\}$, $C_2 = \{9, 10\}$. **Iteration 0, update.** $\mu_1 = \frac{1+2}{2} = 1.5$, $\mu_2 = \frac{9+10}{2} = 9.5$. **Iteration 1, assignment.** Distances to $\mu_1 = 1.5$: $0.5, 0.5, 7.5, 8.5$; to $\mu_2 = 9.5$: $8.5, 7.5, 0.5, 0.5$. Points $1, 2$ stay with $\mu_1$; $9, 10$ stay with $\mu_2$. Assignments unchanged, so the algorithm has **converged**. **Objective.** $J = (1 - 1.5)^2 + (2 - 1.5)^2 + (9 - 9.5)^2 + (10 - 9.5)^2 = 0.25 \cdot 4 = 1$. The obvious grouping is recovered in one update. □

**Exercise 11.2.** A two-component, equal-weight, unit-variance Gaussian mixture on $\mathbb{R}$ has means $\mu_1 = 0$ and $\mu_2 = 4$. Compute the responsibility $\gamma_1(x)$ of component 1 for the points $x = 2$ and $x = 1$, and explain the difference.

*Solution.* With equal weights and equal variances the normalizers cancel, leaving $\gamma_1(x) = \dfrac{e^{-x^2/2}}{e^{-x^2/2} + e^{-(x-4)^2/2}}$. *At* $x = 2$ (the midpoint): the exponents are $-\frac{4}{2} = -2$ and $-\frac{4}{2} = -2$, equal, so $\gamma_1(2) = \frac{e^{-2}}{2e^{-2}} = \frac{1}{2}$. The point is

genuinely ambiguous, split evenly. *At* $x = 1$: the exponents are $-\frac{1}{2} = -0.5$ and $-\frac{9}{2} = -4.5$, giving $e^{-0.5} = 0.6065$ and $e^{-4.5} = 0.0111$, so $\gamma_1(1) = \frac{0.6065}{0.6065+0.0111} = 0.982$. Being closer to $\mu_1$, the point is assigned mostly to component 1. The responsibilities are *soft*: a point between the means is shared, a point near a mean is nearly certain. This graded membership is exactly what distinguishes EM from the hard assignments of $k$-means. □

**Exercise 11.3.** Centered data has empirical covariance $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$. Find the first principal component and the fraction of variance it explains.

*Solution.* By Theorem 11.5 the first principal component is the top eigenvector of $\boldsymbol{\Sigma}$. **Eigenvalues:** $(2-\lambda)^2 - 1 = 0 \Rightarrow 2 - \lambda = \pm 1 \Rightarrow \lambda_1 = 3,\ \lambda_2 = 1$. **Eigenvectors:** for $\lambda_1 = 3$, $(\boldsymbol{\Sigma} - 3I)\boldsymbol{v} = \left[\begin{smallmatrix} -1 & 1 \\ 1 & -1 \end{smallmatrix}\right]\boldsymbol{v} = \mathbf{0}$ gives $v_1 = v_2$, so $\boldsymbol{v}_1 = \frac{1}{\sqrt{2}}(1,1)$; for $\lambda_2 = 1$, $\boldsymbol{v}_2 = \frac{1}{\sqrt{2}}(1,-1)$. The first principal component is $\boldsymbol{v}_1 = \frac{1}{\sqrt{2}}(1,1)$, the $45^\circ$ direction along which the data spreads most. **Explained variance:** $\lambda_1/(\lambda_1+\lambda_2) = 3/4 = 75\%$, so the single coordinate $y_i = \boldsymbol{v}_1^\top \tilde{\boldsymbol{x}}_i$ retains three quarters of the variation. □

**Exercise 11.4.** Explain the precise sense in which $k$-means is a special case of fitting a Gaussian mixture by EM.

*Solution.* Take a Gaussian mixture in which every component has the same spherical covariance $\sigma^2 I$ and equal weights $\pi_j = 1/k$. The EM responsibility of component $j$ for point $\boldsymbol{x}_i$ is $\gamma_{ij} = \dfrac{\exp(-\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2/2\sigma^2)}{\sum_\ell \exp(-\|\boldsymbol{x}_i - \boldsymbol{\mu}_\ell\|^2/2\sigma^2)}$, a softmax over the negative squared distances. As $\sigma^2 \to 0$ the term with the smallest distance dominates the denominator, so the softmax sharpens to a hard $\arg\min$: $\gamma_{ij} \to z_{ij} \in \{0,1\}$ with all the weight on the nearest centroid. The E-step becomes the $k$-means assignment step, and the M-step mean update $\boldsymbol{\mu}_j = \frac{1}{N_j}\sum_i \gamma_{ij}\boldsymbol{x}_i$ becomes the cluster-mean update of Proposition 11.1. So $k$-means is the zero-variance, hard-assignment limit of EM for an equal-weight spherical GMM. The general GMM is strictly more flexible: it keeps soft responsibilities and learns each cluster's shape through $\boldsymbol{\Sigma}_j$. □

**Exercise 11.5.** Two independent audio sources are linearly mixed into two microphone recordings. Explain why PCA cannot recover the original sources but ICA can, and what assumption makes the difference.

*Solution.* PCA finds orthogonal directions of maximal variance, which makes the recovered signals *uncorrelated* but determines them only up to an arbitrary rotation: decorrelation alone does not pick out the true mixing directions. If the sources were Gaussian the problem would be genuinely unidentifiable, since any rotation of jointly Gaussian uncorrelated variables is again uncorrelated and Gaussian, so no amount of second-order (variance/covariance) information can separate them. ICA adds the assumption that the sources are *non-Gaussian* and statistically *independent*. By the central limit theorem a mixture of independent sources is more Gaussian than any single source, so ICA recovers the sources by rotating (after whitening) to make the outputs maximally non-Gaussian, for example maximizing kurtosis or negentropy as in FastICA. Independence plus non-Gaussianity is exactly the information PCA lacks. □

**Exercise 11.6.** Show that the principal components of a centered data matrix can be obtained from its singular value decomposition without ever forming the covariance matrix, and relate the singular values to the variances.

*Solution.* Let the centered data be the columns of $\tilde{X} \in \mathbb{R}^{n \times m}$ with SVD $\tilde{X} = U\Sigma_X V^\top$. The covariance is

$$\boldsymbol{\Sigma} = \frac{1}{m}\tilde{X}\tilde{X}^\top = \frac{1}{m}U\Sigma_X V^\top V \Sigma_X U^\top = \frac{1}{m}U\Sigma_X^2 U^\top,$$

using $V^\top V = I$. This is already an eigendecomposition of $\boldsymbol{\Sigma}$ with orthonormal eigenvectors $U$ and eigenvalues $\sigma_j^2/m$. Hence the principal components are the left singular vectors (columns of $U$) of the centered data, and the variance explained by component $j$ is $\lambda_j = \sigma_j^2/m$. Computing the SVD of $\tilde{X}$ directly is more numerically stable than forming and diagonalizing $\tilde{X}\tilde{X}^\top$, for the conditioning reason noted in Section 2.11 (forming $\tilde{X}\tilde{X}^\top$ squares the singular values and the condition number). □

**Exercise 11.7.** A dataset's covariance matrix has eigenvalues $6, 3, 1$. (a) What fraction of the total variance is captured by the first two principal components? (b) If you keep only the first component, what is the mean squared reconstruction error?

*Solution.* The eigenvalues of the covariance are the variances along the principal directions, and they sum to the total variance $6 + 3 + 1 = 10$. (a) The top two components carry $\dfrac{6+3}{10} = 0.9$, so 90% of the variance lives in a two-dimensional subspace, and projecting there loses only a tenth of the spread. (b) By the reconstruction identity (Example 11.7), dropping components costs exactly their eigenvalues: keeping only the first discards the second and third, so the mean squared reconstruction error is $\lambda_2 + \lambda_3 = 3 + 1 = 4$. Reading these numbers off a scree plot is how one chooses where to truncate: here the jump from 90% (two components) is the natural stopping point, since the third eigenvalue 1 adds little. □

PART VII

# Problem Sets, Worked in Full

CHAPTER 12

# Problem Set 1: Linear Algebra and Matrix Methods

This first set drills the mechanics of Chapters 1 and 2: multiplying matrices, testing independence, solving systems by elimination, inverting by Gauss–Jordan, taking determinants by hand, reading a matrix as a geometric map, and fitting by least squares through the normal equations and the pseudoinverse. Every problem is restated in full and solved one arithmetic step at a time, so nothing is left to "it follows that." Check each answer against the definitions as you go; the habit of verifying ($A^{\top}\boldsymbol{r} = \mathbf{0}$, $\det \neq 0$, a vector that re-expands correctly) is the whole point.

## 12.1 Matrix and vector operations

### Problem 1.1: Matrix products and the transpose rule

**Problem.** Let $A = \left[\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix}\right]$ and $B = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$. (a) Compute $AB$ and $BA$ and confirm that matrix multiplication does not commute. (b) Verify the transpose rule $(AB)^{\top} = B^{\top}A^{\top}$.

**Solution.** Each entry of a product is a row of the left matrix dotted with a column of the right matrix: $(AB)_{ij} = \sum_k A_{ik}B_{kj}$.

*Part (a).* Compute $AB$ entry by entry. The matrix $B$ swaps the two columns of whatever it multiplies on the right, but we compute it honestly:

$$\begin{aligned} (AB)_{11} &= 1\cdot 0 + 2\cdot 1 = 2, \quad (AB)_{12} = 1\cdot 1 + 2\cdot 0 = 1, \\ (AB)_{21} &= 3\cdot 0 + 4\cdot 1 = 4, \quad (AB)_{22} = 3\cdot 1 + 4\cdot 0 = 3, \end{aligned} \qquad \Longrightarrow \qquad AB = \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix}.$$

Now $BA$, where $B$ on the left swaps the two rows of $A$:

$$\begin{aligned} (BA)_{11} &= 0\cdot 1 + 1\cdot 3 = 3, \quad (BA)_{12} = 0\cdot 2 + 1\cdot 4 = 4, \\ (BA)_{21} &= 1\cdot 1 + 0\cdot 3 = 1, \quad (BA)_{22} = 1\cdot 2 + 0\cdot 4 = 2, \end{aligned} \qquad \Longrightarrow \qquad BA = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix}.$$

Since $AB = \left[\begin{smallmatrix} 2 & 1 \\ 4 & 3 \end{smallmatrix}\right] \neq \left[\begin{smallmatrix} 3 & 4 \\ 1 & 2 \end{smallmatrix}\right] = BA$, the product does not commute. This is the norm rather than the exception: the order in which linear maps are applied matters, exactly as rotating-then-shearing differs from shearing-then-rotating.

*Part (b).* Transposing $AB$ reflects it across the diagonal:

$$(AB)^{\top} = \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix}^{\top} = \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix}.$$

Now build $B^{\top}A^{\top}$. Here $A^{\top} = \left[\begin{smallmatrix} 1 & 3 \\ 2 & 4 \end{smallmatrix}\right]$ and $B^{\top} = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$ ($B$ is symmetric, so $B^{\top} = B$). Then

$$B^{\top}A^{\top} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 0\cdot 1 + 1\cdot 2 & 0\cdot 3 + 1\cdot 4 \\ 1\cdot 1 + 0\cdot 2 & 1\cdot 3 + 0\cdot 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix}.$$

This equals $(AB)^{\top}$, confirming the rule. Note the *reversal* of order: the transpose of a product flips the factors, because the inner dimensions must still match after every matrix is flipped. The same reversal governs the inverse, $(AB)^{-1} = B^{-1}A^{-1}$.

## Problem 1.2: Independence, rank, and an explicit dependence

**Problem.** Decide whether the vectors $\boldsymbol{v}_1 = (1, 2, 1)$, $\boldsymbol{v}_2 = (2, 1, 0)$, $\boldsymbol{v}_3 = (3, 3, 1)$ in $\mathbb{R}^3$ are linearly independent. If they are not, exhibit an explicit linear dependence, and state the rank and a basis of their span.

**Solution.** Independence asks whether $c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + c_3\boldsymbol{v}_3 = \mathbf{0}$ forces $c_1 = c_2 = c_3 = 0$. Stack the vectors as the rows of a matrix and row-reduce; the number of nonzero rows that survive is the rank, and any shortfall from 3 signals a dependence.

$$M = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 0 \\ 3 & 3 & 1 \end{bmatrix} \xrightarrow{R_2 \leftarrow R_2 - 2R_1} \begin{bmatrix} 1 & 2 & 1 \\ 0 & -3 & -2 \\ 3 & 3 & 1 \end{bmatrix} \xrightarrow{R_3 \leftarrow R_3 - 3R_1} \begin{bmatrix} 1 & 2 & 1 \\ 0 & -3 & -2 \\ 0 & -3 & -2 \end{bmatrix}.$$

The last step subtracts $3R_1$ from $R_3$, giving $(3{-}3,\ 3{-}6,\ 1{-}3) = (0, -3, -2)$, which is *identical* to the new $R_2$. One more elimination makes the dependence explicit:

$$\xrightarrow{R_3 \leftarrow R_3 - R_2} \begin{bmatrix} 1 & 2 & 1 \\ 0 & -3 & -2 \\ 0 & 0 & 0 \end{bmatrix}.$$

A zero row appeared, so the rank is 2 and the three vectors are *dependent*. To name the dependence, notice the third row became zero by subtracting the first two, which traces back to $\boldsymbol{v}_3 = \boldsymbol{v}_1 + \boldsymbol{v}_2$. Check directly:

$$\boldsymbol{v}_1 + \boldsymbol{v}_2 = (1+2,\ 2+1,\ 1+0) = (3, 3, 1) = \boldsymbol{v}_3. \checkmark$$

Equivalently $\boldsymbol{v}_1 + \boldsymbol{v}_2 - \boldsymbol{v}_3 = \mathbf{0}$, a nontrivial dependence. The rank is 2, and since $\boldsymbol{v}_1, \boldsymbol{v}_2$ are independent (neither is a multiple of the other), a basis for the span is $\{\boldsymbol{v}_1, \boldsymbol{v}_2\}$; the span is a plane through the origin in $\mathbb{R}^3$, and $\boldsymbol{v}_3$ already lies in it. A determinant check agrees: $\det M = 0$, the unmistakable signature of dependence for a square stack.

## Problem 1.3: A $3 \times 3$ system by Gaussian elimination

**Problem.** Solve the system

$$\begin{aligned} x + y + z &= 6, \\ 2x + y - z &= 1, \\ x - y + 2z &= 5, \end{aligned}$$

by Gaussian elimination, showing every row operation, then back-substitute.

**Solution.** Work on the augmented matrix $[\,A \mid \boldsymbol{b}\,]$ and drive it to upper-triangular form, clearing the first column, then the second.

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 2 & 1 & -1 & 1 \\ 1 & -1 & 2 & 5 \end{array}\right] \xrightarrow[R_3 \leftarrow R_3 - R_1]{R_2 \leftarrow R_2 - 2R_1} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & -1 & -3 & -11 \\ 0 & -2 & 1 & -1 \end{array}\right].$$

The second row came from $(2{-}2,\ 1{-}2,\ {-}1{-}2 \mid 1{-}12) = (0, -1, -3 \mid -11)$, and the third from $(1{-}1,\ {-}1{-}1,\ 2{-}1 \mid 5{-}6) = (0, -2, 1 \mid -1)$. Now clear the second column below the pivot:

$$\xrightarrow{R_3 \leftarrow R_3 - 2R_2} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & -1 & -3 & -11 \\ 0 & 0 & 7 & 21 \end{array}\right],$$

since $(0,\ -2 - 2(-1),\ 1 - 2(-3) \mid -1 - 2(-11)) = (0, 0, 7 \mid 21)$. The matrix is now triangular, so back-substitute from the bottom:

$$7z = 21 \Rightarrow z = 3; \qquad -y - 3z = -11 \Rightarrow -y - 9 = -11 \Rightarrow y = 2; \qquad x + y + z = 6 \Rightarrow x + 5 = 6 \Rightarrow x = 1.$$

The solution is $(x, y, z) = (1, 2, 3)$. Verify in all three original equations: $1 + 2 + 3 = 6$, $2 + 2 - 3 = 1$, and $1 - 2 + 6 = 5$. All hold, so the answer is exact. The single pivot in each column (no zero pivots, nonzero 7 at the end) confirms the coefficient matrix is invertible and the solution is unique.

## Problem 1.4: A determinant by cofactor expansion

**Problem.** Compute $\det A$ for $A = \left[\begin{smallmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{smallmatrix}\right]$ by cofactor expansion along the first row, and state whether $A$ is invertible.

**Solution.** Cofactor expansion along row 1 reads $\det A = \sum_j A_{1j}(-1)^{1+j} M_{1j}$, where $M_{1j}$ is the $2 \times 2$ minor obtained by deleting row 1 and column $j$. The sign pattern on the first row is $+, -, +$.

$$\det A = 2 \underbrace{\det \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}}_{M_{11}} - 1 \underbrace{\det \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}}_{M_{12}} + 0 \underbrace{\det \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}}_{M_{13}}.$$

Evaluate the $2 \times 2$ minors with the rule $\det\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right] = ad - bc$:

$$M_{11} = 3 \cdot 2 - 1 \cdot 1 = 5, \qquad M_{12} = 1 \cdot 2 - 1 \cdot 0 = 2, \qquad M_{13} = 1 \cdot 1 - 3 \cdot 0 = 1.$$

The third term carries a factor 0, so it drops out regardless of $M_{13}$. Therefore

$$\det A = 2(5) - 1(2) + 0(1) = 10 - 2 = 8.$$

Since $\det A = 8 \neq 0$, the matrix is invertible: its columns are independent, it has full rank 3, and $A\boldsymbol{x} = \boldsymbol{b}$ has a unique solution for every $\boldsymbol{b}$. (As a sanity check, this symmetric tridiagonal matrix is positive definite, and indeed its determinant is positive.)

## Problem 1.5: Norms, angle, projection, and Cauchy–Schwarz

**Problem.** For $\boldsymbol{u} = (3, 4, 0)$ and $\boldsymbol{v} = (0, 4, 3)$, compute the lengths $\|\boldsymbol{u}\|$ and $\|\boldsymbol{v}\|$, the inner product $\boldsymbol{u}^\top \boldsymbol{v}$, the angle between them, the projection of $\boldsymbol{u}$ onto $\boldsymbol{v}$, and verify the Cauchy–Schwarz inequality.

**Solution.** Lengths come from the inner product of a vector with itself:

$$\|\boldsymbol{u}\| = \sqrt{3^2 + 4^2 + 0^2} = \sqrt{25} = 5, \qquad \|\boldsymbol{v}\| = \sqrt{0^2 + 4^2 + 3^2} = \sqrt{25} = 5.$$

The inner product is the sum of componentwise products:

$$\boldsymbol{u}^\top \boldsymbol{v} = 3 \cdot 0 + 4 \cdot 4 + 0 \cdot 3 = 16.$$

The angle follows from $\cos\vartheta = \dfrac{\boldsymbol{u}^\top \boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|}$:

$$\cos\vartheta = \frac{16}{5 \cdot 5} = \frac{16}{25} = 0.64, \qquad \vartheta = \arccos(0.64) \approx 50.2^\circ.$$

The projection of $\boldsymbol{u}$ onto the direction of $\boldsymbol{v}$ keeps only the component along $\boldsymbol{v}$:

$$\operatorname{proj}_{\boldsymbol{v}} \boldsymbol{u} = \frac{\boldsymbol{u}^\top \boldsymbol{v}}{\boldsymbol{v}^\top \boldsymbol{v}}\, \boldsymbol{v} = \frac{16}{25}(0, 4, 3) = \left(0,\ \tfrac{64}{25},\ \tfrac{48}{25}\right) = (0,\ 2.56,\ 1.92).$$

Finally, Cauchy–Schwarz states $|\boldsymbol{u}^\top \boldsymbol{v}| \le \|\boldsymbol{u}\|\|\boldsymbol{v}\|$. Here $|16| = 16 \le 25 = 5 \cdot 5$, comfortably satisfied, with equality only if the vectors were parallel. The gap ($16 < 25$) is exactly why $\cos\vartheta < 1$ and the vectors sit at a genuine angle rather than pointing the same way.

## Problem 1.6: A matrix as a map, and the determinant as area

**Problem.** The matrix $A = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$ acts on the plane by $\boldsymbol{x} \mapsto A\boldsymbol{x}$. (a) Find the images of the standard basis vectors and of the four corners of the unit square. (b) What is the area of the image of the unit square, and how does $\det A$ explain it?

**Solution.** *Part (a).* A matrix sends the standard basis to its own columns, so

$$A\boldsymbol{e}_1 = A \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad A\boldsymbol{e}_2 = A \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The unit square has corners $(0,0),(1,0),(1,1),(0,1)$. Applying $A$ (which adds the $y$-coordinate to the $x$-coordinate and leaves $y$ alone) sends them to

$$(0,0)\mapsto(0,0),\quad (1,0)\mapsto(1,0),\quad (1,1)\mapsto(2,1),\quad (0,1)\mapsto(1,1).$$

The image is a parallelogram: the bottom edge stays put while the top edge slides one unit to the right. This is a *shear*. *Part (b).* The area of the image parallelogram is the absolute value of the determinant of $A$:

$$\det A=\det\begin{bmatrix}1&1\\0&1\end{bmatrix}=1\cdot 1-1\cdot 0=1,\qquad \text{area}=|\det A|=1.$$

The sheared square has the same area as the original unit square, which matches the picture: a shear slides layers sideways without stretching them, so it preserves area. In general $|\det A|$ is the factor by which $A$ scales every area (volume in higher dimensions), and the sign of $\det A$ records whether orientation is preserved $(+)$ or flipped $(-)$. A determinant of zero would mean the square collapses to a segment of area 0, the geometric face of a singular, non-invertible map.

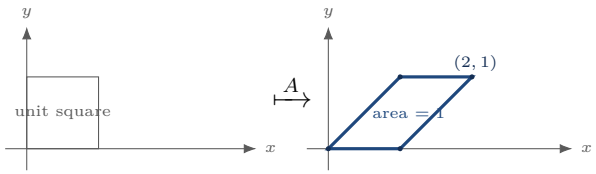



**Figure 12.1.** The shear $A$ with columns $(1,0)$ and $(1,1)$ slides the top edge of the unit square one unit to the right. The bottom stays fixed, the area is unchanged at $|\det A|=1$, and orientation is preserved.

## Problem 1.7: A $3\times 3$ inverse by Gauss–Jordan elimination

**Problem.** Find $A^{-1}$ for $A=\left[\begin{smallmatrix}1&1&0\\0&1&1\\1&0&1\end{smallmatrix}\right]$ by Gauss–Jordan elimination on the augmented matrix $[\,A\mid I\,]$.

**Solution.** Augment $A$ with the identity and apply row operations until the left block becomes $I$; whatever the right block becomes is $A^{-1}$, because the same sequence of operations is the matrix $A^{-1}$ applied to $I$.

$$\left[\begin{array}{ccc|ccc}1&1&0&1&0&0\\0&1&1&0&1&0\\1&0&1&0&0&1\end{array}\right]\xrightarrow{R_3\leftarrow R_3-R_1}\left[\begin{array}{ccc|ccc}1&1&0&1&0&0\\0&1&1&0&1&0\\0&-1&1&-1&0&1\end{array}\right].$$

Clear the second column below the pivot, then scale the third row:

$$\xrightarrow{R_3\leftarrow R_3+R_2}\left[\begin{array}{ccc|ccc}1&1&0&1&0&0\\0&1&1&0&1&0\\0&0&2&-1&1&1\end{array}\right]\xrightarrow{R_3\leftarrow \frac{1}{2}R_3}\left[\begin{array}{ccc|ccc}1&1&0&1&0&0\\0&1&1&0&1&0\\0&0&1&-\frac12&\frac12&\frac12\end{array}\right].$$

Now eliminate upward to clear the third and second columns above their pivots:

$$\xrightarrow{R_2\leftarrow R_2-R_3}\left[\begin{array}{ccc|ccc}1&1&0&1&0&0\\0&1&0&\frac12&\frac12&-\frac12\\0&0&1&-\frac12&\frac12&\frac12\end{array}\right]\xrightarrow{R_1\leftarrow R_1-R_2}\left[\begin{array}{ccc|ccc}1&0&0&\frac12&-\frac12&\frac12\\0&1&0&\frac12&\frac12&-\frac12\\0&0&1&-\frac12&\frac12&\frac12\end{array}\right].$$

The left block is the identity, so

$$A^{-1}=\frac12\begin{bmatrix}1&-1&1\\1&1&-1\\-1&1&1\end{bmatrix}.$$

A quick check confirms it: the first row of $A$ times the first column of $A^{-1}$ is $\frac12(1\cdot 1+1\cdot 1+0\cdot(-1))=\frac12(2)=1$, and the off-diagonal products cancel to 0, so $AA^{-1}=I$. The pivots were all nonzero (the last was 2), so $\det A=2$ and the inverse exists.

### Problem 1.8: Splitting a vector into parallel and perpendicular parts

**Problem.** Write $\boldsymbol{b} = (3, 1)$ as the sum of a vector parallel to $\boldsymbol{a} = (1, 1)$ and a vector perpendicular to $\boldsymbol{a}$. Verify the two parts are orthogonal and sum to $\boldsymbol{b}$.

**Solution.** The parallel part is the projection of $\boldsymbol{b}$ onto $\boldsymbol{a}$, and the perpendicular part is whatever is left over:

$$\boldsymbol{b}_{\parallel} = \frac{\boldsymbol{b}^{\top}\boldsymbol{a}}{\boldsymbol{a}^{\top}\boldsymbol{a}}\,\boldsymbol{a}, \qquad \boldsymbol{b}_{\perp} = \boldsymbol{b} - \boldsymbol{b}_{\parallel}.$$

Compute the scalars: $\boldsymbol{b}^{\top}\boldsymbol{a} = 3 \cdot 1 + 1 \cdot 1 = 4$ and $\boldsymbol{a}^{\top}\boldsymbol{a} = 1 + 1 = 2$, so the coefficient is $4/2 = 2$ and

$$\boldsymbol{b}_{\parallel} = 2(1, 1) = (2, 2), \qquad \boldsymbol{b}_{\perp} = (3, 1) - (2, 2) = (1, -1).$$

Check orthogonality, $\boldsymbol{b}_{\perp}^{\top}\boldsymbol{a} = 1 \cdot 1 + (-1) \cdot 1 = 0$, and reassembly, $\boldsymbol{b}_{\parallel} + \boldsymbol{b}_{\perp} = (2{+}1,\ 2{-}1) = (3, 1) = \boldsymbol{b}$. Both hold. This split is the engine of least squares: $\boldsymbol{b}_{\parallel}$ is the part of $\boldsymbol{b}$ that the line through $\boldsymbol{a}$ can reproduce, and $\boldsymbol{b}_{\perp}$ is the residual error, necessarily orthogonal to the fitting direction.

### Problem 1.9: A system with a free variable

**Problem.** Describe all solutions of

$$x + 2y + z = 4, \qquad 2x + 4y + 2z = 8.$$

**Solution.** Look at the two equations together. The second is exactly twice the first ($2x + 4y + 2z = 2(x + 2y + z)$ and $8 = 2 \cdot 4$), so it carries no new information. The system collapses to the single constraint $x + 2y + z = 4$ on three unknowns, leaving two degrees of freedom. Choose the free variables $y = s$ and $z = t$ and solve for the pivot variable $x$:

$$x = 4 - 2s - t.$$

The full solution set is the family

$$(x, y, z) = (4 - 2s - t,\ s,\ t) = \underbrace{(4, 0, 0)}_{\text{particular}} + s\ \underbrace{(-2, 1, 0)}_{\text{null direction}} + t\ \underbrace{(-1, 0, 1)}_{\text{null direction}}, \qquad s, t \in \mathbb{R}.$$

This is a particular solution plus the null space of the coefficient matrix, the general shape of every consistent linear system. Geometrically it is a plane of solutions in $\mathbb{R}^3$. Spot checks: $(s, t) = (0, 0)$ gives $(4, 0, 0)$ with $4 + 0 + 0 = 4$; $(1, 0)$ gives $(2, 1, 0)$ with $2 + 2 + 0 = 4$; $(0, 1)$ gives $(3, 0, 1)$ with $3 + 0 + 1 = 4$. All satisfy the equation.

### Problem 1.10: The trace is cyclic

**Problem.** For $A = \left[\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix}\right]$ and $B = \left[\begin{smallmatrix} 2 & 0 \\ 1 & 2 \end{smallmatrix}\right]$, compute $\operatorname{tr}(AB)$ and $\operatorname{tr}(BA)$ and confirm they are equal even though $AB \neq BA$.

**Solution.** First the two products:

$$AB = \begin{bmatrix} 1 \cdot 2 + 2 \cdot 1 & 1 \cdot 0 + 2 \cdot 2 \\ 3 \cdot 2 + 4 \cdot 1 & 3 \cdot 0 + 4 \cdot 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 10 & 8 \end{bmatrix}, \quad BA = \begin{bmatrix} 2 \cdot 1 + 0 \cdot 3 & 2 \cdot 2 + 0 \cdot 4 \\ 1 \cdot 1 + 2 \cdot 3 & 1 \cdot 2 + 2 \cdot 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 7 & 10 \end{bmatrix}.$$

These are different matrices, yet their diagonals sum to the same number:

$$\operatorname{tr}(AB) = 4 + 8 = 12, \qquad \operatorname{tr}(BA) = 2 + 10 = 12.$$

The equality is no accident. In general $\operatorname{tr}(AB) = \sum_i \sum_k A_{ik} B_{ki} = \sum_k \sum_i B_{ki} A_{ik} = \operatorname{tr}(BA)$, because both double sums run over the same products $A_{ik}B_{ki}$ in a different order. This *cyclic* property, $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ and more generally $\operatorname{tr}(ABC) = \operatorname{tr}(CAB)$, is what makes the trace blind to a change of basis: $\operatorname{tr}(P^{-1}MP) = \operatorname{tr}(MPP^{-1}) = \operatorname{tr}(M)$, so the trace is an intrinsic property of the linear map, equal to the sum of its eigenvalues.

### Problem 1.11: Rotations compose by adding angles

**Problem.** Let $R(\theta) = \left[\begin{smallmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{smallmatrix}\right]$. Show by direct multiplication that $R(30^\circ)R(60^\circ) = R(90^\circ)$.

**Solution.** Use $\cos 30^\circ = \frac{\sqrt{3}}{2}$, $\sin 30^\circ = \frac{1}{2}$ and $\cos 60^\circ = \frac{1}{2}$, $\sin 60^\circ = \frac{\sqrt{3}}{2}$, so

$$R(30^\circ) = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}, \qquad R(60^\circ) = \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix}.$$

The top-left entry of the product is $\frac{\sqrt{3}}{2}\cdot\frac{1}{2} + (-\frac{1}{2})\cdot\frac{\sqrt{3}}{2} = \frac{\sqrt{3}}{4} - \frac{\sqrt{3}}{4} = 0$, and the bottom-left entry is $\frac{1}{2}\cdot\frac{1}{2} + \frac{\sqrt{3}}{2}\cdot\frac{\sqrt{3}}{2} = \frac{1}{4} + \frac{3}{4} = 1$. The top-right is $\frac{\sqrt{3}}{2}(-\frac{\sqrt{3}}{2}) + (-\frac{1}{2})\frac{1}{2} = -\frac{3}{4} - \frac{1}{4} = -1$, and the bottom-right is $\frac{1}{2}(-\frac{\sqrt{3}}{2}) + \frac{\sqrt{3}}{2}\cdot\frac{1}{2} = 0$. Hence

$$R(30^\circ)R(60^\circ) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = R(90^\circ),$$

since $\cos 90^\circ = 0$ and $\sin 90^\circ = 1$. The angles added, $30^\circ + 60^\circ = 90^\circ$, exactly as the angle-addition formulas for sine and cosine guarantee. Rotations therefore form a group under multiplication, and $R(\theta)^{-1} = R(-\theta)$ undoes a rotation by turning back through the same angle.

## 12.2 Least squares and the pseudoinverse

### Problem 1.12: A least-squares line and its $R^2$

**Problem.** Fit the line $y = a + bx$ to the four points $(1,2), (2,2), (3,4), (4,5)$ by least squares, and report the coefficient of determination $R^2$.

**Solution.** Stack a column of ones (for the intercept) and the $x$-values (for the slope) into the design matrix $A$, with the targets in $\boldsymbol{y}$:

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 2 \\ 2 \\ 4 \\ 5 \end{bmatrix}, \qquad A^\top A = \begin{bmatrix} 4 & 10 \\ 10 & 30 \end{bmatrix}, \quad A^\top \boldsymbol{y} = \begin{bmatrix} 13 \\ 38 \end{bmatrix},$$

where $A^\top A$ collects $\sum 1 = 4$, $\sum x_i = 10$, $\sum x_i^2 = 30$, and $A^\top \boldsymbol{y}$ collects $\sum y_i = 13$, $\sum x_i y_i = 38$. The determinant of $A^\top A$ is $4 \cdot 30 - 10 \cdot 10 = 20$, so by Cramer's rule

$$a = \frac{13 \cdot 30 - 10 \cdot 38}{20} = \frac{390 - 380}{20} = 0.5, \qquad b = \frac{4 \cdot 38 - 10 \cdot 13}{20} = \frac{152 - 130}{20} = 1.1.$$

The fit is $\hat{y} = 0.5 + 1.1x$, with predictions $(1.6, 2.7, 3.8, 4.9)$ and residuals $\boldsymbol{r} = (0.4, -0.7, 0.2, 0.1)$. The residual sum of squares is $\mathrm{SS}_{\mathrm{res}} = 0.16 + 0.49 + 0.04 + 0.01 = 0.70$, and against the mean $\bar{y} = \frac{13}{4} = 3.25$ the total sum of squares is $\mathrm{SS}_{\mathrm{tot}} = (2-3.25)^2 + (2-3.25)^2 + (4-3.25)^2 + (5-3.25)^2 = 1.5625 + 1.5625 + 0.5625 + 3.0625 = 6.75$. Therefore

$$R^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}} = 1 - \frac{0.70}{6.75} \approx 0.896,$$

so the line explains about 90% of the variation in $y$. As a check, the residuals sum to $0.4 - 0.7 + 0.2 + 0.1 = 0$, which the intercept term always forces.

### Problem 1.13: Projecting onto a column space

**Problem.** Let $A = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{smallmatrix}\right]$ and $\boldsymbol{b} = (1, 1, 1)$. Find the least-squares coefficients $\hat{\boldsymbol{x}}$, the projection $\hat{\boldsymbol{b}} = A\hat{\boldsymbol{x}}$ of $\boldsymbol{b}$ onto $\mathrm{Col}(A)$, and the residual, and verify the residual is orthogonal to both columns of $A$.

**Solution.** The least-squares coefficients solve the normal equations $A^\top A\,\hat{\boldsymbol{x}} = A^\top \boldsymbol{b}$. Assemble the pieces:

$$A^\top A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \qquad A^\top \boldsymbol{b} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}.$$

With $\det(A^\top A) = 2\cdot 2 - 1\cdot 1 = 3$, Cramer's rule gives $\hat{x}_1 = \frac{2\cdot 2 - 1\cdot 2}{3} = \frac{2}{3}$ and $\hat{x}_2 = \frac{2\cdot 2 - 1\cdot 2}{3} = \frac{2}{3}$, so $\hat{\boldsymbol{x}} = (\frac{2}{3}, \frac{2}{3})$. The projection is

$$\hat{\boldsymbol{b}} = A\hat{\boldsymbol{x}} = \tfrac{2}{3}(1,0,1) + \tfrac{2}{3}(0,1,1) = \left(\tfrac{2}{3}, \tfrac{2}{3}, \tfrac{4}{3}\right), \qquad \boldsymbol{r} = \boldsymbol{b} - \hat{\boldsymbol{b}} = \left(\tfrac{1}{3}, \tfrac{1}{3}, -\tfrac{1}{3}\right).$$

Verify orthogonality against each column: column $1 = (1,0,1)$ gives $\frac{1}{3} + 0 - \frac{1}{3} = 0$, and column $2 = (0,1,1)$ gives $0 + \frac{1}{3} - \frac{1}{3} = 0$. The residual is perpendicular to the whole plane $\mathrm{Col}(A)$, which is the geometric statement of the normal equations $A^\top \boldsymbol{r} = \boldsymbol{0}$. The squared error is $\|\boldsymbol{r}\|^2 = \frac{1}{9} + \frac{1}{9} + \frac{1}{9} = \frac{1}{3}$, the closest $\boldsymbol{b}$ can be approached from within the plane.

### Problem 1.14: The pseudoinverse of a tall matrix

**Problem.** For the same $A = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{smallmatrix}\right]$, form the Moore–Penrose pseudoinverse $A^\dagger = (A^\top A)^{-1} A^\top$ and confirm it returns the least-squares coefficients of Problem 1.13.

**Solution.** From Problem 1.13, $A^\top A = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ with $\det = 3$, so its inverse is

$$(A^\top A)^{-1} = \frac{1}{3}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}.$$

Multiply by $A^\top = \left[\begin{smallmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{smallmatrix}\right]$:

$$A^\dagger = (A^\top A)^{-1} A^\top = \frac{1}{3}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 & -1 & 1 \\ -1 & 2 & 1 \end{bmatrix}.$$

Applying it to $\boldsymbol{b} = (1,1,1)$,

$$A^\dagger \boldsymbol{b} = \frac{1}{3}\begin{bmatrix} 2-1+1 \\ -1+2+1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2 \\ 2 \end{bmatrix} = \left(\tfrac{2}{3}, \tfrac{2}{3}\right),$$

exactly the $\hat{\boldsymbol{x}}$ from the normal equations. The pseudoinverse packages "solve the normal equations" into a single matrix: $A^\dagger$ is the left inverse that, among all ways of mapping $\boldsymbol{b}$ back to coefficient space, returns the least-squares answer, and for a tall full-rank $A$ it satisfies $A^\dagger A = I$ (here a $2\times 2$ identity), which one can check directly.

## 12.3 Rank, structure, and norms

### Problem 1.15: Powers of a shear

**Problem.** For $A = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$, compute $A^2$, $A^3$, and conjecture a formula for $A^n$. Prove it.

**Solution.** Multiply step by step:

$$A^2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \quad A^3 = A^2 A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}.$$

The pattern is clear: $A^n = \left[\begin{smallmatrix} 1 & n \\ 0 & 1 \end{smallmatrix}\right]$. Prove it by induction. The base case $n = 1$ is $A$ itself. Assuming $A^n = \left[\begin{smallmatrix} 1 & n \\ 0 & 1 \end{smallmatrix}\right]$,

$$A^{n+1} = A^n A = \begin{bmatrix} 1 & n \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & n+1 \\ 0 & 1 \end{bmatrix},$$

completing the induction. Each power shears one unit more, so $A^n$ shears by $n$. (A defective matrix like this has no full eigenbasis, so the clean $A^n = PD^nP^{-1}$ shortcut of Chapter 3 is unavailable; direct multiplication still works.)

## Problem 1.16: Symmetric and skew-symmetric parts

**Problem.** Write $M = \left[\begin{smallmatrix} 2 & 3 \\ 1 & 4 \end{smallmatrix}\right]$ as the sum of a symmetric matrix and a skew-symmetric matrix.

**Solution.** Every square matrix splits uniquely as $M = \frac{1}{2}(M + M^\top) + \frac{1}{2}(M - M^\top)$, the first part symmetric and the second skew-symmetric. With $M^\top = \left[\begin{smallmatrix} 2 & 1 \\ 3 & 4 \end{smallmatrix}\right]$,

$$S = \tfrac{1}{2}(M + M^\top) = \tfrac{1}{2}\begin{bmatrix} 4 & 4 \\ 4 & 8 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 4 \end{bmatrix}, \quad K = \tfrac{1}{2}(M - M^\top) = \tfrac{1}{2}\begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Check: $S + K = \left[\begin{smallmatrix} 2 & 3 \\ 1 & 4 \end{smallmatrix}\right] = M$, $S^\top = S$, and $K^\top = -K$. This decomposition underlies the fact that a quadratic form $\boldsymbol{x}^\top M \boldsymbol{x}$ depends only on the symmetric part, since $\boldsymbol{x}^\top K \boldsymbol{x} = 0$ for every skew $K$.

## Problem 1.17: Rank and nullity

**Problem.** Find the rank and the nullity (dimension of the null space) of $A = \left[\begin{smallmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \end{smallmatrix}\right]$.

**Solution.** Row-reduce: $R_2 \leftarrow R_2 - 2R_1$ turns the second row into $(0, 0, 0)$, since $(2, 4, 6) = 2(1, 2, 3)$. One nonzero row survives, so $\operatorname{rank} A = 1$. The rank–nullity theorem states $\operatorname{rank} A + \dim \operatorname{Nul} A = n$ (the number of columns), here $1 + \dim \operatorname{Nul} A = 3$, so the nullity is 2. Concretely, $A\boldsymbol{x} = \boldsymbol{0}$ reduces to the single equation $x_1 + 2x_2 + 3x_3 = 0$, a plane through the origin in $\mathbb{R}^3$, which is two-dimensional. The two rows pointed in the same direction, so the map only "sees" one dimension of its input and crushes the other two to zero.

## Problem 1.18: Three norms of a vector

**Problem.** For $\boldsymbol{x} = (1, -2, 2)$, compute the $\ell_1$, $\ell_2$, and $\ell_\infty$ norms and order them.

**Solution.** Each norm measures size differently:

$$\|\boldsymbol{x}\|_1 = |1| + |-2| + |2| = 5, \quad \|\boldsymbol{x}\|_2 = \sqrt{1^2 + (-2)^2 + 2^2} = \sqrt{9} = 3, \quad \|\boldsymbol{x}\|_\infty = \max(1, 2, 2) = 2.$$

So $\|\boldsymbol{x}\|_\infty = 2 \le \|\boldsymbol{x}\|_2 = 3 \le \|\boldsymbol{x}\|_1 = 5$. The ordering $\|\cdot\|_\infty \le \|\cdot\|_2 \le \|\cdot\|_1$ holds for every vector in $\mathbb{R}^n$: the $\ell_1$ norm sums all coordinates and so is largest, while $\ell_\infty$ keeps only the biggest. These are the norms behind lasso ($\ell_1$, sparsity), ridge ($\ell_2$, shrinkage), and worst-case ($\ell_\infty$) analysis.

## Problem 1.19: $A^\top A$ is symmetric positive semidefinite

**Problem.** For $A = \left[\begin{smallmatrix} 2 & 1 \\ 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, form $A^\top A$, confirm it is symmetric, and find its eigenvalues to confirm it is positive definite.

**Solution.** Compute the Gram matrix:

$$A^\top A = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} 2 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 2 & 2 \end{bmatrix},$$

where the top-left $5 = 2^2 + 0^2 + 1^2$ is the squared length of column 1 of $A$, the bottom-right $2 = 1^2 + 1^2 + 0^2$ is column 2's squared length, and the off-diagonal $2 = 2 \cdot 1 + 0 \cdot 1 + 1 \cdot 0$ is their inner product. It is symmetric because $(A^\top A)^\top = A^\top A$ always. Its eigenvalues solve $\lambda^2 - 7\lambda + 6 = 0$ (trace 7, determinant $5 \cdot 2 - 2 \cdot 2 = 6$), giving $\lambda = 6$ and $\lambda = 1$. Both are positive, so $A^\top A$ is positive definite. This is no coincidence: for any $A$ with independent columns, $\boldsymbol{z}^\top A^\top A \boldsymbol{z} = \|A\boldsymbol{z}\|^2 > 0$ for $\boldsymbol{z} \neq \boldsymbol{0}$, which is exactly why the normal equations have a unique solution.

## Problem 1.20: The triangle inequality

**Problem.** For $\boldsymbol{u} = (3, 0)$ and $\boldsymbol{v} = (0, 4)$, verify the triangle inequality $\|\boldsymbol{u} + \boldsymbol{v}\| \le \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$ and explain when equality holds.

**Solution.** The sum is $\boldsymbol{u} + \boldsymbol{v} = (3, 4)$, so

$$\|\boldsymbol{u} + \boldsymbol{v}\| = \sqrt{3^2 + 4^2} = 5, \qquad \|\boldsymbol{u}\| + \|\boldsymbol{v}\| = 3 + 4 = 7.$$

Indeed $5 \le 7$, so the inequality holds with room to spare. Equality $\|\boldsymbol{u} + \boldsymbol{v}\| = \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$ requires $\boldsymbol{u}$ and $\boldsymbol{v}$ to point in the same direction (one a nonnegative multiple of the other); here they are orthogonal, the farthest from parallel, so the gap is largest. Geometrically, the straight path from the origin to $\boldsymbol{u} + \boldsymbol{v}$ (length 5) is shorter than detouring through the tip of $\boldsymbol{u}$ (length $3 + 4 = 7$), the reason it is called the triangle inequality.

CHAPTER 13

# Problem Set 2: Eigenvalues, Diagonalization, and the SVD

This set works through Chapters 3 and 4: characteristic polynomials, eigenvectors, diagonalization and matrix powers, the spectral theorem, definiteness, the Rayleigh quotient, power iteration, complex and defective spectra, then the singular value decomposition, low-rank approximation, conditioning, and the pseudoinverse. As before, every problem is restated and solved step by step, and every eigenvalue, singular value, and reconstruction is one you can check by re-multiplying.

## 13.1 Eigenvalues and the spectral theorem

### Problem 2.1: Eigenvalues and eigenvectors of a $2 \times 2$

**Problem.** Find the eigenvalues and eigenvectors of $A = \left[\begin{smallmatrix} 4 & -2 \\ 1 & 1 \end{smallmatrix}\right]$.

**Solution.** The eigenvalues are the roots of $\det(A - \lambda I) = 0$. With trace 5 and determinant $4 \cdot 1 - (-2) \cdot 1 = 6$,

$$p_A(\lambda) = \lambda^2 - 5\lambda + 6 = (\lambda - 2)(\lambda - 3),$$

so $\lambda_1 = 3$, $\lambda_2 = 2$. For $\lambda = 3$, $(A - 3I) = \left[\begin{smallmatrix} 1 & -2 \\ 1 & -2 \end{smallmatrix}\right]$ gives $v_1 = 2v_2$, so $\boldsymbol{v}_1 = (2, 1)$; check $A(2, 1) = (8 - 2,\ 2 + 1) = (6, 3) = 3(2, 1)$. For $\lambda = 2$, $(A - 2I) = \left[\begin{smallmatrix} 2 & -2 \\ 1 & -1 \end{smallmatrix}\right]$ gives $v_1 = v_2$, so $\boldsymbol{v}_2 = (1, 1)$; check $A(1, 1) = (2, 2) = 2(1, 1)$. Both confirm.

### Problem 2.2: Diagonalization and a matrix power

**Problem.** Diagonalize $A = \left[\begin{smallmatrix} 3 & 1 \\ 1 & 3 \end{smallmatrix}\right]$ and use it to compute $A^2$.

**Solution.** The matrix is symmetric, so the spectral theorem applies. Its characteristic polynomial is $(3 - \lambda)^2 - 1 = 0$, i.e. $3 - \lambda = \pm 1$, giving $\lambda_1 = 4$ and $\lambda_2 = 2$. The eigenvectors are $\boldsymbol{v}_1 = (1, 1)$ (for 4) and $\boldsymbol{v}_2 = (1, -1)$ (for 2), orthogonal as promised. With $P = \left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$, $D = \operatorname{diag}(4, 2)$, and $P^{-1} = \frac{1}{2}\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$,

$$A^2 = PD^2P^{-1} = \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 16 & 0 \\ 0 & 4 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 10 & 6 \\ 6 & 10 \end{bmatrix}.$$

Direct squaring confirms it: $\left[\begin{smallmatrix} 3 & 1 \\ 1 & 3 \end{smallmatrix}\right]^2 = \left[\begin{smallmatrix} 9+1 & 3+3 \\ 3+3 & 1+9 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 10 & 6 \\ 6 & 10 \end{smallmatrix}\right]$. The point of diagonalization is that any power costs the same: $A^k$ just raises the eigenvalues to the $k$.

### Problem 2.3: A spectral decomposition

**Problem.** Write $A = \left[\begin{smallmatrix} 2 & -1 \\ -1 & 2 \end{smallmatrix}\right]$ in the spectral form $A = \lambda_1 \boldsymbol{q}_1\boldsymbol{q}_1^{\top} + \lambda_2 \boldsymbol{q}_2\boldsymbol{q}_2^{\top}$ with orthonormal $\boldsymbol{q}_i$.

**Solution.** The eigenvalues come from $(2 - \lambda)^2 - 1 = 0$, so $\lambda_1 = 3$, $\lambda_2 = 1$. For $\lambda = 3$, $(A - 3I) = \left[\begin{smallmatrix} -1 & -1 \\ -1 & -1 \end{smallmatrix}\right]$ gives $\boldsymbol{q}_1 \propto (1, -1)$, normalized $\frac{1}{\sqrt{2}}(1, -1)$. For $\lambda = 1$, $\boldsymbol{q}_2 = \frac{1}{\sqrt{2}}(1, 1)$. The outer products are

$$\boldsymbol{q}_1\boldsymbol{q}_1^{\top} = \tfrac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \qquad \boldsymbol{q}_2\boldsymbol{q}_2^{\top} = \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

so

$$A = 3 \cdot \tfrac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + 1 \cdot \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \tfrac{1}{2}\begin{bmatrix} 3+1 & -3+1 \\ -3+1 & 3+1 \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}.$$

The decomposition rebuilds $A$ as a weighted sum of orthogonal projectors, each onto an eigenvector, with the eigenvalue as the weight.

## Problem 2.4: Eigenvalues of a triangular matrix

**Problem.** Find the eigenvalues of $A = \left[\begin{smallmatrix} 2 & 0 & 0 \\ 1 & 3 & 0 \\ 0 & 1 & 4 \end{smallmatrix}\right]$ and an eigenvector for the smallest one.

**Solution.** For a triangular matrix the determinant of $A-\lambda I$ is the product of the diagonal entries $(2-\lambda)(3-\lambda)(4-\lambda)$, so the eigenvalues are exactly the diagonal: $\lambda = 2, 3, 4$. For the smallest, $\lambda = 2$, solve $(A - 2I)\boldsymbol{v} = \mathbf{0}$:

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 2 \end{bmatrix} \boldsymbol{v} = \mathbf{0} \;\Rightarrow\; v_1 + v_2 = 0,\; v_2 + 2v_3 = 0.$$

Take $v_3 = 1$, then $v_2 = -2$ and $v_1 = 2$, so $\boldsymbol{v} = (2, -2, 1)$. Triangular matrices hand you their eigenvalues for free, which is exactly why elimination and the $QR$ algorithm aim to make a matrix triangular.

## Problem 2.5: Trace, determinant, and the eigenvalues

**Problem.** Using $A = \left[\begin{smallmatrix} 4 & -2 \\ 1 & 1 \end{smallmatrix}\right]$ from Problem 2.1, verify that the trace equals the sum of the eigenvalues and the determinant equals their product.

**Solution.** From Problem 2.1 the eigenvalues are 3 and 2. The trace is $4 + 1 = 5 = 3 + 2$, and the determinant is $4 \cdot 1 - (-2) \cdot 1 = 6 = 3 \cdot 2$. Both identities hold. They are general: expanding the characteristic polynomial $\prod_i (\lambda - \lambda_i)$ shows the coefficient of $\lambda^{n-1}$ is $-\sum \lambda_i$ (the trace) and the constant term is $\prod \lambda_i$ (the determinant). A zero among the eigenvalues would force $\det A = 0$, the signature of a singular matrix.

## Problem 2.6: Classifying quadratic forms

**Problem.** Classify the quadratic forms $f(x, y) = 2x^2 + 2xy + 2y^2$ and $g(x, y) = x^2 - 4xy + y^2$ as positive definite, negative definite, or indefinite.

**Solution.** Write each as $\boldsymbol{x}^\top M \boldsymbol{x}$ with $M$ symmetric, splitting the cross term evenly. For $f$, $M_f = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ with eigenvalues from $(2 - \lambda)^2 - 1 = 0$, i.e. $\lambda = 3, 1$, both positive, so $f$ is *positive definite*: $f > 0$ for every nonzero $\boldsymbol{x}$. For $g$, $M_g = \left[\begin{smallmatrix} 1 & -2 \\ -2 & 1 \end{smallmatrix}\right]$ with eigenvalues from $(1-\lambda)^2 - 4 = 0$, i.e. $1 - \lambda = \pm 2$, giving $\lambda = 3, -1$. Mixed signs make $g$ *indefinite*: it curves up along one eigenvector and down along the other, a saddle. The sign pattern of the eigenvalues is the whole story of a quadratic form's shape.

## Problem 2.7: The Rayleigh quotient

**Problem.** For $A = \left[\begin{smallmatrix} 5 & 2 \\ 2 & 2 \end{smallmatrix}\right]$, find the maximum and minimum of $\boldsymbol{x}^\top A \boldsymbol{x}$ over unit vectors, and the directions where they occur.

**Solution.** For a symmetric matrix the Rayleigh quotient $\boldsymbol{x}^\top A\boldsymbol{x}/\boldsymbol{x}^\top \boldsymbol{x}$ ranges exactly between the smallest and largest eigenvalues, attained at the corresponding eigenvectors. The eigenvalues solve $\lambda^2 - 7\lambda + 6 = 0$ (trace 7, determinant $10 - 4 = 6$), giving $\lambda_{\max} = 6$ and $\lambda_{\min} = 1$. So over unit vectors $\boldsymbol{x}^\top A\boldsymbol{x}$ reaches a maximum of 6 and a minimum of 1. For $\lambda = 6$, $(A - 6I) = \left[\begin{smallmatrix} -1 & 2 \\ 2 & -4 \end{smallmatrix}\right]$ gives the direction $\frac{1}{\sqrt{5}}(2, 1)$; for $\lambda = 1$, the direction is $\frac{1}{\sqrt{5}}(-1, 2)$. This is the variational principle behind PCA: the top eigenvector is the direction of maximum quadratic value, which for a covariance matrix means maximum variance.

### Problem 2.8: Power iteration by hand

**Problem.** Run power iteration on $A = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ from $\boldsymbol{x}_0 = (1, 0)$ for four steps, tracking the Rayleigh quotient, and say what it converges to.

**Solution.** Each step applies $A$; normalization only rescales, so track the raw iterates and the Rayleigh quotient $R(\boldsymbol{x}) = \boldsymbol{x}^\top A\boldsymbol{x}/\boldsymbol{x}^\top\boldsymbol{x}$:

| $\boldsymbol{x}_k$ | $A\boldsymbol{x}_k$ | $R(\boldsymbol{x}_k)$ |
|---|---|---|
| $(1, 0)$ | $(2, 1)$ | $2/1 = 2.000$ |
| $(2, 1)$ | $(5, 4)$ | $14/5 = 2.800$ |
| $(5, 4)$ | $(14, 13)$ | $122/41 = 2.976$ |
| $(14, 13)$ | $(41, 40)$ | $1094/365 = 2.997$ |

The direction tilts toward $(1, 1)$ and the Rayleigh quotient climbs toward 3, the dominant eigenvalue of $A$ (its eigenvalues are 3 and 1). The error shrinks by the ratio $|\lambda_2/\lambda_1| = 1/3$ each step, so convergence is geometric. This is the seed of the algorithms that compute PageRank and principal components at scale.

### Problem 2.9: Complex eigenvalues

**Problem.** Find the eigenvalues of $A = \left[\begin{smallmatrix} 1 & -1 \\ 1 & 1 \end{smallmatrix}\right]$ and their magnitudes.

**Solution.** The trace is 2 and the determinant is $1 \cdot 1 - (-1) \cdot 1 = 2$, so $p_A(\lambda) = \lambda^2 - 2\lambda + 2$ and

$$\lambda = \frac{2 \pm \sqrt{4 - 8}}{2} = \frac{2 \pm 2i}{2} = 1 \pm i.$$

The discriminant is negative, so the eigenvalues are the complex conjugate pair $1 \pm i$, with magnitude $|1 \pm i| = \sqrt{1^2 + 1^2} = \sqrt{2}$. A real matrix can have no real eigenvalue: this $A$ is a rotation by $45^\circ$ scaled by $\sqrt{2}$, and a rotation turns every real vector, so none is merely stretched. Magnitude greater than 1 means iterating $A$ spirals outward.

### Problem 2.10: A defective matrix

**Problem.** Show that $A = \left[\begin{smallmatrix} 2 & 1 \\ 0 & 2 \end{smallmatrix}\right]$ is not diagonalizable.

**Solution.** The characteristic polynomial is $(2 - \lambda)^2$, so $\lambda = 2$ with algebraic multiplicity 2. The eigenvectors solve $(A - 2I)\boldsymbol{v} = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]\boldsymbol{v} = \boldsymbol{0}$, which forces the second component to be zero and leaves $\boldsymbol{v} = (1, 0)$ (up to scale). The geometric multiplicity is only 1, short of the 2 needed to span $\mathbb{R}^2$. With too few independent eigenvectors there is no eigenbasis, so $A$ cannot be written $PDP^{-1}$: it is *defective*. Such matrices are exactly where diagonalization fails and the Jordan form or the SVD must take over.

## 13.2 The singular value decomposition

### Problem 2.11: A singular value decomposition

**Problem.** Find the SVD of $A = \left[\begin{smallmatrix} 0 & 2 \\ 3 & 0 \end{smallmatrix}\right]$.

**Solution.** The right singular vectors are eigenvectors of $A^\top A$, and the singular values are the square roots of its eigenvalues. Here

$$A^\top A = \begin{bmatrix} 0 & 3 \\ 2 & 0 \end{bmatrix}\begin{bmatrix} 0 & 2 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 9 & 0 \\ 0 & 4 \end{bmatrix},$$

which is diagonal, so its eigenvectors are $\boldsymbol{e}_1, \boldsymbol{e}_2$ and its eigenvalues are $9, 4$; the singular values are $\sigma_1 = 3$, $\sigma_2 = 2$. Take $\boldsymbol{v}_1 = (1, 0)$, $\boldsymbol{v}_2 = (0, 1)$. The left singular vectors come from $\boldsymbol{u}_i = A\boldsymbol{v}_i/\sigma_i$:

$$\boldsymbol{u}_1 = \tfrac{1}{3}A(1, 0) = \tfrac{1}{3}(0, 3) = (0, 1), \qquad \boldsymbol{u}_2 = \tfrac{1}{2}A(0, 1) = \tfrac{1}{2}(2, 0) = (1, 0).$$

Hence

$$A = U\Sigma V^\top = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The map first reads off the input coordinates ($V^\top = I$), stretches them by 3 and 2, then swaps the axes ($U$). Multiplying the three factors returns $A$.

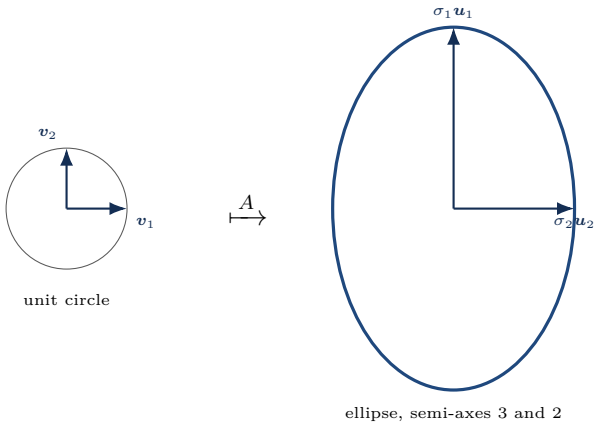



**Figure 13.1.** The SVD as geometry: $A$ sends the unit circle to an ellipse whose semi-axis lengths are the singular values $\sigma_1 = 3,\ \sigma_2 = 2$, along the left singular vectors. Here $A$ also swaps the axes, so the long axis $\sigma_1 \boldsymbol{u}_1$ points up.

## Problem 2.12: Condition number

**Problem.** A matrix has singular values $\sigma = (4, 1)$. Give its 2-norm condition number and the worst-case relative error amplification when solving a linear system with it.

**Solution.** The condition number is the ratio of largest to smallest singular value,

$$\kappa_2 = \frac{\sigma_{\max}}{\sigma_{\min}} = \frac{4}{1} = 4.$$

Solving $A\boldsymbol{x} = \boldsymbol{b}$ can amplify a relative error in $\boldsymbol{b}$ by at most $\kappa_2$: a 1% error in the data can become as much as a 4% error in the solution. A condition number near 1 is benign; a large $\kappa_2$ (a tiny $\sigma_{\min}$) signals a nearly singular, sensitive problem, which is the numerical reason ridge regularization lifts the small singular values away from zero.

## Problem 2.13: Low-rank approximation error

**Problem.** A matrix has singular values $\sigma = (5, 3, 1)$. By the Eckart–Young theorem, what is the Frobenius error of the best rank-1 and best rank-2 approximations?

**Solution.** Eckart–Young says the best rank-$k$ approximation keeps the top $k$ terms of the SVD, and the squared Frobenius error is the sum of the squares of the dropped singular values. Dropping $\sigma_2, \sigma_3$ for the rank-1 fit,

$$\|A - A_1\|_F = \sqrt{\sigma_2^2 + \sigma_3^2} = \sqrt{9 + 1} = \sqrt{10} \approx 3.16.$$

Dropping only $\sigma_3$ for the rank-2 fit,

$$\|A - A_2\|_F = \sqrt{\sigma_3^2} = \sqrt{1} = 1.$$

Each retained singular value buys back its own energy, so a sharp drop in the spectrum is exactly what makes truncation faithful.

## Problem 2.14: Pseudoinverse of a rank-one matrix

**Problem.** Find the Moore–Penrose pseudoinverse of the rank-deficient $A = \left[\begin{smallmatrix} 1 & 2 \\ 2 & 4 \end{smallmatrix}\right]$.

**Solution.** The matrix is rank one: both columns lie along $(1, 2)$. Its single nonzero singular value satisfies $\sigma_1^2 = \|A\|_F^2 = 1 + 4 + 4 + 16 = 25$, so $\sigma_1 = 5$, with $\boldsymbol{u}_1 = \boldsymbol{v}_1 = \frac{1}{\sqrt{5}}(1, 2)$. The SVD pseudoinverse inverts the nonzero singular value and transposes,

$$A^\dagger = \frac{1}{\sigma_1} \boldsymbol{v}_1 \boldsymbol{u}_1^\top = \frac{1}{5} \cdot \frac{1}{5} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} = \frac{1}{25} \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} = \frac{A}{25}.$$

Verify the defining identity $AA^\dagger A = A$: since $A^2 = \left[\begin{smallmatrix} 5 & 10 \\ 10 & 20 \end{smallmatrix}\right] = 5A$, we get $AA^\dagger A = A \cdot \frac{A}{25} \cdot A = \frac{A^3}{25} = \frac{25A}{25} = A$. The ordinary formula $(A^\top A)^{-1} A^\top$ is unavailable here because $A^\top A$ is singular, yet the pseudoinverse exists and is unique.

### Problem 2.15: Energy retained by truncation

**Problem.** With singular values $\sigma = (6, 3, 2)$, what fraction of the Frobenius energy is kept by the best rank-1 and rank-2 approximations?

**Solution.** The total energy is $\sum \sigma_i^2 = 36 + 9 + 4 = 49$. The rank-1 fit keeps $\sigma_1^2 = 36$, a fraction $36/49 \approx 0.735$, so about 73.5%. The rank-2 fit keeps $36 + 9 = 45$, a fraction $45/49 \approx 0.918$, about 91.8%. Plotting these cumulative fractions against the rank is the scree curve; one truncates where it flattens, typically at a target such as 90–95%.

### Problem 2.16: Storage in a low-rank approximation

**Problem.** A $200 \times 100$ matrix is stored by its best rank-5 approximation. How many numbers does that take, and what is the compression ratio?

**Solution.** A rank-$k$ truncation stores $k$ left singular vectors of length $m$, $k$ right singular vectors of length $n$, and $k$ singular values: $k(m + n + 1)$ numbers. Here

$$5(200 + 100 + 1) = 5(301) = 1505,$$

against the full $200 \cdot 100 = 20{,}000$. The compression ratio is $20{,}000/1505 \approx 13.3\times$, keeping about 7.5% of the storage. Low rank pays off whenever $k \ll mn/(m + n)$, which is why it underlies image compression and compact model representations.

### Problem 2.17: Spectral and Frobenius norms

**Problem.** A matrix has singular values $\sigma = (6, 3, 2)$. Give its spectral (2-) norm and its Frobenius norm.

**Solution.** The spectral norm is the largest singular value, the most a unit vector can be stretched: $\|A\|_2 = \sigma_1 = 6$. The Frobenius norm is the square root of the sum of squared singular values (equivalently the square root of the sum of squared entries):

$$\|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2} = \sqrt{36 + 9 + 4} = \sqrt{49} = 7.$$

Always $\|A\|_2 \le \|A\|_F$, with equality only for a rank-one matrix; here $6 < 7$ because three singular values contribute to the Frobenius norm but only the largest sets the spectral norm.

### Problem 2.18: For a symmetric PD matrix, SVD equals eigendecomposition

**Problem.** Show that for $A = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ the singular values equal the eigenvalues and the singular vectors equal the eigenvectors.

**Solution.** The matrix is symmetric with eigenvalues 3 and 1 (from $(2 - \lambda)^2 - 1 = 0$) and orthonormal eigenvectors $\boldsymbol{q}_1 = \frac{1}{\sqrt{2}}(1, 1)$, $\boldsymbol{q}_2 = \frac{1}{\sqrt{2}}(1, -1)$. Because $A$ is symmetric, $A^\top A = A^2$, whose eigenvalues are $\lambda_i^2 = 9, 1$ with the same eigenvectors. The singular values are therefore $\sigma_i = \sqrt{\lambda_i^2} = |\lambda_i| = 3, 1$, equal to the eigenvalues since both are positive, and the singular vectors are the eigenvectors $\boldsymbol{q}_i$. So $A = Q\Lambda Q^\top$ and $A = U\Sigma V^\top$ coincide. This is special to symmetric positive definite matrices; for a symmetric matrix with a *negative* eigenvalue the singular value is its absolute value and a sign moves into the singular vectors, so the two factorizations differ only by that sign.

## 13.3 Further eigenvalue and SVD practice

### Problem 2.19: Eigenvalues of the inverse

**Problem.** Given that $A = \left[\begin{smallmatrix} 3 & 1 \\ 1 & 3 \end{smallmatrix}\right]$ has eigenvalues 4 and 2, find the eigenvalues of $A^{-1}$ without inverting $A$.

**Solution.** If $A\boldsymbol{v} = \lambda\boldsymbol{v}$ with $\lambda \neq 0$, multiply both sides by $A^{-1}$ and divide by $\lambda$: $A^{-1}\boldsymbol{v} = \frac{1}{\lambda}\boldsymbol{v}$. So $A^{-1}$ has the same eigenvectors as $A$ with *reciprocal* eigenvalues. Here the eigenvalues of $A^{-1}$ are $\frac{1}{4}$ and $\frac{1}{2}$. This is why a large condition number hurts: if $A$ has a tiny eigenvalue, $A^{-1}$ has a huge one, and solving $A\boldsymbol{x} = \boldsymbol{b}$ blows that direction up.

## Problem 2.20: $A$ and $A^\top$ share eigenvalues

**Problem.** Verify that $A = \left[\begin{smallmatrix} 4 & -2 \\ 1 & 1 \end{smallmatrix}\right]$ and $A^\top = \left[\begin{smallmatrix} 4 & 1 \\ -2 & 1 \end{smallmatrix}\right]$ have the same eigenvalues.

**Solution.** Both have trace 5 and determinant $4 - (-2) = 6$ (transposing changes neither), so both have characteristic polynomial $\lambda^2 - 5\lambda + 6$ and the same eigenvalues $3, 2$. In general $\det(A^\top - \lambda I) = \det((A - \lambda I)^\top) = \det(A - \lambda I)$, so $A$ and $A^\top$ always share a characteristic polynomial, hence eigenvalues. Their eigen*vectors* generally differ, though: the eigenvectors of $A^\top$ are the *left* eigenvectors of $A$.

## Problem 2.21: Eigenvalues of a rotation have modulus one

**Problem.** Find the eigenvalues of the orthogonal matrix $Q = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$ and confirm each has modulus 1.

**Solution.** The trace is 0 and the determinant is 1, so $p_Q(\lambda) = \lambda^2 + 1$ and $\lambda = \pm i$. Each has modulus $|\pm i| = 1$. This is general for orthogonal matrices: they preserve length, so $\|Q\boldsymbol{v}\| = \|\boldsymbol{v}\|$ forces $|\lambda| = 1$ for every (possibly complex) eigenvalue. A rotation neither stretches nor shrinks, so its eigenvalues live on the unit circle.

## Problem 2.22: A cube by diagonalization

**Problem.** Compute $A^3$ for $A = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ using its eigendecomposition.

**Solution.** The eigenvalues are 3 and 1 with eigenvectors $(1, 1)$ and $(1, -1)$. Cubing the eigenvalues gives 27 and 1, and rebuilding,

$$A^3 = P\,\mathrm{diag}(27, 1)P^{-1} = \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 27 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 28 & 26 \\ 26 & 28 \end{bmatrix} = \begin{bmatrix} 14 & 13 \\ 13 & 14 \end{bmatrix}.$$

Diagonalization turns a matrix power into a scalar power on the eigenvalues, far cheaper than multiplying $A$ by itself three times once the matrices grow.

## Problem 2.23: SVD of a wide matrix

**Problem.** Find the singular values of the $2 \times 3$ matrix $A = \left[\begin{smallmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{smallmatrix}\right]$.

**Solution.** For a wide matrix it is easier to use the smaller Gram matrix $AA^\top$ (which is $2 \times 2$ here) whose eigenvalues are the squared singular values:

$$AA^\top = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

Its eigenvalues are 3 and 1, so the singular values are $\sigma_1 = \sqrt{3}$ and $\sigma_2 = 1$. The matrix has rank 2 (two nonzero singular values), so its three columns span all of $\mathbb{R}^2$; the third dimension of the input is sent to zero, the lone trivial direction of a wide map.

## Problem 2.24: Rank from the singular values

**Problem.** A matrix has singular values $\sigma = (4, 2, 0, 0)$. What is its rank, and what does this say about its four fundamental subspaces?

**Solution.** The rank is the number of nonzero singular values, here 2. So the column space and row space are each 2-dimensional, spanned by the first two left and right singular vectors respectively. The two zero singular values mark the null space: their right singular vectors span a 2-dimensional null space (assuming the matrix has at least four columns), the directions the matrix annihilates. The SVD reads off all four fundamental subspaces at once from where the singular values cross from positive to zero.

## Problem 2.25: A covariance matrix and its principal directions

**Problem.** A dataset has covariance matrix $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 4 & 2 \\ 2 & 4 \end{smallmatrix}\right]$. Find the principal directions and the fraction of variance captured by the first.

**Solution.** The principal directions are the eigenvectors of $\boldsymbol{\Sigma}$, and the variance along each is its eigenvalue. From $(4-\lambda)^2 - 4 = 0$, i.e. $4 - \lambda = \pm 2$, the eigenvalues are $\lambda_1 = 6$ and $\lambda_2 = 2$. The first eigenvector solves $(\boldsymbol{\Sigma} - 6I)\boldsymbol{v} = \left[\begin{smallmatrix} -2 & 2 \\ 2 & -2 \end{smallmatrix}\right]\boldsymbol{v} = \mathbf{0}$, giving the 45° direction $\frac{1}{\sqrt{2}}(1,1)$; the second is $\frac{1}{\sqrt{2}}(1,-1)$. The first principal component captures

$$\frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{6}{6+2} = 0.75,$$

so 75% of the variance lies along $(1,1)$. The positive correlation tilts the data cloud along the diagonal, exactly the direction PCA selects.

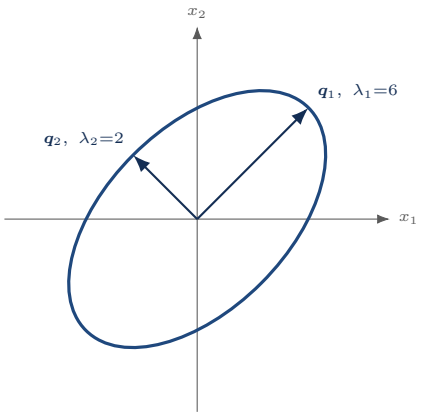



**Figure 13.2.** The covariance ellipse for the matrix with diagonal 4 and off-diagonal 2. Its axes are the eigenvectors $(1,1)$ and $(1,-1)$, and the semi-axis lengths are $\sqrt{\lambda_i}$. The first principal component points along the 45° diagonal and carries 75% of the variance.

## Problem 2.26: The spectral norm as a stretch bound

**Problem.** A matrix $A$ has singular values $\sigma = (3,1)$. What is the largest possible value of $\|A\boldsymbol{x}\|$ over unit vectors $\boldsymbol{x}$, and in what direction is it achieved?

**Solution.** Writing $\boldsymbol{x}$ in the right-singular-vector basis, $\boldsymbol{x} = c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2$ with $c_1^2 + c_2^2 = 1$, the image has squared length $\|A\boldsymbol{x}\|^2 = \sigma_1^2 c_1^2 + \sigma_2^2 c_2^2 = 9c_1^2 + c_2^2$. This is largest when all the weight sits on the top singular direction, $c_1 = 1, c_2 = 0$, giving $\|A\boldsymbol{x}\| = \sigma_1 = 3$, achieved at $\boldsymbol{x} = \boldsymbol{v}_1$. So $\|A\boldsymbol{x}\| \le \sigma_1\|\boldsymbol{x}\|$ for every $\boldsymbol{x}$, with equality along the top right singular vector: the spectral norm $\|A\|_2 = \sigma_1 = 3$ is exactly the worst-case stretch the matrix applies.

## Problem 2.27: The Cayley–Hamilton theorem on a $2 \times 2$

**Problem.** Verify that $A = \left[\begin{smallmatrix} 4 & -2 \\ 1 & 1 \end{smallmatrix}\right]$ satisfies its own characteristic polynomial $p_A(\lambda) = \lambda^2 - 5\lambda + 6$.

**Solution.** Cayley–Hamilton claims $p_A(A) = A^2 - 5A + 6I = 0$ (the zero matrix). Compute the pieces. First $A^2 = \left[\begin{smallmatrix} 4 & -2 \\ 1 & 1 \end{smallmatrix}\right]\left[\begin{smallmatrix} 4 & -2 \\ 1 & 1 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 16-2 & -8-2 \\ 4+1 & -2+1 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 14 & -10 \\ 5 & -1 \end{smallmatrix}\right]$. Then

$$A^2 - 5A + 6I = \begin{bmatrix} 14 & -10 \\ 5 & -1 \end{bmatrix} - \begin{bmatrix} 20 & -10 \\ 5 & 5 \end{bmatrix} + \begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix} = \begin{bmatrix} 14-20+6 & -10+10+0 \\ 5-5+0 & -1-5+6 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The theorem holds. One payoff: it lets you write $A^2 = 5A - 6I$, reducing any power of $A$ to a linear combination of $A$ and $I$, and (when $A$ is invertible) it gives $A^{-1} = \frac{1}{6}(5I - A)$ for free.

## Problem 2.28: Shifting a matrix shifts its eigenvalues

**Problem.** Given that $A = \left[\begin{smallmatrix} 3 & 1 \\ 1 & 3 \end{smallmatrix}\right]$ has eigenvalues $4, 2$, find the eigenvalues of $A + 2I$.

**Solution.** If $A\boldsymbol{v} = \lambda\boldsymbol{v}$ then $(A + cI)\boldsymbol{v} = (\lambda + c)\boldsymbol{v}$, so adding $cI$ shifts every eigenvalue by $c$ and leaves the eigenvectors untouched. With $c = 2$ the eigenvalues become 6 and 4. A direct check agrees: $A + 2I = \left[\begin{smallmatrix} 5 & 1 \\ 1 & 5 \end{smallmatrix}\right]$ has trace $10 = 6 + 4$ and determinant $24 = 6 \cdot 4$. This shift is exactly the trick ridge regression uses on $A^\top A$: adding $\lambda I$ lifts every eigenvalue by $\lambda$, pushing the smallest away from zero and curing near-singularity.

## Problem 2.29: Determinant as the product of singular values

**Problem.** For $A = \left[\begin{smallmatrix} 0 & 2 \\ 3 & 0 \end{smallmatrix}\right]$ with singular values $3, 2$ (Problem 2.11), confirm $|\det A| = \sigma_1\sigma_2$.

**Solution.** Directly, $\det A = 0 \cdot 0 - 2 \cdot 3 = -6$, so $|\det A| = 6$, which equals $\sigma_1\sigma_2 = 3 \cdot 2 = 6$. The identity follows from $A = U\Sigma V^\top$ and the multiplicativity of determinants: $\det A = \det U \det \Sigma \det V^\top$, and since $U, V$ are orthogonal their determinants are $\pm 1$, leaving $|\det A| = \det \Sigma = \prod_i \sigma_i$. Geometrically, $|\det A|$ is the volume scaling of the map, and the SVD says that scaling is the product of the axiswise stretches $\sigma_i$.

## Problem 2.30: Trace equals the sum of eigenvalues, $3 \times 3$

**Problem.** Confirm the trace identity for the triangular matrix $A = \left[\begin{smallmatrix} 2 & 0 & 0 \\ 1 & 3 & 0 \\ 0 & 1 & 4 \end{smallmatrix}\right]$ of Problem 2.4.

**Solution.** The eigenvalues are the diagonal entries $2, 3, 4$ (Problem 2.4), so their sum is 9. The trace is the diagonal sum $2 + 3 + 4 = 9$. They match, as the identity $\operatorname{tr} A = \sum_i \lambda_i$ guarantees for every square matrix. The trace gives a one-glance estimate of the eigenvalues' total even when the individual values are hard to find, and combined with $\det A = \prod_i \lambda_i = 24$ it pins down a great deal of the spectrum.

## Problem 2.31: Eigenvalues of $A^2$

**Problem.** Given that $A = \left[\begin{smallmatrix} 3 & 1 \\ 1 & 3 \end{smallmatrix}\right]$ has eigenvalues 4 and 2, find the eigenvalues of $A^2$ without squaring the matrix.

**Solution.** If $A\boldsymbol{v} = \lambda\boldsymbol{v}$ then $A^2\boldsymbol{v} = A(\lambda\boldsymbol{v}) = \lambda A\boldsymbol{v} = \lambda^2\boldsymbol{v}$, so $A^2$ has the same eigenvectors with squared eigenvalues. The eigenvalues of $A^2$ are $4^2 = 16$ and $2^2 = 4$. A direct check agrees: $A^2 = \left[\begin{smallmatrix} 10 & 6 \\ 6 & 10 \end{smallmatrix}\right]$ has trace $20 = 16 + 4$ and determinant $64 = 16 \cdot 4$. More generally $A^k$ has eigenvalues $\lambda^k$, which is why powers of a matrix are dominated by its largest eigenvalue, the fact behind power iteration.

## Problem 2.32: Eigenvalues of a projection

**Problem.** Show that the projection matrix $P = \frac{1}{2}\left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$ has eigenvalues 1 and 0, and explain why every projection does.

**Solution.** The trace is 1 and the determinant is $\frac{1}{4}(1 \cdot 1 - 1 \cdot 1) = 0$, so the eigenvalues satisfy $\lambda^2 - \lambda = 0$, giving $\lambda = 1$ and $\lambda = 0$. The eigenvector for 1 is $(1, 1)$ (the line $P$ projects onto), and for 0 it is $(1, -1)$ (the direction $P$ kills). Every projection satisfies $P^2 = P$, so its eigenvalues obey $\lambda^2 = \lambda$, forcing $\lambda \in \{0, 1\}$: a projection leaves its range fixed (eigenvalue 1) and annihilates its complement (eigenvalue 0). The number of 1s equals the rank, here 1.

## Problem 2.33: Gershgorin disks

**Problem.** Use Gershgorin's theorem to bound the eigenvalues of $A = \left[\begin{smallmatrix} 5 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, then compare with the true values.

**Solution.** Gershgorin's theorem places every eigenvalue in a disk centered at a diagonal entry with radius the sum of the absolute off-diagonal entries in that row. Row 1 gives the disk centered at 5 with radius 1, the interval $[4, 6]$; row 2 gives center 2, radius 1, the interval $[1, 3]$. So the eigenvalues lie in $[4, 6] \cup [1, 3]$. The true eigenvalues solve $\lambda^2 - 7\lambda + 9 = 0$, i.e. $\lambda = \frac{7 \pm \sqrt{13}}{2} \approx 5.30$ and $1.70$, comfortably inside the two disks. Gershgorin gives a cheap, no-arithmetic localization, useful for spotting whether a matrix is, say, diagonally dominant and hence invertible at a glance.

## Problem 2.34: Trace of $A^2$ is the sum of squared eigenvalues

**Problem.** For $A = \left[\begin{smallmatrix} 3 & 1 \\ 1 & 3 \end{smallmatrix}\right]$ (eigenvalues $4, 2$), verify $\operatorname{tr}(A^2) = \sum_i \lambda_i^2$.

**Solution.** The eigenvalues of $A^2$ are 16 and 4 (Problem 2.31), and the trace is the sum of eigenvalues, so $\operatorname{tr}(A^2) = 16 + 4 = 20$. Directly, $A^2 = \left[\begin{smallmatrix} 10 & 6 \\ 6 & 10 \end{smallmatrix}\right]$ has trace $10 + 10 = 20$. They agree. The identity $\operatorname{tr}(A^2) = \sum_i \lambda_i^2$ holds for any square matrix, and for a symmetric $A$ it also equals the squared Frobenius norm $\sum_{ij} A_{ij}^2 = 9 + 1 + 1 + 9 = 20$, linking the eigenvalues directly to the raw entries.

CHAPTER 14

# Problem Set 3: Probability, Distributions, and Learning Theory

This set covers Chapters 5 to 7: Bayes' theorem and base rates, expectation and variance, the binomial and Poisson laws, maximum likelihood and conjugate Bayesian updates, covariance and correlation, the Gaussian under affine maps and conditioning, and the concentration inequalities (Markov, Chebyshev, Hoeffding, Chernoff) with VC dimension and the bias–variance split. Every probability is computed from the rules, every estimator derived, and every bound stated with its constants.

## 14.1 Bayes, expectation, and estimation

### Problem 3.1: A medical test and the base rate

**Problem.** A disease affects 2% of a population. A test is 99% sensitive ($\mathbb{P}(+ \mid D) = 0.99$) and 95% specific ($\mathbb{P}(- \mid \neg D) = 0.95$). A patient tests positive. What is $\mathbb{P}(D \mid +)$?

**Solution.** By Bayes' theorem, $\mathbb{P}(D \mid +) = \dfrac{\mathbb{P}(+ \mid D)\mathbb{P}(D)}{\mathbb{P}(+)}$. The evidence $\mathbb{P}(+)$ splits over the two ways to test positive (truly sick, or a false alarm), using $\mathbb{P}(+ \mid \neg D) = 1 - 0.95 = 0.05$:

$$\mathbb{P}(+) = 0.99(0.02) + 0.05(0.98) = 0.0198 + 0.0490 = 0.0688.$$

Therefore

$$\mathbb{P}(D \mid +) = \frac{0.0198}{0.0688} \approx 0.288.$$

Even after a positive on a very accurate test, the patient is only about 29% likely to be sick, because the disease is rare: the $0.98 \times 0.05$ false alarms nearly outnumber the $0.02 \times 0.99$ true detections. This base-rate effect is why screening a low-risk population produces mostly false positives.

### Problem 3.2: A second positive test

**Problem.** The patient from Problem 3.1 takes a second independent test and again tests positive. Update $\mathbb{P}(D \mid ++)$.

**Solution.** Bayesian updating is sequential: yesterday's posterior is today's prior. The new prior is $\mathbb{P}(D) = 0.288$, and we apply Bayes again with the same likelihoods:

$$\mathbb{P}(D \mid ++) = \frac{0.99(0.288)}{0.99(0.288) + 0.05(0.712)} = \frac{0.2851}{0.2851 + 0.0356} = \frac{0.2851}{0.3207} \approx 0.889.$$

Two independent positives push the probability from 29% to about 89%. Evidence compounds: each positive multiplies the odds in favor of disease by the likelihood ratio $0.99/0.05 \approx 20$, so the rare disease becomes probable once corroborated.

## Problem 3.3: Total probability with two urns

**Problem.** Urn $A$ (chosen with probability 0.7) holds 3 red and 7 blue balls; urn $B$ (probability 0.3) holds 6 red and 4 blue. A ball is drawn at random and is red. What is $\mathbb{P}(A \mid \text{red})$?

**Solution.** First the total probability of red, summing over which urn was chosen:

$$\mathbb{P}(\text{red}) = \mathbb{P}(\text{red} \mid A)\mathbb{P}(A) + \mathbb{P}(\text{red} \mid B)\mathbb{P}(B) = 0.3(0.7) + 0.6(0.3) = 0.21 + 0.18 = 0.39.$$

Then invert with Bayes:

$$\mathbb{P}(A \mid \text{red}) = \frac{0.21}{0.39} \approx 0.538.$$

Urn $A$ is the more likely source (0.7 prior), but its lower red fraction pulls the posterior down to 54%. The prior favors $A$ while the likelihood favors $B$; Bayes weighs the two.

## Problem 3.4: Expectation and variance of a discrete variable

**Problem.** A spinner shows $1, 3, 5$ with probabilities $0.5, 0.3, 0.2$. Compute $\mathbb{E}[X]$ and $\text{Var}(X)$.

**Solution.** The expectation is the probability-weighted average:

$$\mathbb{E}[X] = 1(0.5) + 3(0.3) + 5(0.2) = 0.5 + 0.9 + 1.0 = 2.4.$$

For the variance, take the second moment $\mathbb{E}[X^2] = 1(0.5) + 9(0.3) + 25(0.2) = 0.5 + 2.7 + 5.0 = 8.2$ and subtract the squared mean:

$$\text{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = 8.2 - 2.4^2 = 8.2 - 5.76 = 2.44,$$

so the standard deviation is $\sqrt{2.44} \approx 1.56$. The mean 2.4 is the balance point of the three weighted outcomes, and the spread of about 1.56 says a spin typically lands within one and a half of that center.

## Problem 3.5: Binomial mean and variance

**Problem.** Let $X \sim \text{Binomial}(20, 0.25)$. Find $\mathbb{E}[X]$, $\text{Var}(X)$, and the standard deviation.

**Solution.** A binomial counts the successes in $n$ independent Bernoulli trials, so its mean and variance add over the trials:

$$\mathbb{E}[X] = np = 20(0.25) = 5, \qquad \text{Var}(X) = np(1-p) = 20(0.25)(0.75) = 3.75,$$

and the standard deviation is $\sqrt{3.75} \approx 1.94$. On average 5 of the 20 trials succeed, give or take about two. The variance is largest at $p = \frac{1}{2}$ and shrinks toward the extremes, where outcomes become predictable.

## Problem 3.6: Maximum likelihood for a coin

**Problem.** A coin lands heads 7 times in 10 flips. Find the maximum likelihood estimate of the bias $\mu$.

**Solution.** The likelihood of $m = 7$ heads in $n = 10$ flips is $\mu^7(1-\mu)^3$. Maximize its logarithm $\ell(\mu) = 7\ln\mu + 3\ln(1-\mu)$:

$$\ell'(\mu) = \frac{7}{\mu} - \frac{3}{1-\mu} = 0 \implies 7(1-\mu) = 3\mu \implies \hat{\mu} = \frac{7}{10} = 0.7.$$

The MLE is the sample frequency $m/n$. The second derivative is negative throughout $(0, 1)$, confirming a maximum. With only ten flips this estimate is noisy and assigns zero probability to events it has not seen, the shortcoming the Bayesian prior repairs next.

### Problem 3.7: A Beta–Bernoulli posterior

**Problem.** Place a Beta$(2,2)$ prior on the coin's bias and observe the 7 heads, 3 tails of Problem 3.6. Give the posterior, its mean, and the MAP estimate.

**Solution.** The Beta is conjugate to the Bernoulli, so the posterior just adds the counts to the prior parameters as pseudo-observations:

$$\text{Beta}(a+m,\ b+n-m) = \text{Beta}(2+7,\ 2+3) = \text{Beta}(9,5).$$

Its mean is $\dfrac{a'}{a'+b'} = \dfrac{9}{14} \approx 0.643$, and the MAP (the mode) is $\dfrac{a'-1}{a'+b'-2} = \dfrac{8}{12} \approx 0.667$. Both sit below the MLE of 0.7, pulled toward the prior's center of 0.5. The prior's two phantom heads and two phantom tails act as a smoothing cushion, which matters most when data is scarce and vanishes as $n$ grows.

### Problem 3.8: Law of total probability

**Problem.** On 30% of mornings there is heavy traffic. The probability of being late is 0.6 with traffic and 0.1 without. Find the overall probability of being late, and the probability there was traffic given you were late.

**Solution.** Total probability sums over the two traffic states:

$$\mathbb{P}(\text{late}) = 0.6(0.3) + 0.1(0.7) = 0.18 + 0.07 = 0.25.$$

Inverting with Bayes,

$$\mathbb{P}(\text{traffic} \mid \text{late}) = \frac{0.6(0.3)}{0.25} = \frac{0.18}{0.25} = 0.72.$$

Being late raises the probability of traffic from the 30% base rate to 72%, because lateness is far more likely under traffic. This is the same two-step pattern as the urn and disease problems: marginalize to get the evidence, then divide to invert.

## 14.2 Distributions, covariance, and the Gaussian

### Problem 3.9: Multinomial maximum likelihood

**Problem.** A die-like categorical variable with three faces is rolled 100 times, giving counts $(30, 50, 20)$. Find the MLE of the face probabilities.

**Solution.** Maximizing the multinomial log-likelihood $\sum_k m_k \ln p_k$ subject to $\sum_k p_k = 1$ with a Lagrange multiplier gives the intuitive answer: each probability is its observed frequency,

$$\hat{p}_k = \frac{m_k}{N} \implies \hat{\boldsymbol{p}} = \left(\tfrac{30}{100}, \tfrac{50}{100}, \tfrac{20}{100}\right) = (0.3, 0.5, 0.2).$$

The constraint is what forbids simply setting each $p_k$ as large as possible; the multiplier ties them together so they sum to one. As with the coin, the MLE is the empirical frequency.

### Problem 3.10: Sample covariance and correlation

**Problem.** For the four points $(1,1), (2,3), (3,2), (4,4)$, compute the sample covariance matrix (dividing by $n$) and the correlation coefficient.

**Solution.** The mean is $(\bar{x}, \bar{y}) = (2.5, 2.5)$, so the centered points are $(-1.5,-1.5), (-0.5, 0.5), (0.5,-0.5), (1.5, 1.5)$. Then

$$\text{Var}(x) = \tfrac{2.25+0.25+0.25+2.25}{4} = 1.25, \quad \text{Var}(y) = 1.25,$$

$$\text{Cov}(x,y) = \tfrac{(-1.5)(-1.5)+(-0.5)(0.5)+(0.5)(-0.5)+(1.5)(1.5)}{4} = \tfrac{2.25-0.25-0.25+2.25}{4} = 1.$$

So $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 1.25 & 1 \\ 1 & 1.25 \end{smallmatrix}\right]$ and the correlation is

$$\rho = \frac{\text{Cov}(x,y)}{\sqrt{\text{Var}(x)\,\text{Var}(y)}} = \frac{1}{1.25} = 0.8,$$

a strong positive association: the points trend upward together but not perfectly.

## Problem 3.11: From covariance to correlation

**Problem.** Two variables have covariance matrix $\left[\begin{smallmatrix} 9 & 6 \\ 6 & 16 \end{smallmatrix}\right]$. Find their correlation coefficient.

**Solution.** The diagonal gives the variances $\sigma_X^2 = 9$ and $\sigma_Y^2 = 16$, so $\sigma_X = 3$, $\sigma_Y = 4$. The off-diagonal is the covariance 6, so

$$\rho = \frac{6}{3 \cdot 4} = \frac{6}{12} = 0.5.$$

The correlation strips out the units: although the raw covariance is 6, the moderate value $\rho = 0.5$ shows only a middling linear relationship once each variable is standardized by its own spread.

## Problem 3.12: A Gaussian under an affine map

**Problem.** Let $X \sim \mathcal{N}(2, 9)$ and $Y = -2X + 1$. Find the distribution of $Y$.

**Solution.** An affine function of a Gaussian is Gaussian. The mean transforms by the same affine map, $\mathbb{E}[Y] = -2(2) + 1 = -3$, while the variance picks up the square of the multiplier, $\mathrm{Var}(Y) = (-2)^2 \mathrm{Var}(X) = 4(9) = 36$. So

$$Y \sim \mathcal{N}(-3,\ 36),$$

with standard deviation 6. The constant $+1$ shifts the center but never the spread, and the factor $-2$ both flips and doubles the scale, which is why the variance quadruples.

## Problem 3.13: A sum of independent Gaussians

**Problem.** Let $X \sim \mathcal{N}(1, 2)$ and $Y \sim \mathcal{N}(3, 5)$ be independent. Find the distribution of $X + Y$.

**Solution.** Independent Gaussians add to a Gaussian, with means and variances each summing:

$$X + Y \sim \mathcal{N}(1 + 3,\ 2 + 5) = \mathcal{N}(4,\ 7).$$

The standard deviation is $\sqrt{7} \approx 2.65$, not $\sqrt{2} + \sqrt{5} \approx 3.65$: independent spreads combine in quadrature, not by simple addition. Independence is essential; with correlation a cross term $2\,\mathrm{Cov}(X, Y)$ would appear in the variance.

## Problem 3.14: Gaussian conditioning

**Problem.** For a zero-mean joint Gaussian with covariance $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 9 & 3 \\ 3 & 4 \end{smallmatrix}\right]$, find $\mathbb{E}[Y \mid X = x]$ and $\mathrm{Var}(Y \mid X)$.

**Solution.** The conditional of a joint Gaussian is Gaussian with

$$\mathbb{E}[Y \mid X = x] = \frac{\Sigma_{YX}}{\Sigma_{XX}}\, x = \frac{3}{9}\, x = \tfrac{1}{3}x, \qquad \mathrm{Var}(Y \mid X) = \Sigma_{YY} - \frac{\Sigma_{YX}^2}{\Sigma_{XX}} = 4 - \frac{9}{9} = 3.$$

The conditional mean is linear in $x$ with slope $\frac{1}{3}$, which is exactly the least-squares regression coefficient, and conditioning on $X$ shrinks $Y$'s variance from 4 to 3, the uncertainty that knowing $X$ removes.

## Problem 3.15: Mahalanobis distance

**Problem.** For a zero-mean Gaussian with $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 4 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, find the squared Mahalanobis distance of the point $\boldsymbol{x} = (2, 1)$.

**Solution.** The squared Mahalanobis distance is $\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x}$. With $\boldsymbol{\Sigma}$ diagonal, $\boldsymbol{\Sigma}^{-1} = \mathrm{diag}(\frac{1}{4}, 1)$, so

$$\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x} = \frac{2^2}{4} + \frac{1^2}{1} = 1 + 1 = 2.$$

Each coordinate is divided by its own variance before squaring, so the distance counts standard deviations, not raw units: the point is 1 standard deviation out along the wide $x$-axis (variance 4) and 1 along the narrow $y$-axis (variance 1). Mahalanobis distance is what makes the Gaussian's level sets ellipses rather than circles.

### Problem 3.16: A Poisson probability

**Problem.** Calls arrive at rate $\lambda = 4$ per hour, Poisson distributed. Find the probability of exactly 2 calls in an hour.

**Solution.** The Poisson law gives

$$\mathbb{P}(X = 2) = \frac{\lambda^2 e^{-\lambda}}{2!} = \frac{4^2 e^{-4}}{2} = 8e^{-4} \approx 8(0.0183) = 0.147.$$

About a 15% chance. The Poisson's mean and variance both equal $\lambda = 4$, so the typical count is 4 give or take 2, and seeing only 2 calls is a mild below-average hour.

### Problem 3.17: A Dirichlet posterior

**Problem.** With a uniform Dirichlet$(1, 1, 1)$ prior on three categories and observed counts $(4, 2, 4)$, find the posterior and its mean.

**Solution.** The Dirichlet is conjugate to the multinomial, so the posterior adds the counts to the prior parameters:

$$\text{Dirichlet}(1 + 4,\ 1 + 2,\ 1 + 4) = \text{Dirichlet}(5, 3, 5).$$

Its mean is each parameter over their sum $5 + 3 + 5 = 13$:

$$\left(\tfrac{5}{13}, \tfrac{3}{13}, \tfrac{5}{13}\right) \approx (0.385, 0.231, 0.385).$$

Compared with the raw frequencies $(0.4, 0.2, 0.4)$, the uniform prior nudges every estimate gently toward $\frac{1}{3}$, the multivariate version of the Beta's smoothing.

## 14.3 Concentration and learning theory

### Problem 3.18: Markov's inequality

**Problem.** A nonnegative random variable has mean 5. Bound $\mathbb{P}(X \geq 20)$.

**Solution.** Markov's inequality, for $X \geq 0$ and $a > 0$, says $\mathbb{P}(X \geq a) \leq \mathbb{E}[X]/a$:

$$\mathbb{P}(X \geq 20) \leq \frac{5}{20} = 0.25.$$

At most a 25% chance of reaching four times the mean. This bound uses only the mean and nonnegativity, so it is weak but completely distribution-free; with knowledge of the variance, Chebyshev sharpens it.

### Problem 3.19: Chebyshev's inequality

**Problem.** A variable has mean 100 and standard deviation 10. Bound the probability it deviates from 100 by at least 30.

**Solution.** Chebyshev's inequality says $\mathbb{P}(|X - \mu| \geq k\sigma) \leq 1/k^2$, here with $k\sigma = 30$, i.e. $k = 3$:

$$\mathbb{P}(|X - 100| \geq 30) \leq \frac{\sigma^2}{30^2} = \frac{100}{900} = \frac{1}{9} \approx 0.111.$$

At most 11% of the mass lies three or more standard deviations out, again for any distribution with this variance. For a Gaussian the true tail is far smaller (about 0.3%), which is the gap a distribution-specific bound like Hoeffding closes.

## Problem 3.20: Hoeffding sample complexity

**Problem.** How many independent bounded samples guarantee that the sample mean is within $\varepsilon = 0.05$ of the true mean with probability at least 0.95?

**Solution.** Hoeffding's two-sided bound is $\mathbb{P}(|\bar{X} - \mu| \geq \varepsilon) \leq 2e^{-2n\varepsilon^2}$. Setting the right side to $\delta = 0.05$ and solving for $n$,

$$n \geq \frac{\ln(2/\delta)}{2\varepsilon^2} = \frac{\ln(40)}{2(0.05)^2} = \frac{3.689}{0.005} \approx 738.$$

So about 738 samples suffice. The accuracy demand $\varepsilon$ enters as $1/\varepsilon^2$, so halving the tolerance quadruples the sample size, while the confidence $\delta$ enters only logarithmically and is cheap to improve.

## Problem 3.21: A Chernoff bound

**Problem.** A fair coin is flipped 1000 times. Use the Chernoff bound to bound $\mathbb{P}(X \geq 600)$, where $X$ is the number of heads.

**Solution.** The Chernoff bound for a sum of Bernoulli($p$) variables gives $\mathbb{P}(\bar{X} \geq q) \leq e^{-n\,\mathrm{KL}(q\,\|\,p)}$, where the binary relative entropy is $\mathrm{KL}(q\|p) = q\ln\frac{q}{p} + (1-q)\ln\frac{1-q}{1-p}$. With $q = 0.6$, $p = 0.5$,

$$\mathrm{KL}(0.6\|0.5) = 0.6\ln\tfrac{0.6}{0.5} + 0.4\ln\tfrac{0.4}{0.5} \approx 0.0201,$$

so

$$\mathbb{P}(X \geq 600) \leq e^{-1000(0.0201)} = e^{-20.1} \approx 1.8 \times 10^{-9}.$$

The exponential decay in $n$ makes this astronomically smaller than the Chebyshev or Markov bounds would give: at 1000 flips, 600 heads is essentially impossible for a fair coin.

## Problem 3.22: VC dimension of intervals

**Problem.** Let $\mathcal{H}$ be the class of closed intervals $[a, b]$ on the real line, labeling a point 1 if it lies inside. Find $d_{\mathrm{VC}}(\mathcal{H})$.

**Solution.** Two points can be shattered: an interval can include both, exclude both, include only the left (put $[a, b]$ around it alone), or include only the right, realizing all four labelings, so $d_{\mathrm{VC}} \geq 2$. Three points $x_1 < x_2 < x_3$ cannot be shattered: the labeling $(1, 0, 1)$, including the outer two but not the middle, is impossible, since any interval containing $x_1$ and $x_3$ must contain everything between, including $x_2$. Hence no set of three is shattered and $d_{\mathrm{VC}}(\mathcal{H}) = 2$. The VC dimension counts the effective parameters here, the two endpoints.

## Problem 3.23: The union bound and multiple comparisons

**Problem.** You test 5 models, each with at most a 2% chance of looking good by luck. Bound the probability that at least one looks good by luck.

**Solution.** The union bound caps the probability of a union by the sum of probabilities, with no independence needed:

$$\mathbb{P}\Big(\bigcup_{i=1}^{5} E_i\Big) \leq \sum_{i=1}^{5} \mathbb{P}(E_i) \leq 5(0.02) = 0.10.$$

At most a 10% chance some model fools you. Testing more hypotheses inflates the chance of a false discovery, which is the multiple-comparisons problem and the reason a finite hypothesis class of size $K$ pays a $\ln K$ price in its generalization bound.

### Problem 3.24: The bias–variance decomposition

**Problem.** An estimator has bias 0.2, variance 0.05, and the data carries irreducible noise of variance 0.1. Find the expected squared error.

**Solution.** The mean squared error splits into three nonnegative pieces:

$$\text{MSE} = \text{bias}^2 + \text{variance} + \text{noise} = 0.2^2 + 0.05 + 0.1 = 0.04 + 0.05 + 0.10 = 0.19.$$

The bias enters squared, so 0.2 of bias costs 0.04. Lowering model complexity raises bias but lowers variance, and the reverse for a richer model; the sweet spot minimizes their sum. The noise term 0.1 is irreducible: no estimator can beat it.

### Problem 3.25: A confidence interval from the CLT

**Problem.** From $n = 100$ samples with sample mean 50 and known population standard deviation 20, give a 95% confidence interval for the true mean.

**Solution.** The central limit theorem makes the sample mean approximately Gaussian with standard error $\sigma/\sqrt{n}$:

$$\text{SE} = \frac{20}{\sqrt{100}} = 2.$$

A 95% interval reaches 1.96 standard errors either side of the sample mean:

$$50 \pm 1.96(2) = 50 \pm 3.92 = [46.08,\ 53.92].$$

The width shrinks as $1/\sqrt{n}$, so quadrupling the sample halves the interval. This is the law of large numbers made quantitative: more data sharpens the estimate at a predictable rate.

### Problem 3.26: Entropy of a distribution

**Problem.** Compute the entropy (in bits) of the distribution $\boldsymbol{p} = (0.5, 0.25, 0.25)$.

**Solution.** Entropy is $H(\boldsymbol{p}) = -\sum_k p_k \log_2 p_k$, the average number of bits to encode an outcome:

$$H = -\big(0.5\log_2 0.5 + 0.25\log_2 0.25 + 0.25\log_2 0.25\big) = -\big(0.5(-1) + 0.25(-2) + 0.25(-2)\big) = 0.5 + 0.5 + 0.5 = 1.5 \text{ bits}.$$

This sits between the 0 bits of a certain outcome and the $\log_2 3 \approx 1.58$ bits of a uniform distribution over three values; the slight imbalance toward the first outcome makes $\boldsymbol{p}$ a touch more predictable than uniform, so it carries a little less than the maximum entropy. For a two-outcome variable the entropy traces the curve of Fig. 14.1.

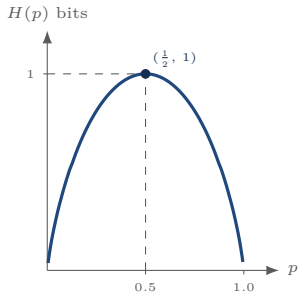



**Figure 14.1.** The binary entropy $H(p) = -p\log_2 p - (1-p)\log_2(1-p)$ of a two-outcome variable. Uncertainty is greatest at $p = \frac{1}{2}$, one full bit, and vanishes at the certain extremes $p = 0$ and $p = 1$. The shape mirrors the Bernoulli variance $p(1-p)$, which peaks at the same place: a fair coin is the hardest to predict and the most variable.

## 14.4 Further probability practice

### Problem 3.27: Mean of a geometric distribution

**Problem.** Trials succeed independently with probability $p = 0.2$. What is the expected number of trials until the first success?

**Solution.** The number of trials until the first success is geometric, with $\mathbb{P}(X = k) = (1-p)^{k-1}p$. Its mean is $\mathbb{E}[X] = 1/p$, which one can derive from the recursion $\mathbb{E}[X] = 1 + (1-p)\mathbb{E}[X]$ (one trial always happens; with probability $1-p$ we start over). Solving, $p\mathbb{E}[X] = 1$, so

$$\mathbb{E}[X] = \frac{1}{p} = \frac{1}{0.2} = 5.$$

On average it takes five trials. The lower the success probability, the longer the wait, inversely; this is the distribution of "how many attempts until it works," from coin flips to retries on a flaky network.

### Problem 3.28: The exponential distribution

**Problem.** Waiting times are exponential with rate $\lambda = 2$ per hour. Find the mean wait and $\mathbb{P}(X > 1)$.

**Solution.** The exponential density is $\lambda e^{-\lambda x}$ for $x \geq 0$, with mean $1/\lambda$:

$$\mathbb{E}[X] = \frac{1}{\lambda} = \frac{1}{2} \text{ hour}, \qquad \mathbb{P}(X > 1) = e^{-\lambda \cdot 1} = e^{-2} \approx 0.135.$$

So the average wait is half an hour, and there is about a 13.5% chance of waiting more than an hour. The exponential is memoryless, $\mathbb{P}(X > s + t \mid X > s) = \mathbb{P}(X > t)$: having waited already tells you nothing about the remaining wait, the continuous cousin of the geometric and the model for inter-arrival times in a Poisson process.

### Problem 3.29: Variance of a sum with covariance

**Problem.** Two variables have $\mathrm{Var}(X) = 4$, $\mathrm{Var}(Y) = 9$, and $\mathrm{Cov}(X, Y) = 2$. Find $\mathrm{Var}(X + Y)$.

**Solution.** The variance of a sum carries a cross term:

$$\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\,\mathrm{Cov}(X, Y) = 4 + 9 + 2(2) = 17.$$

The positive covariance inflates the sum's variance beyond the $4 + 9 = 13$ that independence would give: when $X$ and $Y$ tend to move together, their sum swings wider. Had the covariance been $-2$, the variance would have dropped to 9, the basis of diversification, where negatively correlated assets reduce total risk.

### Problem 3.30: Variance of the sample mean

**Problem.** A measurement has variance $\sigma^2 = 16$. You average $n = 4$ independent measurements. What is the variance of the average, and the standard error?

**Solution.** For independent measurements the variance of the mean is $\sigma^2/n$:

$$\mathrm{Var}(\bar{X}) = \frac{\sigma^2}{n} = \frac{16}{4} = 4, \qquad \text{standard error} = \sqrt{4} = 2.$$

Averaging four measurements quarters the variance and halves the standard error from 4 to 2. The $1/n$ law is why more data sharpens an estimate, and the $1/\sqrt{n}$ on the standard error is the rate: to halve the error you need four times the data, the same scaling behind confidence intervals and the law of large numbers.

### Problem 3.31: Bayes with three hypotheses

**Problem.** Three machines make a part: $A$ (50% of output, 2% defective), $B$ (30%, 3%), $C$ (20%, 5%). A part is defective. Find $\mathbb{P}(A \mid \text{def})$ and $\mathbb{P}(C \mid \text{def})$.

**Solution.** Total probability of a defect sums over the three sources:

$$\mathbb{P}(\text{def}) = 0.5(0.02) + 0.3(0.03) + 0.2(0.05) = 0.010 + 0.009 + 0.010 = 0.029.$$

Then Bayes for each machine:

$$\mathbb{P}(A \mid \text{def}) = \frac{0.010}{0.029} \approx 0.345, \qquad \mathbb{P}(C \mid \text{def}) = \frac{0.010}{0.029} \approx 0.345.$$

Machines $A$ and $C$ are equally likely culprits despite very different sizes: $A$ makes many parts but rarely fails, while $C$ makes few but fails often, and the two effects ($0.5 \times 0.02$ versus $0.2 \times 0.05$) exactly cancel. Bayes weighs volume against defect rate.

### Problem 3.32: Covariance from a joint table

**Problem.** Discrete $X, Y \in \{1, 2\}$ have joint probabilities $\mathbb{P}(1,1) = 0.3$, $\mathbb{P}(1,2) = 0.2$, $\mathbb{P}(2,1) = 0.1$, $\mathbb{P}(2,2) = 0.4$. Find $\operatorname{Cov}(X, Y)$.

**Solution.** Compute the needed expectations. The marginals give $\mathbb{E}[X] = 1(0.3 + 0.2) + 2(0.1 + 0.4) = 0.5 + 1.0 = 1.5$ and $\mathbb{E}[Y] = 1(0.3 + 0.1) + 2(0.2 + 0.4) = 0.4 + 1.2 = 1.6$. The cross moment is

$$\mathbb{E}[XY] = 1 \cdot 1(0.3) + 1 \cdot 2(0.2) + 2 \cdot 1(0.1) + 2 \cdot 2(0.4) = 0.3 + 0.4 + 0.2 + 1.6 = 2.5.$$

Therefore $\operatorname{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] = 2.5 - 1.5(1.6) = 2.5 - 2.4 = 0.1$. The small positive covariance reflects the heavy $\mathbb{P}(2,2) = 0.4$ corner: large $X$ and large $Y$ occur together a little more than independence would predict.

### Problem 3.33: A sum of Poissons

**Problem.** Independent Poisson counts have rates $\lambda_1 = 2$ and $\lambda_2 = 3$. What is the distribution of their sum, and $\mathbb{P}(\text{sum} = 0)$?

**Solution.** Independent Poissons add to a Poisson whose rate is the sum of the rates: $X_1 + X_2 \sim \text{Poisson}(2 + 3) = \text{Poisson}(5)$. The probability of zero total is

$$\mathbb{P}(\text{sum} = 0) = e^{-5}\frac{5^0}{0!} = e^{-5} \approx 0.0067.$$

Only about a 0.7% chance of no events when five are expected. The additivity of rates is why Poisson processes merge cleanly: combining two independent streams gives a third Poisson stream at the combined rate.

### Problem 3.34: A $z$-score

**Problem.** A score of 85 comes from a distribution with mean 70 and standard deviation 10. Find its $z$-score and interpret it.

**Solution.** The $z$-score standardizes by subtracting the mean and dividing by the standard deviation:

$$z = \frac{x - \mu}{\sigma} = \frac{85 - 70}{10} = 1.5.$$

The score sits 1.5 standard deviations above the mean. Standardization removes units and recenters, so a $z$ of 1.5 is comparable across any distribution; under a Gaussian it corresponds to roughly the 93rd percentile. This is the same rescaling that turns a general Gaussian into the standard $\mathcal{N}(0, 1)$ and that whitening applies coordinatewise.

## Problem 3.35: A KL divergence

**Problem.** Compute the Kullback–Leibler divergence $\mathrm{KL}(p \,\|\, q)$ between two Bernoullis with $p = 0.7$ and $q = 0.5$ (in nats).

**Solution.** For Bernoullis, $\mathrm{KL}(p\|q) = p \ln \frac{p}{q} + (1-p) \ln \frac{1-p}{1-q}$:

$$\mathrm{KL}(0.7\|0.5) = 0.7 \ln \frac{0.7}{0.5} + 0.3 \ln \frac{0.3}{0.5} = 0.7(0.3365) + 0.3(-0.5108) \approx 0.0823.$$

The divergence is positive (it is zero only when $p = q$) and asymmetric in general. It measures the extra nats needed to encode samples from $p$ using a code built for $q$, and it is the quantity the Chernoff bound's exponent uses and that maximum likelihood implicitly minimizes between the data and the model.

## Problem 3.36: Maximum likelihood for a Poisson rate

**Problem.** Counts $\{3, 5, 4\}$ are observed from a Poisson. Find the MLE of the rate $\lambda$.

**Solution.** The Poisson log-likelihood for counts $x_i$ is $\ell(\lambda) = \sum_i (x_i \ln \lambda - \lambda)$ (dropping constants). Differentiating, $\ell'(\lambda) = \sum_i (x_i/\lambda - 1) = 0$ gives $\lambda = \frac{1}{n} \sum_i x_i$, the sample mean:

$$\hat{\lambda} = \frac{3 + 5 + 4}{3} = 4.$$

As for the Bernoulli and multinomial, the Poisson MLE is just the empirical average. The second derivative $-\sum_i x_i/\lambda^2 < 0$ confirms a maximum, and since the Poisson mean equals $\lambda$, estimating the rate by the sample mean is the natural moment match.

## Problem 3.37: A confidence interval's width

**Problem.** With known $\sigma = 15$ and $n = 25$ samples, what is the half-width of a 95% confidence interval for the mean?

**Solution.** The half-width is 1.96 standard errors, and the standard error is $\sigma/\sqrt{n}$:

$$1.96 \, \frac{\sigma}{\sqrt{n}} = 1.96 \, \frac{15}{\sqrt{25}} = 1.96 \, \frac{15}{5} = 1.96(3) = 5.88.$$

So the interval reaches 5.88 either side of the sample mean. To halve this width you would need four times the data ($n = 100$), the $1/\sqrt{n}$ law again. The factor 1.96 is the 97.5th percentile of the standard Gaussian, capturing the central 95%.

## Problem 3.38: Where the Bernoulli variance is largest

**Problem.** The variance of a Bernoulli is $p(1-p)$. Show it is maximized at $p = \frac{1}{2}$ and give the maximum.

**Solution.** Differentiate $v(p) = p(1-p) = p - p^2$: $v'(p) = 1 - 2p = 0$ gives $p = \frac{1}{2}$, and $v''(p) = -2 < 0$ confirms a maximum. The maximum variance is $\frac{1}{2}(1 - \frac{1}{2}) = \frac{1}{4} = 0.25$. A fair coin is the most uncertain Bernoulli; as $p$ moves toward 0 or 1 the outcome becomes predictable and the variance falls to 0. This is why $p = \frac{1}{2}$ is the hardest bias to estimate and why balanced classes carry the most label entropy.

CHAPTER 15

# Problem Set 4: Vector Calculus and Optimization

This set covers Chapters 8 and 9: gradients, Jacobians, the chain rule and backpropagation, Hessians and convexity, gradient descent and Newton's method, then linear programming, Lagrange multipliers, duality, and the KKT conditions. Every derivative is taken componentwise, every descent step traced on numbers, and every optimum checked against the conditions that certify it.

## 15.1 Gradients, Jacobians, and the chain rule

### Problem 4.1: A gradient

**Problem.** Find the gradient of $f(x, y) = x^2 + 3xy + 2y^2$ and evaluate it at $(1, 2)$.

**Solution.** Differentiate with respect to each variable, holding the other fixed:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) = (2x + 3y,\ 3x + 4y).$$

At $(1, 2)$ this is $(2 + 6,\ 3 + 8) = (8, 11)$. The gradient points in the direction of steepest ascent, so $f$ increases fastest from $(1, 2)$ along $(8, 11)$, and the negative gradient $(-8, -11)$ is the direction gradient descent would step.

### Problem 4.2: A Jacobian

**Problem.** For $\boldsymbol{F}(x, y) = (x^2 y,\ x + y^2)$, compute the Jacobian and evaluate it at $(2, 1)$.

**Solution.** The Jacobian stacks the gradients of each output as rows:

$$\boldsymbol{J} = \begin{bmatrix} \partial(x^2y)/\partial x & \partial(x^2y)/\partial y \\ \partial(x + y^2)/\partial x & \partial(x + y^2)/\partial y \end{bmatrix} = \begin{bmatrix} 2xy & x^2 \\ 1 & 2y \end{bmatrix}.$$

At $(2, 1)$ this is $\left[\begin{smallmatrix} 4 & 4 \\ 1 & 2 \end{smallmatrix}\right]$. The Jacobian is the best linear approximation of $\boldsymbol{F}$ near the point; its determinant $4 \cdot 2 - 4 \cdot 1 = 4$ is the local area-scaling factor of the map.

### Problem 4.3: The chain rule

**Problem.** Let $h(x) = \ln(x^2 + 1)$. Find $h'(x)$ and $h'(2)$.

**Solution.** Write $h = f(g(x))$ with $g(x) = x^2 + 1$ (inner) and $f(u) = \ln u$ (outer). The chain rule multiplies the derivatives, outer evaluated at the inner times inner:

$$h'(x) = \frac{1}{g(x)} \cdot g'(x) = \frac{2x}{x^2 + 1}, \qquad h'(2) = \frac{4}{5}.$$

This is backpropagation in miniature: the derivative flows from the outer ln back through the inner square, each layer contributing one factor.

## Problem 4.4: A critical point and the Hessian test

**Problem.** Find and classify the critical point of $f(x, y) = x^2 - xy + y^2$.

**Solution.** Set the gradient to zero: $\nabla f = (2x - y,\ -x + 2y) = \mathbf{0}$ gives $y = 2x$ and $x = 2y$, so $x = 4x$ and the only solution is $(0, 0)$. The Hessian is constant,

$$\nabla^2 f = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix},$$

with eigenvalues from $(2 - \lambda)^2 - 1 = 0$, i.e. $\lambda = 3, 1$, both positive. The Hessian is positive definite, so $(0, 0)$ is a strict local (indeed global, since $f$ is convex) *minimum*, where $f = 0$.

## Problem 4.5: Gradient descent on an elliptical bowl

**Problem.** Minimize $f(x, y) = \frac{1}{2}(x^2 + 4y^2)$ by gradient descent with step $\alpha = 0.2$ from $(1, 1)$. Trace four iterations.

**Solution.** The gradient is $\nabla f = (x,\ 4y)$, so the update $(x, y) \leftarrow (x, y) - \alpha(x, 4y)$ multiplies each coordinate by its own factor: $x \leftarrow (1 - 0.2)x = 0.8x$ and $y \leftarrow (1 - 0.8)y = 0.2y$.

$$(1, 1) \to (0.8,\ 0.2) \to (0.64,\ 0.04) \to (0.512,\ 0.008).$$

The steep $y$-direction (curvature 4) collapses fast as $0.2^t$, while the gentle $x$-direction (curvature 1) crawls as $0.8^t$. The slow direction sets the pace, the hallmark of an ill-conditioned bowl, and the reason momentum and preconditioning help (Fig. 15.1).

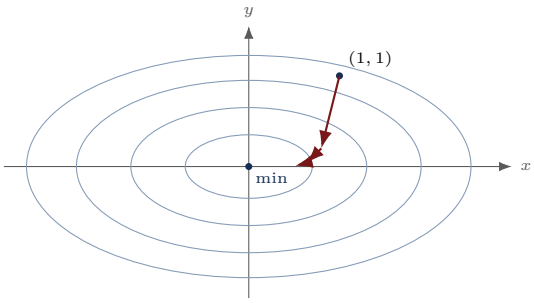



**Figure 15.1.** Gradient descent zig-zags across the elliptical contours of $f = \frac{1}{2}(x^2 + 4y^2)$: the steep $y$-direction is killed in one or two steps, then the iterate creeps along the shallow $x$-axis toward the minimum at the origin.

## Problem 4.6: Newton's method computes a square root

**Problem.** Use Newton's method on $f(x) = x^2 - 2$ from $x_0 = 1$ to compute $\sqrt{2}$. Trace three iterations.

**Solution.** Newton's update is $x_{t+1} = x_t - \dfrac{f(x_t)}{f'(x_t)}$ with $f'(x) = 2x$, which simplifies to the classic averaging iteration

$$x_{t+1} = x_t - \frac{x_t^2 - 2}{2x_t} = \frac{1}{2}\left(x_t + \frac{2}{x_t}\right).$$

From $x_0 = 1$: $x_1 = \frac{1}{2}(1 + 2) = 1.5$; $x_2 = \frac{1}{2}(1.5 + \frac{2}{1.5}) = 1.41667$; $x_3 = \frac{1}{2}(1.41667 + \frac{2}{1.41667}) = 1.41422$, already correct to five digits. Newton doubles the number of correct digits each step (quadratic convergence) once close, because it models $f$ by its tangent line and lands at the tangent's root.

### Problem 4.7: A directional derivative

**Problem.** For $f(x, y) = x^2 + y^2$, find the directional derivative at $(3, 4)$ in the direction of the unit vector $\boldsymbol{u} = (1, 0)$, and the direction of fastest increase.

**Solution.** The gradient is $\nabla f = (2x, 2y) = (6, 8)$ at $(3, 4)$. The directional derivative is the gradient projected onto $\boldsymbol{u}$:

$$D_{\boldsymbol{u}} f = \nabla f^{\top} \boldsymbol{u} = (6, 8)^{\top} (1, 0) = 6.$$

The fastest increase is along the gradient itself, direction $(6, 8)/10 = (0.6, 0.8)$, at the rate $\|\nabla f\| = \sqrt{36 + 64} = 10$. Moving along $(1, 0)$ captures only the $x$-component 6 of that maximal 10.

### Problem 4.8: A convexity check

**Problem.** Is $f(x) = x^4$ convex on $\mathbb{R}$?

**Solution.** A twice-differentiable one-variable function is convex exactly when $f'' \geq 0$ everywhere. Here $f'(x) = 4x^3$ and $f''(x) = 12x^2 \geq 0$ for all $x$, so $f$ is convex. It is not *strictly* convex in the second-derivative sense at $x = 0$ (where $f'' = 0$), yet it is still strictly convex as a function, a reminder that $f'' > 0$ is sufficient but not necessary for strict convexity.

### Problem 4.9: One step of logistic-regression gradient

**Problem.** A logistic model has weights $\boldsymbol{\theta} = \mathbf{0}$. For a single example $\boldsymbol{x} = (1, 2)$ with label $y = 1$, compute the gradient of the cross-entropy loss.

**Solution.** The predicted probability is $p = \sigma(\boldsymbol{\theta}^{\top} \boldsymbol{x}) = \sigma(0) = 0.5$. The cross-entropy gradient for one example is $(p - y)\boldsymbol{x}$:

$$\nabla \ell = (p - y)\boldsymbol{x} = (0.5 - 1)(1, 2) = (-0.5, -1.0).$$

A descent step $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla \ell$ moves the weights in the direction $(0.5, 1.0)$, raising $\boldsymbol{\theta}^{\top} \boldsymbol{x}$ and so raising the predicted probability on this positive example. The "prediction minus target" form $(p - y)$ is the same residual shape as linear regression.

## 15.2 Linear programming and duality

### Problem 4.10: A linear program by vertex enumeration

**Problem.** Maximize $3x + 2y$ subject to $x + y \leq 4$, $x \leq 3$, $y \leq 3$, $x, y \geq 0$.

**Solution.** The feasible region is a polygon, and a linear objective is maximized at a corner. The vertices and objective values are

$$(0, 0) : 0, \quad (3, 0) : 9, \quad (3, 1) : 11, \quad (1, 3) : 9, \quad (0, 3) : 6,$$

where $(3, 1)$ is the intersection of $x = 3$ and $x + y = 4$. The maximum is 11 at $(3, 1)$, where the two constraints $x \leq 3$ and $x + y \leq 4$ are both active. The objective increases in the direction $\boldsymbol{c} = (3, 2)$, and the corner farthest along that direction wins (Fig. 15.2).

### Problem 4.11: A Lagrange multiplier

**Problem.** Minimize $x^2 + y^2$ subject to $x + 2y = 10$.

**Solution.** At the constrained minimum the gradient of the objective is parallel to the gradient of the constraint: $\nabla(x^2 + y^2) = \lambda \nabla(x + 2y)$, i.e. $(2x, 2y) = \lambda(1, 2)$. So $2x = \lambda$ and $2y = 2\lambda$, giving $y = 2x$. Substituting into the constraint $x + 2(2x) = 10$ gives $5x = 10$, so $x = 2$, $y = 4$, and $\lambda = 4$. The minimum value is $2^2 + 4^2 = 20$. Geometrically $(2, 4)$ is the point of the line closest to the origin, the foot of the perpendicular, and $\sqrt{20}$ is that shortest distance.

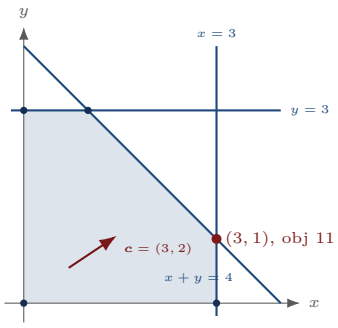



**Figure 15.2.** The feasible polygon (shaded) and the objective direction $\boldsymbol{c} = (3, 2)$. The maximum sits at the vertex $(3, 1)$, the corner farthest along $\boldsymbol{c}$, where two constraints are active.

### Problem 4.12: KKT with an inequality constraint

**Problem.** Minimize $(x-3)^2 + (y-2)^2$ subject to $x + y \le 4$.

**Solution.** First test whether the constraint is active. The unconstrained minimum is $(3, 2)$, but $3 + 2 = 5 > 4$ violates $x + y \le 4$, so the constraint binds and we minimize on the line $x + y = 4$. By KKT, $\nabla f = \mu \nabla(4 - x - y)$ form, or directly project $(3, 2)$ onto $x + y = 4$:

$$(x, y) = (3, 2) - \frac{(3 + 2) - 4}{2}(1, 1) = (3, 2) - \tfrac{1}{2}(1, 1) = (2.5,\ 1.5).$$

The objective value is $(2.5 - 3)^2 + (1.5 - 2)^2 = 0.25 + 0.25 = 0.5$. The multiplier is positive ($\mu = 1 > 0$) and the constraint is tight, so all KKT conditions, stationarity, primal and dual feasibility, and complementary slackness, hold; for this convex problem they certify the global minimum.

### Problem 4.13: The dual of a linear program

**Problem.** Write the dual of the LP in Problem 4.10 and verify strong duality.

**Solution.** The primal is max $3x + 2y$ subject to $x + y \le 4$, $x \le 3$, $y \le 3$. With one dual variable per constraint $(u, v, w \ge 0)$, the dual is

$$\min\ 4u + 3v + 3w \quad \text{s.t.} \quad u + v \ge 3, \quad u + w \ge 2, \quad u, v, w \ge 0.$$

At the primal optimum $(3, 1)$, the constraints $x + y \le 4$ and $x \le 3$ are active while $y \le 3$ is slack, so by complementary slackness $w = 0$ and the first two dual constraints are tight: $u + v = 3$ and $u + w = 2$, giving $u = 2$, $v = 1$. The dual objective is $4(2) + 3(1) + 3(0) = 11$, equal to the primal optimum: strong duality holds. The dual values $u = 2, v = 1$ are the shadow prices of the binding resources.

## 15.3 The optimization landscape and Taylor expansion

### Problem 4.14: Unconstrained quadratic minimization

**Problem.** Minimize $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top A \boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$ with $A = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 4 \end{smallmatrix}\right]$ and $\boldsymbol{b} = (2, 8)$.

**Solution.** The gradient is $\nabla f = A\boldsymbol{x} - \boldsymbol{b}$, and setting it to zero gives the linear system $A\boldsymbol{x} = \boldsymbol{b}$:

$$\begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 2 \\ 8 \end{bmatrix} \implies \boldsymbol{x} = \left(\tfrac{2}{2}, \tfrac{8}{4}\right) = (1, 2).$$

Because $A$ is positive definite (eigenvalues $2, 4 > 0$), $f$ is strictly convex and $(1, 2)$ is the unique global minimum. This is the prototype every least-squares problem reduces to: minimizing a positive-definite quadratic is the same as solving $A\boldsymbol{x} = \boldsymbol{b}$.

### Problem 4.15: When does gradient descent converge?

**Problem.** For $f(x) = \frac{1}{2}x^2$, the gradient-descent update is $x \leftarrow x - \alpha x$. For which step sizes $\alpha$ does it converge, and what happens at $\alpha = 2.5$?

**Solution.** The update is $x_{t+1} = (1-\alpha)x_t$, so $x_t = (1-\alpha)^t x_0$, which decays to zero exactly when $|1-\alpha| < 1$, i.e. $0 < \alpha < 2$. At $\alpha = 0.5$ the factor is 0.5 and the iterate halves each step; at $\alpha = 1$ it reaches the minimum in one step; at $\alpha = 2.5$ the factor is $1 - 2.5 = -1.5$, whose magnitude exceeds 1, so the iterate *diverges*, oscillating with growing amplitude. The stable range $0 < \alpha < 2/L$ (here $L = f'' = 1$) is the bound every quadratic obeys, and overshoot past $2/L$ is the classic "learning rate too large" failure.

### Problem 4.16: The softmax

**Problem.** Compute the softmax of the logits $\boldsymbol{z} = (1, 2, 3)$.

**Solution.** The softmax exponentiates and normalizes: $p_k = e^{z_k} / \sum_j e^{z_j}$. The exponentials are $e^1 = 2.718$, $e^2 = 7.389$, $e^3 = 20.086$, summing to 30.19, so

$$\boldsymbol{p} = \left(\tfrac{2.718}{30.19}, \tfrac{7.389}{30.19}, \tfrac{20.086}{30.19}\right) \approx (0.090,\ 0.245,\ 0.665).$$

The probabilities are positive and sum to 1, and the largest logit gets the largest share. Softmax is the multiclass generalization of the logistic sigmoid, and the gap between 0.245 and 0.665 shows how it sharpens differences through the exponential.

### Problem 4.17: A second-order Taylor expansion

**Problem.** Give the second-order Taylor expansion of $e^x$ about 0 and use it to approximate $e^{0.1}$.

**Solution.** With $f(x) = e^x$, every derivative at 0 is 1, so

$$e^x \approx f(0) + f'(0)x + \tfrac{1}{2}f''(0)x^2 = 1 + x + \tfrac{x^2}{2}.$$

At $x = 0.1$ this gives $1 + 0.1 + 0.005 = 1.105$, against the true $e^{0.1} = 1.10517\ldots$, an error of about $2 \times 10^{-4}$. The quadratic term $x^2/2$ is exactly the curvature correction a Newton step uses; keeping it turns a linear tangent estimate into a far more accurate parabola.

### Problem 4.18: A Hessian in three variables

**Problem.** Find the Hessian of $f(x, y, z) = x^2 + y^2 + z^2 + xy$ and decide whether $f$ is convex.

**Solution.** The Hessian collects all second partials. The only cross term is $xy$, contributing $\partial^2 f / \partial x \partial y = 1$:

$$\nabla^2 f = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

It is constant and symmetric. Its eigenvalues are 2 (from the isolated $z$ block) and the eigenvalues of $\left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, namely 3 and 1. All three eigenvalues $\{3, 2, 1\}$ are positive, so the Hessian is positive definite and $f$ is strictly convex, with its unique minimum at the origin.

### Problem 4.19: A one-dimensional second-derivative test

**Problem.** Find and classify the critical points of $f(x) = x^3 - 3x$.

**Solution.** Set $f'(x) = 3x^2 - 3 = 0$, so $x = \pm 1$. The second derivative is $f''(x) = 6x$: at $x = 1$, $f'' = 6 > 0$, a local *minimum* with $f(1) = -2$; at $x = -1$, $f'' = -6 < 0$, a local *maximum* with $f(-1) = 2$. The cubic rises, dips to a valley at $x = 1$, after cresting a hill at $x = -1$. This is the nonconvex case where a zero derivative alone cannot tell minimum from maximum; the sign of the curvature decides.

### Problem 4.20: The gradient is the direction of steepest ascent

**Problem.** For $f$ with $\nabla f = (3, 4)$ at a point, what is the maximum rate of increase there, and along which unit direction?

**Solution.** The directional derivative $D_{\boldsymbol{u}} f = \nabla f^{\top} \boldsymbol{u} = \|\nabla f\| \cos\vartheta$ is largest when $\boldsymbol{u}$ aligns with the gradient ($\vartheta = 0$). The maximum rate is

$$\|\nabla f\| = \sqrt{3^2 + 4^2} = 5,$$

achieved along the unit gradient $\boldsymbol{u} = (3, 4)/5 = (0.6, 0.8)$. Every other direction gives less, and the perpendicular directions give zero change, which is why contour lines run perpendicular to the gradient.

## 15.4 Further calculus and optimization practice

### Problem 4.21: Gradient of a quadratic form

**Problem.** For the quadratic form $f(\boldsymbol{x}) = \boldsymbol{x}^{\top} A \boldsymbol{x}$ with the symmetric $A = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 3 \end{smallmatrix}\right]$, find $\nabla f$ and evaluate it at $(1, 2)$.

**Solution.** For symmetric $A$, the gradient of $\boldsymbol{x}^{\top} A \boldsymbol{x}$ is $2A\boldsymbol{x}$ (the two appearances of $\boldsymbol{x}$ each contribute $A\boldsymbol{x}$, and symmetry merges them). So

$$\nabla f = 2A\boldsymbol{x} = 2 \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = 2 \begin{bmatrix} 4 \\ 7 \end{bmatrix} = (8, 14).$$

This is the matrix-calculus identity behind the normal equations: differentiating $\|A\boldsymbol{x} - \boldsymbol{b}\|^2$ uses exactly $\nabla(\boldsymbol{x}^{\top} A^{\top} A \boldsymbol{x}) = 2A^{\top} A \boldsymbol{x}$.

### Problem 4.22: A constrained maximum on the circle

**Problem.** Maximize $x + y$ subject to $x^2 + y^2 = 1$.

**Solution.** By Lagrange multipliers, $\nabla(x + y) = \lambda \nabla(x^2 + y^2)$ gives $(1, 1) = \lambda(2x, 2y)$, so $x = y$. The constraint $x^2 + y^2 = 1$ then forces $2x^2 = 1$, i.e. $x = y = \frac{1}{\sqrt{2}}$, and the maximum value is $x + y = \frac{2}{\sqrt{2}} = \sqrt{2} \approx 1.414$. Geometrically the objective $x + y$ has gradient $(1, 1)$, so it is largest where the circle's outward normal points along $(1, 1)$, the point $\frac{1}{\sqrt{2}}(1, 1)$. This is the unit-vector version of the Cauchy–Schwarz bound $\boldsymbol{x}^{\top}(1, 1) \le \|\boldsymbol{x}\|\sqrt{2}$.

### Problem 4.23: Partial derivatives

**Problem.** For $f(x, y, z) = x^2 y + \sin z$, find the three partial derivatives at $(2, 3, 0)$.

**Solution.** Differentiate with respect to each variable, treating the others as constants:

$$\frac{\partial f}{\partial x} = 2xy = 2(2)(3) = 12, \quad \frac{\partial f}{\partial y} = x^2 = 4, \quad \frac{\partial f}{\partial z} = \cos z = \cos 0 = 1.$$

So $\nabla f(2, 3, 0) = (12, 4, 1)$. Each partial isolates the function's sensitivity to one input; stacked together they form the gradient, the direction the function climbs fastest.

### Problem 4.24: A Newton step in two dimensions

**Problem.** Minimize $f(x, y) = x^2 + xy + y^2 - 3x$ with one Newton step from $(0, 0)$.

**Solution.** The gradient is $\nabla f = (2x + y - 3,\ x + 2y)$, so at $(0, 0)$ it is $(-3, 0)$. The Hessian is constant, $\nabla^2 f = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, with inverse $\frac{1}{3}\left[\begin{smallmatrix} 2 & -1 \\ -1 & 2 \end{smallmatrix}\right]$. The Newton step is

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - (\nabla^2 f)^{-1} \nabla f = (0, 0) - \tfrac{1}{3} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} -3 \\ 0 \end{bmatrix} = (0, 0) - \tfrac{1}{3}(-6, 3) = (2, -1).$$

One step reaches the exact minimum $(2, -1)$. For a quadratic, Newton's method converges in a single step, because the local quadratic model it minimizes *is* the function.

### Problem 4.25: A maximum by AM–GM, via Lagrange

**Problem.** Maximize $xy$ subject to $x + y = 10$ with $x, y \geq 0$.

**Solution.** Lagrange gives $\nabla(xy) = \lambda\nabla(x + y)$, i.e. $(y, x) = \lambda(1, 1)$, so $x = y$. The constraint forces $x = y = 5$ and $xy = 25$. This is the arithmetic-mean–geometric-mean inequality in disguise: among numbers with a fixed sum, the product is largest when they are equal, so $\sqrt{xy} \leq \frac{x+y}{2} = 5$ with equality at $x = y = 5$. The same principle explains why a square encloses the most area for a fixed perimeter.

### Problem 4.26: Jensen's inequality

**Problem.** For a random variable $X$ taking values 0 and 2 with equal probability, verify $\mathbb{E}[X^2] \geq \mathbb{E}[X]^2$, an instance of Jensen's inequality for the convex function $g(t) = t^2$.

**Solution.** The mean is $\mathbb{E}[X] = \frac{1}{2}(0) + \frac{1}{2}(2) = 1$, so $\mathbb{E}[X]^2 = 1$. The second moment is $\mathbb{E}[X^2] = \frac{1}{2}(0) + \frac{1}{2}(4) = 2$. Indeed $2 \geq 1$, and the gap $\mathbb{E}[X^2] - \mathbb{E}[X]^2 = 1$ is exactly the variance. Jensen's inequality says $\mathbb{E}[g(X)] \geq g(\mathbb{E}[X])$ for any convex $g$: averaging inside a convex function underestimates the average of the function. Here it is the reason variance is never negative.

### Problem 4.27: Steepest descent direction

**Problem.** At a point where $\nabla f = (2, -2)$, in which unit direction does $f$ decrease fastest, and at what rate?

**Solution.** A function decreases fastest in the direction opposite its gradient. The descent direction is $-\nabla f = (-2, 2)$, normalized to $\frac{1}{2\sqrt{2}}(-2, 2) = \frac{1}{\sqrt{2}}(-1, 1)$. The rate of steepest decrease is $-\|\nabla f\| = -\sqrt{2^2 + (-2)^2} = -2\sqrt{2} \approx -2.83$ per unit step. This is precisely the direction gradient descent moves, and the magnitude $\|\nabla f\|$ is why the method slows as it nears a minimum, where the gradient shrinks to zero.

### Problem 4.28: Convexity from the Hessian

**Problem.** Show that $f(x, y) = x^2 + y^2$ is convex, and state why that makes gradient descent reliable on it.

**Solution.** The Hessian is constant, $\nabla^2 f = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right] = 2I$, with both eigenvalues equal to $2 > 0$, so it is positive definite everywhere and $f$ is (strictly) convex. For a convex function every local minimum is the global minimum and a zero gradient is sufficient for optimality, so gradient descent cannot get trapped: from any start it converges to the unique minimum at the origin. The perfectly round bowl ($\nabla^2 = 2I$, condition number 1) is also the best case for the step size, converging in essentially one step with $\alpha = 1/2$.

### Problem 4.29: Gradient of softmax cross-entropy

**Problem.** For logits $\boldsymbol{z} = (1, 2, 3)$ and the true class 2 (one-hot $\boldsymbol{y} = (0, 1, 0)$), compute the gradient of the softmax cross-entropy loss with respect to $\boldsymbol{z}$.

**Solution.** The cross-entropy gradient with respect to the logits is the clean residual $\boldsymbol{p} - \boldsymbol{y}$, where $\boldsymbol{p}$ is the softmax. From Problem 4.16 (or recomputing), $\boldsymbol{p} = (0.090,\ 0.245,\ 0.665)$, so

$$\nabla_{\boldsymbol{z}} = \boldsymbol{p} - \boldsymbol{y} = (0.090 - 0,\ 0.245 - 1,\ 0.665 - 0) = (0.090,\ -0.755,\ 0.665).$$

The negative entry on the true class 2 means a descent step raises $z_2$, pushing more probability onto the correct class, while the positive entries on classes 1 and 3 pull their logits down. The remarkable simplicity of "predicted minus target," identical in form to linear and logistic regression, is why softmax with cross-entropy is the standard classification head.

## Problem 4.30: Exact line search on a quadratic

**Problem.** Minimizing $f(x, y) = \frac{1}{2}(x^2 + 4y^2)$ from $(1, 1)$, find the step size $\alpha$ that an exact line search takes along the steepest-descent direction.

**Solution.** The gradient at $(1, 1)$ is $\boldsymbol{g} = (x, 4y) = (1, 4)$, and the search direction is $-\boldsymbol{g}$. For a quadratic $\frac{1}{2}\boldsymbol{x}^\top A\boldsymbol{x}$ with $A = \mathrm{diag}(1, 4)$, the exact step minimizing $f(\boldsymbol{x} - \alpha\boldsymbol{g})$ is

$$\alpha^\star = \frac{\boldsymbol{g}^\top \boldsymbol{g}}{\boldsymbol{g}^\top A\boldsymbol{g}} = \frac{1^2 + 4^2}{1 \cdot 1^2 + 4 \cdot 4^2} = \frac{17}{1 + 64} = \frac{17}{65} \approx 0.262.$$

Exact line search picks the step that goes as far as the parabola along $-\boldsymbol{g}$ allows, no tuning required. The catch on an ill-conditioned bowl like this one (curvatures 1 and 4) is that consecutive exact-search directions are orthogonal, so the iterates still zig-zag; this is why conjugate gradients, which correct the directions, converge faster.

## Problem 4.31: An inactive inequality constraint

**Problem.** Minimize $(x - 1)^2 + (y - 1)^2$ subject to $x + y \le 10$.

**Solution.** Check the unconstrained minimum first: it is $(1, 1)$, where the objective is 0. Test feasibility: $1 + 1 = 2 \le 10$, satisfied with room to spare, so the constraint is *inactive*. By KKT complementary slackness, an inactive constraint forces its multiplier to zero ($\mu = 0$), and the problem reduces to the unconstrained one. The minimum is $(1, 1)$ with value 0. The lesson: always test whether the unconstrained optimum is already feasible; if so, the constraint does nothing, and only a binding constraint (as in Problem 4.12) bends the solution.

## Problem 4.32: Gradient of the least-squares objective

**Problem.** For $f(\boldsymbol{x}) = \|A\boldsymbol{x} - \boldsymbol{b}\|^2$ with $A = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$ and $\boldsymbol{b} = (1, 3)$, give $\nabla f$ and evaluate it at $\boldsymbol{x} = \boldsymbol{0}$.

**Solution.** Expanding $f = \boldsymbol{x}^\top A^\top A\boldsymbol{x} - 2\boldsymbol{b}^\top A\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{b}$ and differentiating gives the key identity $\nabla f = 2A^\top(A\boldsymbol{x} - \boldsymbol{b})$. At $\boldsymbol{x} = \boldsymbol{0}$ the residual is $A\boldsymbol{0} - \boldsymbol{b} = -\boldsymbol{b}$, so

$$\nabla f = 2A^\top(-\boldsymbol{b}) = -2\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \end{bmatrix} = -2\begin{bmatrix} 4 \\ 3 \end{bmatrix} = (-8, -6).$$

Setting this to zero recovers the normal equations $A^\top A\boldsymbol{x} = A^\top \boldsymbol{b}$. The gradient points uphill, so a descent step from the origin moves along $(8, 6)$, toward the least-squares solution.

## Problem 4.33: Classifying a critical point

**Problem.** Find and classify the critical point of $f(x, y) = x^2 + y^2 - xy - 3y$.

**Solution.** The gradient is $\nabla f = (2x - y,\ 2y - x - 3)$. Setting it to zero, $y = 2x$ from the first equation, and substituting into $2y - x - 3 = 0$ gives $4x - x - 3 = 0$, so $x = 1$, $y = 2$. The Hessian is constant, $\nabla^2 f = \left[\begin{smallmatrix} 2 & -1 \\ -1 & 2 \end{smallmatrix}\right]$, with eigenvalues 3 and 1, both positive. So $(1, 2)$ is a strict local (and global, by convexity) minimum, where $f = 1 + 4 - 2 - 6 = -3$. The positive-definite Hessian certifies the bowl shape.

## Problem 4.34: Minimizing a norm on a plane

**Problem.** Minimize $x^2 + y^2 + z^2$ subject to $x + y + z = 3$.

**Solution.** By Lagrange, $(2x, 2y, 2z) = \lambda(1, 1, 1)$, so $x = y = z$. The constraint $3x = 3$ gives $x = y = z = 1$, and the minimum value is $1 + 1 + 1 = 3$. The point $(1, 1, 1)$ is the closest point of the plane to the origin, and its distance is $\sqrt{3}$. Symmetry did the work: minimizing a sum of squares under a symmetric linear constraint spreads the budget equally, the same reason the mean (not a lopsided split) minimizes squared error.

### Problem 4.35: One momentum step

**Problem.** Heavy-ball momentum updates $\boldsymbol{v} \leftarrow \mu \boldsymbol{v} - \alpha \nabla f$ then $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{v}$. With $\mu = 0.9$, $\alpha = 0.1$, starting $x = 1$, $v = 0$, and gradient $f'(1) = 2$, take one step.

**Solution.** The velocity updates first: $v = 0.9(0) - 0.1(2) = -0.2$. Then the position: $x = 1 + (-0.2) = 0.8$. On this first step, with $v$ starting at zero, momentum coincides with plain gradient descent (step $-0.2$). Its advantage shows on later steps: the velocity accumulates a $\mu$-weighted memory of past gradients, so in a consistent downhill direction it builds speed (like a ball rolling), accelerating through the shallow valleys that stall vanilla gradient descent.

### Problem 4.36: A second-order Taylor approximation

**Problem.** Give the second-order Taylor expansion of $\cos x$ about 0 and use it to approximate $\cos(0.2)$.

**Solution.** With $\cos 0 = 1$, $-\sin 0 = 0$, $-\cos 0 = -1$, the expansion is

$$\cos x \approx 1 + 0 \cdot x - \tfrac{1}{2}x^2 = 1 - \tfrac{x^2}{2}.$$

At $x = 0.2$: $1 - \frac{0.04}{2} = 1 - 0.02 = 0.98$, against the true $\cos(0.2) = 0.980067\ldots$, an error near $7 \times 10^{-5}$. The quadratic term captures the gentle downward curvature of the cosine near its peak, the same curvature a Hessian encodes for optimization.

### Problem 4.37: Newton on a flat minimum

**Problem.** Apply one Newton step to $f(x) = x^4$ from $x_0 = 1$, and explain why convergence here is slow.

**Solution.** With $f'(x) = 4x^3$ and $f''(x) = 12x^2$, the step is

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} = 1 - \frac{4}{12} = \frac{2}{3} \approx 0.667.$$

Newton only moves a third of the way to the minimum at 0, and each step multiplies $x$ by $\frac{2}{3}$, so it converges *linearly*, not quadratically. The reason is the flat minimum: $f''(0) = 0$, so the function is not locally quadratic and Newton's quadratic model is a poor fit. Quadratic convergence needs a nonzero second derivative at the optimum.

### Problem 4.38: A subgradient of the absolute value

**Problem.** The function $f(x) = |x|$ is not differentiable at 0. Give its subgradient there.

**Solution.** A subgradient $g$ at $x_0$ satisfies $f(x) \geq f(x_0) + g(x - x_0)$ for all $x$. At $x_0 = 0$ this needs $|x| \geq gx$ for all $x$, which holds exactly when $g \in [-1, 1]$. So the subdifferential is the whole interval $\partial f(0) = [-1, 1]$, while for $x > 0$ it is the single value $+1$ and for $x < 0$ it is $-1$. Subgradients let convex but nondifferentiable objectives, like the $\ell_1$ penalty in the lasso or the SVM's hinge, still be optimized by descent: any subgradient gives a valid downhill direction.

### Problem 4.39: The hinge loss is convex

**Problem.** Show that the hinge loss $h(t) = \max(0, 1 - t)$ is convex.

**Solution.** The hinge is the maximum of two affine functions, the constant 0 and the line $1 - t$. A maximum of convex (here affine) functions is always convex, because the region above the graph (the epigraph) is the intersection of the two half-planes above each piece, and an intersection of convex sets is convex. Concretely, $h$ is flat at 0 for $t \geq 1$ and falls along the line $1 - t$ for $t < 1$, with a single kink at $t = 1$; it never curves downward. This convexity is what makes the soft-margin SVM a convex optimization problem with a unique optimum.

CHAPTER 16

# Problem Set 5: Supervised and Unsupervised Learning

This set covers Chapters 10 and 11: the perceptron and the support vector machine, the kernel trick and its Gram matrices, the soft margin, then $k$-means, Gaussian mixtures with EM, and principal component analysis. Every classifier is trained or evaluated on numbers, every kernel checked against its feature map, and every cluster or component recomputed by hand.

## 16.1 The perceptron, SVM, and kernels

### Problem 5.1: One perceptron update

**Problem.** A perceptron starts at $\boldsymbol{w} = (0, 0)$, $b = 0$. The point $\boldsymbol{x} = (2, 1)$ has label $y = +1$. Perform one update.

**Solution.** The current score is $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$, which is not positive, so the positive point is misclassified and triggers an update. The perceptron rule adds the signed point to the weights:

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + y\boldsymbol{x} = (0, 0) + 1 \cdot (2, 1) = (2, 1), \qquad b \leftarrow b + y = 1.$$

After the update the score is $\boldsymbol{w}^\top \boldsymbol{x} + b = (2)(2) + (1)(1) + 1 = 6 > 0$, now correctly positive. Each correction nudges the boundary toward classifying the offending point right, and the perceptron convergence theorem guarantees finitely many such updates when the data is linearly separable.

### Problem 5.2: A two-point support vector machine

**Problem.** Two points lie on the $x$-axis: $\boldsymbol{x}_1 = (1, 0)$ with $y_1 = -1$ and $\boldsymbol{x}_2 = (3, 0)$ with $y_2 = +1$. Find the maximum-margin separator $\boldsymbol{w}, b$ and the margin width.

**Solution.** By symmetry the boundary is the perpendicular bisector $x = 2$, and the weight points along $x$, so $\boldsymbol{w} = (w, 0)$. The support vectors sit exactly on the margins $\boldsymbol{w}^\top \boldsymbol{x}_i + b = y_i$:

$$w(1) + b = -1, \qquad w(3) + b = +1.$$

Subtracting gives $2w = 2$, so $w = 1$ and then $b = -2$. Thus $\boldsymbol{w} = (1, 0)$, $b = -2$, boundary $x = 2$. The margin width is

$$\frac{2}{\|\boldsymbol{w}\|} = \frac{2}{1} = 2,$$

the distance between the two parallel margin lines $x = 1$ and $x = 3$. The machine maximized this gap subject to classifying both points correctly (Fig. 16.1).

### Problem 5.3: Evaluating the decision function

**Problem.** Using the machine of Problem 5.2 ($\boldsymbol{w} = (1, 0)$, $b = -2$), classify the test point $\boldsymbol{x} = (2.5,\ 5)$.

**Solution.** The label is the sign of the score:

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b = (1)(2.5) + (0)(5) - 2 = 0.5 > 0,$$

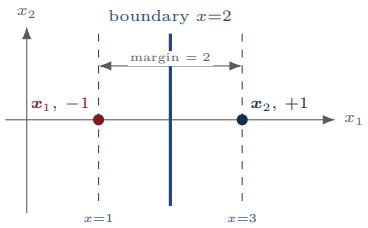



**Figure 16.1.** The two-point SVM. The boundary $x = 2$ is equidistant from the two support vectors, and the margin (the gap between the dashed lines) is as wide as possible.

so the point is class $+1$. The large second coordinate 5 is irrelevant because $\boldsymbol{w} = (1, 0)$ ignores it: this trained machine looks only at $x_1$, since the training data varied only along that axis. A linear classifier is exactly this dot product plus a threshold.

## Problem 5.4: Hinge loss of a point

**Problem.** A point has margin $y\,f(\boldsymbol{x}) = 0.5$. Compute its hinge loss.

**Solution.** The hinge loss is $\ell = \max(0,\ 1 - y\,f(\boldsymbol{x}))$:

$$\ell = \max(0,\ 1 - 0.5) = 0.5.$$

The point is correctly classified ($yf > 0$) but sits inside the margin ($yf < 1$), so it pays a positive loss of 0.5 and remains a support vector. Only points safely beyond the margin ($yf \geq 1$) pay nothing and drop out of the solution, which is the source of the SVM's sparsity.

## Problem 5.5: Margin width from the weights

**Problem.** An SVM has weight vector $\boldsymbol{w} = (1, 2)$. What is its margin width?

**Solution.** The margin between the planes $\boldsymbol{w}^\top \boldsymbol{x} + b = \pm 1$ has width $2/\|\boldsymbol{w}\|$. Here $\|\boldsymbol{w}\| = \sqrt{1^2 + 2^2} = \sqrt{5}$, so

$$\text{margin} = \frac{2}{\sqrt{5}} \approx 0.894.$$

Minimizing $\frac{1}{2}\|\boldsymbol{w}\|^2$, the SVM objective, maximizes this width: a smaller weight vector buys a wider safety corridor around the boundary.

## Problem 5.6: A polynomial kernel and its feature map

**Problem.** For $\boldsymbol{x} = (1, 2)$ and $\boldsymbol{z} = (3, 1)$, compute the kernel $k(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^\top \boldsymbol{z})^2$ directly, and verify it equals $\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{z})$ for the feature map $\boldsymbol{\phi}(\boldsymbol{x}) = (x_1^2,\ \sqrt{2}\,x_1 x_2,\ x_2^2)$.

**Solution.** The inner product is $\boldsymbol{x}^\top \boldsymbol{z} = 1 \cdot 3 + 2 \cdot 1 = 5$, so the kernel is $k = 5^2 = 25$. Now the feature maps: $\boldsymbol{\phi}(\boldsymbol{x}) = (1,\ 2\sqrt{2},\ 4)$ and $\boldsymbol{\phi}(\boldsymbol{z}) = (9,\ 3\sqrt{2},\ 1)$. Their inner product is

$$\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{z}) = 1 \cdot 9 + (2\sqrt{2})(3\sqrt{2}) + 4 \cdot 1 = 9 + 12 + 4 = 25.$$

They agree. The kernel computes a dot product in the three-dimensional quadratic feature space using only the original two-dimensional inner product, the essence of the kernel trick: never form $\boldsymbol{\phi}$, just evaluate $k$.

### Problem 5.7: An RBF kernel value

**Problem.** For the Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\|\boldsymbol{x} - \boldsymbol{z}\|^2/2)$, compute $k$ for $\boldsymbol{x} = (0, 0)$ and $\boldsymbol{z} = (1, 1)$.

**Solution.** The squared distance is $\|\boldsymbol{x} - \boldsymbol{z}\|^2 = 1^2 + 1^2 = 2$, so

$$k = \exp(-2/2) = e^{-1} \approx 0.368.$$

The RBF kernel measures similarity as a bump that decays with distance: identical points give $k = 1$, and far-apart points give $k \to 0$. Its implicit feature space is infinite-dimensional, yet $k$ is a one-line computation, which is why the RBF SVM can draw arbitrarily curved boundaries cheaply.

### Problem 5.8: A kernel matrix is positive semidefinite

**Problem.** A two-point Gram matrix is $K = \left[\begin{smallmatrix} 1 & 0.5 \\ 0.5 & 1 \end{smallmatrix}\right]$. Confirm it is positive semidefinite, as every valid kernel matrix must be.

**Solution.** A symmetric matrix is positive semidefinite when all its eigenvalues are nonnegative. From $(1-\lambda)^2 - 0.25 = 0$, i.e. $1 - \lambda = \pm 0.5$, the eigenvalues are 1.5 and 0.5, both positive, so $K \succ 0$. Mercer's condition requires exactly this of every kernel: for any points, the Gram matrix $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ must be PSD, which guarantees the kernel corresponds to an inner product in some feature space and the SVM's quadratic program stays convex.

## 16.2 Clustering and PCA

### Problem 5.9: One round of $k$-means

**Problem.** Run $k$-means with $k = 2$ on the points $\{1, 2, 9, 10\}$, starting from centroids $\mu_1 = 0$, $\mu_2 = 8$. Carry out one assignment and update step.

**Solution.** *Assignment.* Each point joins its nearer centroid. Distances to $\mu_1 = 0$ are $1, 2, 9, 10$; to $\mu_2 = 8$ they are $7, 6, 1, 2$. So $1, 2$ join cluster 1 and $9, 10$ join cluster 2. *Update.* Recompute each centroid as its cluster's mean:

$$\mu_1 = \frac{1+2}{2} = 1.5, \qquad \mu_2 = \frac{9+10}{2} = 9.5.$$

A second assignment would not change the clusters, so the algorithm has converged in one round to the obvious grouping. $k$-means alternates these two steps, each of which can only decrease the within-cluster sum of squares.

### Problem 5.10: The $k$-means objective

**Problem.** Compute the within-cluster sum of squares for the converged clustering of Problem 5.9.

**Solution.** The objective sums each point's squared distance to its centroid:

$$J = (1 - 1.5)^2 + (2 - 1.5)^2 + (9 - 9.5)^2 + (10 - 9.5)^2 = 0.25 + 0.25 + 0.25 + 0.25 = 1.$$

This is the quantity $k$-means minimizes. Splitting the points the other way (say $\{1, 2, 9\}$ and $\{10\}$) would raise $J$, so the natural grouping is also the optimal one here. Lower $J$ means tighter clusters, and adding clusters always lowers $J$, which is why the elbow in $J$ versus $k$, not $J$ alone, guides the choice of $k$.

### Problem 5.11: A GMM responsibility

**Problem.** A one-dimensional Gaussian mixture has two equally weighted components, $\mathcal{N}(0, 1)$ and $\mathcal{N}(4, 1)$. Compute the responsibility of the first component for the point $x = 1$.

**Solution.** The responsibility is the posterior probability that component 1 generated $x$, by Bayes with equal priors $\frac{1}{2}$:

$$\gamma_1 = \frac{\frac{1}{2}\mathcal{N}(1; 0, 1)}{\frac{1}{2}\mathcal{N}(1; 0, 1) + \frac{1}{2}\mathcal{N}(1; 4, 1)}.$$

The Gaussian densities depend on $x$ only through $\exp(-(x-\mu)^2/2)$: for $\mu = 0$ the exponent is $-\frac{1}{2}$, for $\mu = 4$ it is $-\frac{9}{2}$. So

$$\gamma_1 = \frac{e^{-1/2}}{e^{-1/2} + e^{-9/2}} = \frac{0.6065}{0.6065 + 0.0111} \approx 0.982.$$

The point $x = 1$ is much closer to component 1, so it is assigned there with 98% confidence, a *soft* assignment unlike $k$-means' hard one. This is the E-step of EM.

### Problem 5.12: PCA reconstruction error

**Problem.** A covariance matrix is $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 5 & 4 \\ 4 & 5 \end{smallmatrix}\right]$, with top eigenvector $\boldsymbol{v}_1 = \frac{1}{\sqrt{2}}(1,1)$. Project the point $\boldsymbol{x} = (2,1)$ onto $\boldsymbol{v}_1$, reconstruct, and find the squared reconstruction error.

**Solution.** The first principal coordinate is $y_1 = \boldsymbol{v}_1^\top \boldsymbol{x} = \frac{1}{\sqrt{2}}(2+1) = \frac{3}{\sqrt{2}}$. Reconstructing from it,

$$\hat{\boldsymbol{x}} = y_1 \boldsymbol{v}_1 = \tfrac{3}{\sqrt{2}} \cdot \tfrac{1}{\sqrt{2}}(1,1) = \tfrac{3}{2}(1,1) = (1.5, 1.5).$$

The error is $\boldsymbol{x} - \hat{\boldsymbol{x}} = (0.5, -0.5)$ with squared length $0.25 + 0.25 = 0.5$. This equals the square of the discarded second coordinate $y_2 = \boldsymbol{v}_2^\top \boldsymbol{x} = \frac{1}{\sqrt{2}}(2-1) = \frac{1}{\sqrt{2}}$, since $y_2^2 = \frac{1}{2} = 0.5$: the leftover vector is exactly $y_2 \boldsymbol{v}_2$. Averaged over a dataset, this squared error equals the discarded eigenvalue $\lambda_2$, the working content of Eckart–Young for PCA.

### Problem 5.13: PCA variance explained

**Problem.** For the covariance $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 5 & 4 \\ 4 & 5 \end{smallmatrix}\right]$, find the eigenvalues and the fraction of variance captured by the first principal component.

**Solution.** From $(5-\lambda)^2 - 16 = 0$, i.e. $5 - \lambda = \pm 4$, the eigenvalues are $\lambda_1 = 9$ and $\lambda_2 = 1$. The variance along each principal axis is its eigenvalue, so the first component captures

$$\frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{9}{9+1} = 0.9,$$

that is 90% of the total variance, along the direction $(1,1)$. Dropping the second component therefore loses only 10%, which is exactly the $\lambda_2/(\lambda_1 + \lambda_2)$ the reconstruction error of Problem 5.12 averages to.

### Problem 5.14: Whitening a covariance to the identity

**Problem.** Data has covariance $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 4 & 0 \\ 0 & 9 \end{smallmatrix}\right]$. Find a linear transform $W$ that makes the transformed data have identity covariance.

**Solution.** A linear map $\boldsymbol{x} \mapsto W\boldsymbol{x}$ sends covariance $\boldsymbol{\Sigma}$ to $W\boldsymbol{\Sigma}W^\top$. Choosing $W = \boldsymbol{\Sigma}^{-1/2}$ gives $W\boldsymbol{\Sigma}W^\top = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1/2} = I$. Since $\boldsymbol{\Sigma}$ is diagonal, $\boldsymbol{\Sigma}^{-1/2} = \operatorname{diag}(\frac{1}{2}, \frac{1}{3})$, and

$$W\boldsymbol{\Sigma}W = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 9 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{3} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Whitening rescales each axis by one over its standard deviation, turning an elongated elliptical cloud into a round one with unit variance in every direction, the preprocessing step that makes many algorithms scale-invariant.

## 16.3 More learning fundamentals

### Problem 5.15: Cosine similarity

**Problem.** Compute the cosine similarity between $\boldsymbol{u} = (1,2)$ and $\boldsymbol{v} = (2,1)$.

**Solution.** Cosine similarity is the normalized inner product, $\dfrac{\boldsymbol{u}^\top \boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|}$. Here $\boldsymbol{u}^\top \boldsymbol{v} = 2 + 2 = 4$ and $\|\boldsymbol{u}\| = \|\boldsymbol{v}\| = \sqrt{5}$, so

$$\cos\vartheta = \frac{4}{\sqrt{5} \cdot \sqrt{5}} = \frac{4}{5} = 0.8.$$

The two vectors point in fairly similar directions ($\vartheta \approx 37^\circ$). Cosine similarity is the linear kernel after normalization, and it is the workhorse similarity in text and embedding models because it ignores magnitude and compares direction alone.

### Problem 5.16: A soft-margin slack variable

**Problem.** In a soft-margin SVM a training point has margin $y\,f(\boldsymbol{x}) = 0.3$. What is its slack $\xi$, and is the point inside the margin or misclassified?

**Solution.** The slack measures the margin violation, $\xi = \max(0,\ 1 - y\,f(\boldsymbol{x}))$:

$$\xi = \max(0,\ 1 - 0.3) = 0.7.$$

Since $0 < yf < 1$, the point is correctly classified ($yf > 0$) but sits inside the margin, paying a slack of 0.7. A slack between 0 and 1 means inside the margin; a slack above 1 would mean $yf < 0$, an outright misclassification. The soft-margin objective trades total slack $\sum \xi_i$ against margin width through the penalty $C$.

### Problem 5.17: Counting GMM parameters

**Problem.** How many free parameters does a Gaussian mixture with $K = 3$ full-covariance components in $d = 2$ dimensions have?

**Solution.** Count each ingredient. Each component has a mean in $\mathbb{R}^2$ (2 numbers) and a symmetric $2 \times 2$ covariance (3 distinct entries), so 5 per component, times 3 components is 15. The mixing weights are 3 numbers that sum to 1, contributing $3 - 1 = 2$ free parameters. The total is

$$3(2 + 3) + (3 - 1) = 15 + 2 = 17.$$

Parameter counts like this set the model's capacity: more components fit finer structure but risk overfitting, and full covariances (3 each) cost more than diagonal ones (2 each), which is why diagonal or shared covariances are common shortcuts.

### Problem 5.18: Why XOR is not linearly separable

**Problem.** The XOR data labels $(0, 0)$ and $(1, 1)$ as $+1$, and $(0, 1)$ and $(1, 0)$ as $-1$. Show no line separates the classes, and name a fix.

**Solution.** Suppose a linear classifier $\operatorname{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$ separated them. The two positive points average to $(0.5, 0.5)$ and so do the two negatives, so any linear score $\boldsymbol{w}^\top \boldsymbol{x} + b$ has the same mean on both classes; a single hyperplane cannot put one class entirely on each side when their centroids coincide. Concretely, requiring $\boldsymbol{w}^\top(0, 0) + b > 0$ and $\boldsymbol{w}^\top(1, 1) + b > 0$ but $\boldsymbol{w}^\top(1, 0) + b < 0$ and $\boldsymbol{w}^\top(0, 1) + b < 0$ forces $b > 0$, $w_1 + w_2 + b > 0$, $w_1 + b < 0$, $w_2 + b < 0$; adding the last two gives $w_1 + w_2 + 2b < 0$, which with $b > 0$ contradicts $w_1 + w_2 + b > 0$. The standard fix is a feature map or kernel: add the product feature $x_1 x_2$ (or use a quadratic or RBF kernel) and the classes become separable in the lifted space. XOR is the textbook reason linear models alone are not enough.

### Problem 5.19: Hard versus soft assignment

**Problem.** Using the responsibilities of Problem 5.11 ($\gamma_1 \approx 0.982$), contrast what $k$-means and a GMM would do with the point $x = 1$.

**Solution.** $k$-means makes a *hard* assignment: $x = 1$ is closer to the mean 0 than to 4, so it goes entirely to cluster 1 with weight 1, and cluster 2 gets nothing. The GMM makes a *soft* assignment using the responsibilities: $x = 1$ belongs 0.982 to component 1 and 0.018 to component 2, and both means are updated with these fractional weights in the M-step. In the limit of shrinking, equal variances the responsibilities sharpen to 0 or 1 and the GMM's EM reduces exactly to $k$-means; the GMM is the probabilistic generalization that also captures cluster shape and overlap.

### Problem 5.20: PCA through the SVD of the data

**Problem.** Explain why the principal components of centered data $\tilde{X}$ (rows are samples) are the right singular vectors of $\tilde{X}$, and relate the eigenvalues of the covariance to the singular values.

**Solution.** The sample covariance is $\boldsymbol{\Sigma} = \frac{1}{m}\tilde{X}^\top \tilde{X}$. Write the SVD $\tilde{X} = U\Sigma V^\top$. Then

$$\tilde{X}^\top \tilde{X} = V\Sigma U^\top U\Sigma V^\top = V\Sigma^2 V^\top,$$

using $U^\top U = I$, so $\boldsymbol{\Sigma} = \frac{1}{m}V\Sigma^2 V^\top$ is already an eigendecomposition: the principal directions are the columns of $V$ (the right singular vectors of $\tilde{X}$), and the variance along component $j$ is $\lambda_j = \sigma_j^2/m$. Computing the SVD of $\tilde{X}$ directly is more numerically stable than forming $\tilde{X}^\top \tilde{X}$, which squares the condition number, so production PCA uses the SVD. This is the bridge between Chapter 4 and Chapter 11: PCA is the SVD of the centered data.

## 16.4 Further machine-learning practice

### Problem 5.21: An RBF Gram matrix

**Problem.** For the three points $(0,0)$, $(1,0)$, $(0,1)$ and the kernel $k(\boldsymbol{x},\boldsymbol{z}) = \exp(-\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{z}\|^2)$, compute the $3\times 3$ Gram matrix.

**Solution.** Each entry is $\exp(-\frac{1}{2}d^2)$ for the squared distance $d^2$ between two points. The diagonal is all 1 (distance 0). The squared distances are $\|(0,0)-(1,0)\|^2 = 1$, $\|(0,0)-(0,1)\|^2 = 1$, and $\|(1,0)-(0,1)\|^2 = 2$, so

$$K = \begin{bmatrix} 1 & e^{-1/2} & e^{-1/2} \\ e^{-1/2} & 1 & e^{-1} \\ e^{-1/2} & e^{-1} & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & 0.607 & 0.607 \\ 0.607 & 1 & 0.368 \\ 0.607 & 0.368 & 1 \end{bmatrix}.$$

The matrix is symmetric with 1s on the diagonal, and nearer points have larger entries. Such a Gram matrix is positive semidefinite for the RBF kernel, which is what lets the SVM solve its convex dual using only these similarities, never the (infinite-dimensional) feature vectors.

### Problem 5.22: A Gaussian log-likelihood

**Problem.** Two data points $\{0, 4\}$ are modeled as iid $\mathcal{N}(2, 1)$. Compute the log-likelihood.

**Solution.** The one-dimensional Gaussian density is $\frac{1}{\sqrt{2\pi}}e^{-(x-\mu)^2/2}$, so its log is $-\frac{1}{2}\ln(2\pi) - \frac{1}{2}(x-\mu)^2$. Summing over the two points with $\mu = 2$:

$$\ell = \sum_{x\in\{0,4\}} \left[-\tfrac{1}{2}\ln(2\pi) - \tfrac{1}{2}(x-2)^2\right] = -\ln(2\pi) - \tfrac{1}{2}(4+4) = -1.838 - 4 \approx -5.84.$$

Both points sit two units from the mean, so each contributes the same penalty. The mean $\mu = 2$ is in fact the MLE here (the sample mean of 0 and 4), which maximizes this log-likelihood; any other $\mu$ would make it more negative.

### Problem 5.23: Precision, recall, and the $F_1$ score

**Problem.** A classifier produces 8 true positives, 2 false positives, and 4 false negatives. Compute precision, recall, and the $F_1$ score.

**Solution.** Precision is the fraction of predicted positives that are correct, and recall the fraction of actual positives that are found:

$$\text{precision} = \frac{TP}{TP+FP} = \frac{8}{10} = 0.8, \qquad \text{recall} = \frac{TP}{TP+FN} = \frac{8}{12} \approx 0.667.$$

The $F_1$ score is their harmonic mean,

$$F_1 = \frac{2\cdot \text{precision}\cdot \text{recall}}{\text{precision}+\text{recall}} = \frac{2(0.8)(0.667)}{0.8+0.667} \approx 0.727.$$

The harmonic mean punishes imbalance: a classifier that scored high on one but low on the other would get a low $F_1$. Here the moderate recall (four positives missed) pulls $F_1$ below the precision.

### Problem 5.24: The linear kernel

**Problem.** Evaluate the linear kernel $k(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x}^\top \boldsymbol{z}$ for $\boldsymbol{x} = (1, 2)$, $\boldsymbol{z} = (3, 4)$, and explain what SVM it produces.

**Solution.** The linear kernel is just the dot product: $k = (1)(3) + (2)(4) = 3 + 8 = 11$. Its feature map is the identity, $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x}$, so the SVM with a linear kernel is the ordinary linear SVM in the original space, drawing a straight hyperplane. It is the baseline against which nonlinear kernels (polynomial, RBF) are judged: use the linear kernel when the data is already close to linearly separable, and reach for a richer kernel only when it is not.

### Problem 5.25: An EM mean update

**Problem.** In the M-step of EM, three points $\{1, 2, 9\}$ carry responsibilities $\{0.9, 0.8, 0.1\}$ for component 1. Compute the updated mean $\mu_1$.

**Solution.** The M-step sets each component's mean to a responsibility-weighted average of the points:

$$\mu_1 = \frac{\sum_i \gamma_i x_i}{\sum_i \gamma_i} = \frac{0.9(1) + 0.8(2) + 0.1(9)}{0.9 + 0.8 + 0.1} = \frac{0.9 + 1.6 + 0.9}{1.8} = \frac{3.4}{1.8} \approx 1.89.$$

The point 9 is mostly claimed by the other component (responsibility only 0.1), so it barely shifts $\mu_1$, which sits near the well-claimed points 1 and 2. This soft, weighted update is what distinguishes EM from $k$-means, where each point would count fully for one cluster and not at all for the other.

### Problem 5.26: Why distances concentrate in high dimensions

**Problem.** Explain, with a short calculation, why nearest-neighbor and clustering methods struggle as the dimension $d$ grows.

**Solution.** Take points with independent coordinates, each contributing a roughly constant expected squared gap $c$ to a pairwise squared distance. Then a squared distance is a sum of $d$ such terms, with mean growing like $cd$ and standard deviation growing like $\sqrt{d}$ (the sum of $d$ independent contributions). The ratio

$$\frac{\text{spread of distances}}{\text{typical distance}} \sim \frac{\sqrt{d}}{cd} = \frac{1}{c\sqrt{d}} \to 0$$

shrinks as $d$ grows, so all pairwise distances become nearly equal. When the nearest and farthest neighbors sit at almost the same distance, "nearest neighbor" loses meaning and clustering blurs. This concentration is the curse of dimensionality, and it is exactly why PCA and other dimension-reduction methods (Chapter 11) are run before distance-based learning.

### Problem 5.27: Two perceptron updates

**Problem.** Start a perceptron at $\boldsymbol{w} = (0, 0)$, $b = 0$. Present $\boldsymbol{x}_1 = (1, 1)$ with $y_1 = +1$, then $\boldsymbol{x}_2 = (-1, 0)$ with $y_2 = -1$. Perform the updates.

**Solution.** *First point.* The score is $\boldsymbol{w}^\top \boldsymbol{x}_1 + b = 0$, not positive, so the $+1$ point is misclassified. Update $\boldsymbol{w} \leftarrow \boldsymbol{w} + y_1 \boldsymbol{x}_1 = (1, 1)$ and $b \leftarrow b + 1 = 1$. *Second point.* Now the score is $\boldsymbol{w}^\top \boldsymbol{x}_2 + b = (1)(-1) + (1)(0) + 1 = 0$, not negative, so the $-1$ point is misclassified too. Update $\boldsymbol{w} \leftarrow \boldsymbol{w} + y_2 \boldsymbol{x}_2 = (1, 1) + (-1)(-1, 0) = (2, 1)$ and $b \leftarrow b - 1 = 0$. After two updates the machine is $\boldsymbol{w} = (2, 1)$, $b = 0$. Each correction tilts the boundary toward the offending point; on separable data the perceptron makes only finitely many such corrections before classifying everything correctly.

### Problem 5.28: PCA on diagonal covariance

**Problem.** Data has covariance $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 4 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. Give the principal components and the variance each captures.

**Solution.** A diagonal covariance is already diagonalized, so the principal components are the coordinate axes: $\boldsymbol{e}_1 = (1, 0)$ with variance 4, and $\boldsymbol{e}_2 = (0, 1)$ with variance 1. The first captures $\frac{4}{4+1} = 0.8$, that is 80% of the variance. With no off-diagonal terms the data axes do not need rotating; PCA only does interesting work when correlations tilt the cloud off the axes, as in Problem 5.12. Here the answer is read straight off the diagonal.

## Problem 5.29: Accuracy from a confusion matrix

**Problem.** A classifier has 40 true positives, 50 true negatives, 5 false positives, and 5 false negatives. Compute its accuracy, and say when accuracy misleads.

**Solution.** Accuracy is the fraction of correct predictions:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{40 + 50}{100} = 0.90.$$

A 90% accuracy here is meaningful because the classes are balanced (45 actual positives, 55 negatives). Accuracy misleads on imbalanced data: if only 1% of cases were positive, a model that always predicts "negative" would score 99% accuracy while catching nothing. That is exactly when precision, recall, and $F_1$ (Problem 5.23) become the right metrics.

## Problem 5.30: Ridge regression shrinks the solution

**Problem.** Ridge regression solves $(A^\top A + \lambda I)\boldsymbol{\beta} = A^\top \boldsymbol{y}$. With $A^\top A = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right]$, $A^\top \boldsymbol{y} = (4, 4)$, compare the ordinary least-squares and ridge ($\lambda = 2$) solutions.

**Solution.** Ordinary least squares solves $A^\top A\boldsymbol{\beta} = A^\top \boldsymbol{y}$, i.e. $2\boldsymbol{\beta} = (4, 4)$, giving $\boldsymbol{\beta}_{\text{OLS}} = (2, 2)$. Ridge adds $\lambda I = 2I$ to the matrix:

$$(A^\top A + 2I)\boldsymbol{\beta} = (4I)\boldsymbol{\beta} = (4, 4) \implies \boldsymbol{\beta}_{\text{ridge}} = (1, 1).$$

The penalty halved the coefficients, from $(2, 2)$ to $(1, 1)$. Ridge lifts every eigenvalue of $A^\top A$ by $\lambda$ (here $2 \to 4$), which both shrinks the estimate toward zero and cures ill-conditioning by pushing small eigenvalues away from zero. The bias this introduces is traded for a large drop in variance, the bias–variance bargain.

## Problem 5.31: Orthogonal vectors have zero cosine similarity

**Problem.** Compute the cosine similarity of $\boldsymbol{u} = (1, 0)$ and $\boldsymbol{v} = (0, 1)$, and relate it to the linear kernel.

**Solution.** The inner product is $\boldsymbol{u}^\top \boldsymbol{v} = 1(0) + 0(1) = 0$, so the cosine similarity is $0/(\|\boldsymbol{u}\|\|\boldsymbol{v}\|) = 0$. The vectors are orthogonal, pointing in unrelated directions ($90^\circ$ apart). Since the linear kernel *is* the inner product, orthogonal inputs have zero kernel value: a linear SVM sees them as carrying no shared information. In a bag-of-words model this is the case of two documents with no words in common, the baseline of total dissimilarity.

## Problem 5.32: $k$-means++ seeding

**Problem.** Why does $k$-means++ choose initial centroids spread apart rather than uniformly at random, and what failure does it prevent?

**Solution.** Plain $k$-means with random initial centroids can place two seeds inside the same true cluster, leaving another cluster with no seed; the algorithm then converges to a poor local minimum that splits one group and merges two others. $k$-means++ instead picks each new centroid with probability proportional to the squared distance from the nearest existing centroid, so far-apart points are favored and the seeds land in different clusters. Concretely, after seeding one centroid, a point at squared distance $d^2$ is chosen with probability $d^2 / \sum d^2$, biasing strongly toward the far side of the data. This costs one extra pass but reliably avoids the degenerate initializations, and it comes with a provable approximation guarantee on the final objective.

## Problem 5.33: Why features must be scaled

**Problem.** Two features are measured in different units: $x_1$ in metres (values around 1) and $x_2$ in millimetres (values around 1000). Explain why $k$-means or an RBF kernel can go wrong, and the fix.

**Solution.** Euclidean distance squares each coordinate gap and sums: $\|\boldsymbol{x} - \boldsymbol{z}\|^2 = (\Delta x_1)^2 + (\Delta x_2)^2$. With $x_2$ a thousand times larger, its gaps dominate the sum and $x_1$ becomes invisible, so clustering and the RBF kernel effectively ignore the metre feature entirely. The fix is to standardize each feature to zero mean and unit variance (a $z$-score) before computing distances, so the two contribute comparably. This is why feature scaling is a standard

preprocessing step for every distance- or gradient-based method; only scale-invariant models like decision trees escape it.

### Problem 5.34: The precision–recall tradeoff

**Problem.** A classifier outputs a score and labels positive when the score exceeds a threshold $\tau$. Describe how precision and recall move as $\tau$ rises, and when each extreme is appropriate.

**Solution.** Raising $\tau$ makes the classifier more conservative: it flags fewer positives, so false positives fall and precision rises, while true positives are also missed, so recall falls. Lowering $\tau$ does the reverse, catching more positives (high recall) at the cost of more false alarms (low precision). The two trade off along the precision–recall curve. A high threshold suits costly actions where a false positive is expensive (flagging fraud for manual review); a low threshold suits screening where missing a positive is dangerous (a medical test that must not miss disease). The $F_1$ score (Problem 5.23) picks a balance, and the full curve summarizes every threshold at once.

### Problem 5.35: A logistic-regression decision boundary

**Problem.** A logistic model has weights $\boldsymbol{w} = (1, 1)$ and bias $b = -3$. Give the decision boundary and classify $(1, 1)$ and $(2, 2)$.

**Solution.** Logistic regression predicts class 1 when $\sigma(\boldsymbol{w}^\top \boldsymbol{x} + b) > \frac{1}{2}$, i.e. when the logit $\boldsymbol{w}^\top \boldsymbol{x} + b > 0$. The boundary is the line $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$, here $x_1 + x_2 - 3 = 0$, the line $x_1 + x_2 = 3$. At $(1, 1)$ the logit is $1 + 1 - 3 = -1 < 0$, so class 0 (probability $\sigma(-1) \approx 0.27$); at $(2, 2)$ it is $4 - 3 = 1 > 0$, so class 1 (probability $\sigma(1) \approx 0.73$). Like the SVM the boundary is a hyperplane $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$, but logistic regression also reports a calibrated probability through the sigmoid.

### Problem 5.36: The sigmoid

**Problem.** Evaluate the logistic sigmoid $\sigma(z) = 1/(1 + e^{-z})$ at $z = 0, 2, -2$, and note its symmetry.

**Solution.** Compute each:

$$\sigma(0) = \tfrac{1}{1+1} = 0.5, \quad \sigma(2) = \tfrac{1}{1+e^{-2}} \approx 0.881, \quad \sigma(-2) = \tfrac{1}{1+e^{2}} \approx 0.119.$$

Note $\sigma(2) + \sigma(-2) = 1$, the symmetry $\sigma(-z) = 1 - \sigma(z)$. The sigmoid squashes any real logit into $(0, 1)$, mapping 0 to the neutral 0.5 and saturating toward 1 for large positive and 0 for large negative inputs. Its derivative $\sigma(z)(1 - \sigma(z))$ is largest at $z = 0$ and vanishes in the tails, the saturation that can stall gradient-based training.

### Problem 5.37: Responsibility with unequal mixing weights

**Problem.** A mixture has components $\mathcal{N}(0, 1)$ and $\mathcal{N}(4, 1)$ with mixing weights $\pi = (0.3, 0.7)$. Find the responsibility of component 1 for $x = 1$.

**Solution.** The responsibility weights each component's density by its prior:

$$\gamma_1 = \frac{0.3\,\mathcal{N}(1; 0, 1)}{0.3\,\mathcal{N}(1; 0, 1) + 0.7\,\mathcal{N}(1; 4, 1)}.$$

The densities go as $e^{-(x-\mu)^2/2}$: for $\mu = 0$ the exponent is $-\frac{1}{2}$, for $\mu = 4$ it is $-\frac{9}{2}$. So

$$\gamma_1 = \frac{0.3\,e^{-1/2}}{0.3\,e^{-1/2} + 0.7\,e^{-9/2}} = \frac{0.182}{0.182 + 0.0078} \approx 0.959.$$

Even though component 1 has the smaller prior weight (0.3), $x = 1$ is so much closer to its mean that it claims 96% responsibility. The prior tilts the assignment but the likelihood dominates when the point is decisively nearer one component.

## Problem 5.38: Cross-entropy loss

**Problem.** For a true label $y = 1$, compute the cross-entropy loss $-\ln p$ when the model predicts $p = 0.9$ and when it predicts $p = 0.6$.

**Solution.** The cross-entropy for a positive example is $-\ln p$:

$$-\ln 0.9 \approx 0.105, \qquad -\ln 0.6 \approx 0.511.$$

A confident-and-correct prediction ($p = 0.9$) is cheap; a hesitant one ($p = 0.6$) costs nearly five times as much, and the loss diverges to $\infty$ as $p \to 0$ (confidently wrong). This steep penalty for confident mistakes is why cross-entropy, not accuracy, trains classifiers: it gives a smooth, informative gradient everywhere, pushing the predicted probability toward the truth.

## Problem 5.39: Entropy of a fair coin

**Problem.** Compute the entropy of a Bernoulli(0.5) in bits, and say why it is the maximum for a binary variable.

**Solution.** Entropy is $H = -\sum_k p_k \log_2 p_k$:

$$H = -\big(0.5 \log_2 0.5 + 0.5 \log_2 0.5\big) = -\big(0.5(-1) + 0.5(-1)\big) = 1 \text{ bit}.$$

A fair coin carries exactly one bit, the most any binary variable can. The binary entropy $H(p) = -p \log_2 p - (1 - p) \log_2(1 - p)$ peaks at $p = \frac{1}{2}$ and falls to 0 at $p \in \{0, 1\}$, mirroring the Bernoulli variance $p(1 - p)$: maximum uncertainty sits at the balanced coin, and a decision tree's information gain is largest when a split most reduces this entropy.

## Problem 5.40: Euclidean versus Manhattan distance

**Problem.** For the displacement $\boldsymbol{u} = (1, 2, 2)$, compute the Euclidean ($\ell_2$) and Manhattan ($\ell_1$) distances from the origin, and note when each is used.

**Solution.** The Euclidean distance is the straight-line length, the Manhattan the sum of coordinate moves:

$$\|\boldsymbol{u}\|_2 = \sqrt{1^2 + 2^2 + 2^2} = \sqrt{9} = 3, \qquad \|\boldsymbol{u}\|_1 = |1| + |2| + |2| = 5.$$

Euclidean (3) is shorter than Manhattan (5), as the diagonal beats walking the grid. Euclidean distance underlies $k$-means and the RBF kernel; Manhattan (taxicab) is more robust to outliers and is the geometry of the $\ell_1$ penalty that drives sparsity in the lasso. The choice of metric is a modeling decision in its own right.

PART VIII

# Practice Examinations, Worked in Full

CHAPTER 17

# Practice Midterm, Worked in Full

This practice midterm covers the matrix-methods and probability core (Chapters 1 to 6). Each question is stated with its point value as on a real paper, then solved completely, every part spelled out and every answer checked by re-multiplying or re-substituting. Work each once closed-book before reading the solution.

## 17.1 Vectors, matrices, and linear systems

### Question 1: A matrix product and the transpose rule (10 points)

**Problem.** For $A = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 3 \end{smallmatrix}\right]$ and $B = \left[\begin{smallmatrix} 1 & 2 \\ 0 & 1 \end{smallmatrix}\right]$, compute $AB$ and verify $(AB)^{\top} = B^{\top}A^{\top}$.

**Solution.** Each entry is a row of $A$ dotted with a column of $B$:

$$AB = \begin{bmatrix} 2\cdot 1 + 1\cdot 0 & 2\cdot 2 + 1\cdot 1 \\ 1\cdot 1 + 3\cdot 0 & 1\cdot 2 + 3\cdot 1 \end{bmatrix} = \begin{bmatrix} 2 & 5 \\ 1 & 5 \end{bmatrix}, \qquad (AB)^{\top} = \begin{bmatrix} 2 & 1 \\ 5 & 5 \end{bmatrix}.$$

Now $B^{\top}A^{\top} = \left[\begin{smallmatrix} 1 & 0 \\ 2 & 1 \end{smallmatrix}\right]\left[\begin{smallmatrix} 2 & 1 \\ 1 & 3 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 2 & 1 \\ 5 & 5 \end{smallmatrix}\right]$, which matches. The factors reverse order under the transpose, the same reversal as $(AB)^{-1} = B^{-1}A^{-1}$.

### Question 2: Dot product and angle (10 points)

**Problem.** For $\boldsymbol{u} = (1, 2, 2)$ and $\boldsymbol{v} = (2, 0, 1)$, compute the angle between them.

**Solution.** The inner product is $\boldsymbol{u}^{\top}\boldsymbol{v} = 1(2) + 2(0) + 2(1) = 4$, and the lengths are $\|\boldsymbol{u}\| = \sqrt{1+4+4} = 3$, $\|\boldsymbol{v}\| = \sqrt{4+0+1} = \sqrt{5}$. So

$$\cos\vartheta = \frac{\boldsymbol{u}^{\top}\boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|} = \frac{4}{3\sqrt{5}} \approx 0.596, \qquad \vartheta = \arccos(0.596) \approx 53.4^{\circ}.$$

The vectors point in moderately similar directions. Cauchy–Schwarz is satisfied: $|4| \le 3\sqrt{5} \approx 6.7$.

### Question 3: Coordinates in a basis (10 points)

**Problem.** Express $\boldsymbol{v} = (3, 1)$ in the basis $B = \{(1, 1), (1, -1)\}$.

**Solution.** Solve $c_1(1,1) + c_2(1,-1) = (3,1)$, i.e. $c_1 + c_2 = 3$ and $c_1 - c_2 = 1$. Adding gives $2c_1 = 4$, so $c_1 = 2$ and $c_2 = 1$; the coordinate vector is $[\boldsymbol{v}]_B = (2, 1)$, and indeed $2(1,1) + 1(1,-1) = (3,1)$. Because the basis is orthogonal, the same answer comes from projection: $c_i = \dfrac{\boldsymbol{v}^{\top}\boldsymbol{b}_i}{\boldsymbol{b}_i^{\top}\boldsymbol{b}_i}$ gives $c_1 = \frac{4}{2} = 2$ and $c_2 = \frac{2}{2} = 1$ with no system to solve.

## Question 4: Gram–Schmidt orthonormalization (15 points)

**Problem.** Apply Gram–Schmidt to $\boldsymbol{a}_1 = (1,1,0)$ and $\boldsymbol{a}_2 = (1,1,1)$ to produce an orthonormal pair.

**Solution.** Normalize the first vector: $\|\boldsymbol{a}_1\| = \sqrt{2}$, so $\boldsymbol{q}_1 = \frac{1}{\sqrt{2}}(1,1,0)$. Remove the $\boldsymbol{q}_1$-component of $\boldsymbol{a}_2$. The overlap is $\boldsymbol{a}_2^{\top}\boldsymbol{q}_1 = \frac{1}{\sqrt{2}}(1+1+0) = \sqrt{2}$, so

$$\boldsymbol{v}_2 = \boldsymbol{a}_2 - (\boldsymbol{a}_2^{\top}\boldsymbol{q}_1)\boldsymbol{q}_1 = (1,1,1) - \sqrt{2}\cdot\tfrac{1}{\sqrt{2}}(1,1,0) = (1,1,1) - (1,1,0) = (0,0,1).$$

This already has unit length, so $\boldsymbol{q}_2 = (0,0,1)$. Check: $\boldsymbol{q}_1^{\top}\boldsymbol{q}_2 = 0$ and both are unit vectors, so $\{\boldsymbol{q}_1, \boldsymbol{q}_2\}$ is orthonormal. The procedure peeled the shared $(1,1,0)$ direction out of $\boldsymbol{a}_2$, leaving the pure $z$-direction. Stacking these as columns of $Q$ with the overlaps in $R$ gives the $QR$ factorization.

## Question 5: A $2 \times 2$ system (10 points)

**Problem.** Solve $2x + y = 5$, $x + y = 3$.

**Solution.** Subtract the second equation from the first to eliminate $y$: $(2x + y) - (x + y) = 5 - 3$ gives $x = 2$. Back-substitute into $x + y = 3$ to get $y = 1$. The solution is $(2, 1)$, which checks: $2(2) + 1 = 5$ and $2 + 1 = 3$. Equivalently, with coefficient matrix $\left[\begin{smallmatrix} 2 & 1 \\ 1 & 1 \end{smallmatrix}\right]$ (determinant 1), the inverse is $\left[\begin{smallmatrix} 1 & -1 \\ -1 & 2 \end{smallmatrix}\right]$, and $\left[\begin{smallmatrix} 1 & -1 \\ -1 & 2 \end{smallmatrix}\right]\left[\begin{smallmatrix} 5 \\ 3 \end{smallmatrix}\right] = (2, 1)$.

## Question 6: A $3 \times 3$ system by elimination (15 points)

**Problem.** Solve $x + y + z = 6$, $2x + 3y + z = 11$, $x + 2y + 3z = 14$ by Gaussian elimination.

**Solution.** Eliminate $x$ from the lower rows. Subtract 2×row 1 from row 2 and 1×row 1 from row 3:

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 2 & 3 & 1 & 11 \\ 1 & 2 & 3 & 14 \end{array}\right] \xrightarrow[R_3 - R_1]{R_2 - 2R_1} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & -1 & -1 \\ 0 & 1 & 2 & 8 \end{array}\right].$$

Now subtract row 2 from row 3:

$$\xrightarrow{R_3 - R_2} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 3 & 9 \end{array}\right].$$

Back-substitute from the bottom: $3z = 9 \Rightarrow z = 3$; then $y - z = -1 \Rightarrow y = 2$; then $x + y + z = 6 \Rightarrow x = 1$. The solution is $(1, 2, 3)$, which checks in all three original equations (1+2+3 = 6, 2+6+3 = 11, 1+4+9 = 14). Every pivot was nonzero, so the coefficient matrix is invertible and the solution unique.

## Question 7: A $2 \times 2$ inverse (10 points)

**Problem.** Invert $A = \left[\begin{smallmatrix} 3 & 1 \\ 2 & 4 \end{smallmatrix}\right]$ and verify $AA^{-1} = I$.

**Solution.** The $2 \times 2$ rule is $\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]^{-1} = \frac{1}{ad-bc}\left[\begin{smallmatrix} d & -b \\ -c & a \end{smallmatrix}\right]$. Here $\det A = 3 \cdot 4 - 1 \cdot 2 = 10$, so

$$A^{-1} = \frac{1}{10}\begin{bmatrix} 4 & -1 \\ -2 & 3 \end{bmatrix} = \begin{bmatrix} 0.4 & -0.1 \\ -0.2 & 0.3 \end{bmatrix}.$$

Check: $AA^{-1} = \frac{1}{10}\left[\begin{smallmatrix} 3 & 1 \\ 2 & 4 \end{smallmatrix}\right]\left[\begin{smallmatrix} 4 & -1 \\ -2 & 3 \end{smallmatrix}\right] = \frac{1}{10}\left[\begin{smallmatrix} 12-2 & -3+3 \\ 8-8 & -2+12 \end{smallmatrix}\right] = \frac{1}{10}\left[\begin{smallmatrix} 10 & 0 \\ 0 & 10 \end{smallmatrix}\right] = I$. The nonzero determinant guaranteed the inverse exists; a determinant of zero would have left the formula dividing by zero, the algebraic face of a singular matrix.

### Question 8: A determinant by cofactors (10 points)

**Problem.** Compute $\det A$ for the tridiagonal $A = \left[\begin{smallmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{smallmatrix}\right]$ and state whether $A$ is invertible.

**Solution.** Expand along the first row (signs $+, -, +$):

$$\det A = 3 \det \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} - 1 \det \begin{bmatrix} 1 & 1 \\ 0 & 3 \end{bmatrix} + 0 = 3(9-1) - 1(3-0) = 24 - 3 = 21.$$

Since $\det A = 21 \neq 0$, $A$ is invertible. (The third term vanishes because its cofactor carries the entry $A_{13} = 0$.) A positive determinant for this symmetric tridiagonal matrix is consistent with its being positive definite.

### Question 9: Rank and nullity (10 points)

**Problem.** Find the rank and nullity of $A = \left[\begin{smallmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \end{smallmatrix}\right]$.

**Solution.** Row-reduce: $R_2 - 2R_1$ makes the second row zero, since $(2, 4, 6, 8) = 2(1, 2, 3, 4)$. One nonzero row survives, so $\operatorname{rank} A = 1$. The rank–nullity theorem gives $\operatorname{rank} A + \dim \operatorname{Nul} A = n$ (the number of columns $= 4$), so the nullity is $4 - 1 = 3$. The map sends a 4-dimensional input to a 1-dimensional output, crushing a 3-dimensional subspace to zero; the two rows pointed the same way, so the matrix "sees" only one direction.

## 17.2 Least squares and projections

### Question 10: Least squares and goodness of fit (15 points)

**Problem.** Fit a line $y = a + bx$ to $(1, 2), (2, 5), (3, 7), (4, 8)$. (a) Write the normal equations. (b) Solve for $a, b$. (c) Give the residuals. (d) Compute $R^2$.

**Solution.** *(a)* With $A = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{smallmatrix}\right]$ and $\boldsymbol{y} = (2, 5, 7, 8)$, the normal equations $A^\top A \left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right) = A^\top \boldsymbol{y}$ use

$$A^\top A = \begin{bmatrix} 4 & 10 \\ 10 & 30 \end{bmatrix}, \qquad A^\top \boldsymbol{y} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \end{bmatrix} = \begin{bmatrix} 22 \\ 65 \end{bmatrix},$$

since $\sum y_i = 2 + 5 + 7 + 8 = 22$ and $\sum x_i y_i = 1(2) + 2(5) + 3(7) + 4(8) = 2 + 10 + 21 + 32 = 65$.

*(b)* The determinant of $A^\top A$ is $4(30) - 10(10) = 20$, so by Cramer's rule

$$a = \frac{22(30) - 10(65)}{20} = \frac{660 - 650}{20} = 0.5, \qquad b = \frac{4(65) - 10(22)}{20} = \frac{260 - 220}{20} = 2.$$

The fit is $\hat{y} = 0.5 + 2x$.

*(c)* Predictions are $(2.5, 4.5, 6.5, 8.5)$, so residuals are $\boldsymbol{r} = (2 - 2.5,\ 5 - 4.5,\ 7 - 6.5,\ 8 - 8.5) = (-0.5,\ 0.5,\ 0.5,\ -0.5)$.

*(d)* The residual sum of squares is $\mathrm{SS}_{\mathrm{res}} = 4(0.25) = 1$. The mean is $\bar{y} = \frac{22}{4} = 5.5$, so $\mathrm{SS}_{\mathrm{tot}} = (2 - 5.5)^2 + (5 - 5.5)^2 + (7 - 5.5)^2 + (8 - 5.5)^2 = 12.25 + 0.25 + 2.25 + 6.25 = 21$. Therefore

$$R^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}} = 1 - \frac{1}{21} \approx 0.952,$$

so the line explains about 95% of the variation. The residuals sum to zero, as the intercept always forces.

### Question 11: Projection onto a subspace (10 points)

**Problem.** Project $\boldsymbol{b} = (2, 0, 2)$ onto the column space of $A = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{smallmatrix}\right]$, and verify the residual is orthogonal to $\mathrm{Col}(A)$.

**Solution.** Solve the normal equations $A^\top A \hat{\boldsymbol{x}} = A^\top \boldsymbol{b}$. Here $A^\top A = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 2 \end{smallmatrix}\right]$ and $A^\top \boldsymbol{b} = (2, 2)$, so $\hat{\boldsymbol{x}} = (2, 1)$. The projection is $\hat{\boldsymbol{b}} = A\hat{\boldsymbol{x}} = 2(1, 0, 0) + 1(0, 1, 1) = (2, 1, 1)$, and the residual is $\boldsymbol{r} = \boldsymbol{b} - \hat{\boldsymbol{b}} = (0, -1, 1)$. Check orthogonality: column $1 = (1, 0, 0)$ gives $\boldsymbol{r}^\top (1, 0, 0) = 0$, and column $2 = (0, 1, 1)$ gives $0 - 1 + 1 = 0$. The residual is perpendicular to both columns, the geometric content of least squares.

### Question 12: Projection onto a line (10 points)

**Problem.** Project $\boldsymbol{b} = (3, 4)$ onto the line spanned by $\boldsymbol{a} = (1, 0)$, and give the residual.

**Solution.** The projection keeps the component of $\boldsymbol{b}$ along $\boldsymbol{a}$:

$$\operatorname{proj}_{\boldsymbol{a}} \boldsymbol{b} = \frac{\boldsymbol{b}^\top \boldsymbol{a}}{\boldsymbol{a}^\top \boldsymbol{a}}\, \boldsymbol{a} = \frac{3}{1}(1, 0) = (3, 0),$$

and the residual is $\boldsymbol{b} - \operatorname{proj}_{\boldsymbol{a}} \boldsymbol{b} = (0, 4)$. The residual is orthogonal to $\boldsymbol{a}$ ($(0,4)^\top(1,0) = 0$), as a projection requires; geometrically $(3, 0)$ is the foot of the perpendicular from $(3, 4)$ to the $x$-axis.

## 17.3 Eigenvalues, the spectral theorem, and the SVD

### Question 13: Eigendecomposition and a matrix power (20 points)

**Problem.** Let $A = \left[\begin{smallmatrix} 5 & 2 \\ 2 & 5 \end{smallmatrix}\right]$. (a) Find the eigenvalues. (b) Find the eigenvectors. (c) Write the diagonalization $A = PDP^{-1}$. (d) Use it to compute $A^2$.

**Solution.** *(a)* The matrix is symmetric. Its characteristic polynomial is $\det(A - \lambda I) = (5 - \lambda)^2 - 4 = 0$, so $5 - \lambda = \pm 2$ and the eigenvalues are $\lambda_1 = 7$, $\lambda_2 = 3$.

*(b)* For $\lambda = 7$, $(A - 7I) = \left[\begin{smallmatrix} -2 & 2 \\ 2 & -2 \end{smallmatrix}\right]$ gives $-2v_1 + 2v_2 = 0$, so $v_1 = v_2$ and $\boldsymbol{v}_1 = (1, 1)$. For $\lambda = 3$, $(A - 3I) = \left[\begin{smallmatrix} 2 & 2 \\ 2 & 2 \end{smallmatrix}\right]$ gives $v_1 = -v_2$ and $\boldsymbol{v}_2 = (1, -1)$. As the spectral theorem promises for a symmetric matrix, the eigenvectors are orthogonal: $\boldsymbol{v}_1^\top \boldsymbol{v}_2 = 1 - 1 = 0$.

*(c)* Place the eigenvectors in $P$ and the eigenvalues in $D$. Since the columns of $P$ are orthogonal with squared length 2, $P^{-1} = \frac{1}{2}\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$:

$$A = PDP^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 7 & 0 \\ 0 & 3 \end{bmatrix} \cdot \tfrac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

*(d)* Squaring only squares the eigenvalues, $D^2 = \operatorname{diag}(49, 9)$:

$$A^2 = PD^2P^{-1} = \tfrac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 49 & 0 \\ 0 & 9 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \tfrac{1}{2} \begin{bmatrix} 58 & 40 \\ 40 & 58 \end{bmatrix} = \begin{bmatrix} 29 & 20 \\ 20 & 29 \end{bmatrix}.$$

Direct squaring confirms it: $\left[\begin{smallmatrix} 5 & 2 \\ 2 & 5 \end{smallmatrix}\right]^2 = \left[\begin{smallmatrix} 25+4 & 10+10 \\ 10+10 & 4+25 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 29 & 20 \\ 20 & 29 \end{smallmatrix}\right]$.

### Question 14: Trace, determinant, eigenvalues (10 points)

**Problem.** For $A = \left[\begin{smallmatrix} 4 & -1 \\ 2 & 1 \end{smallmatrix}\right]$, find the eigenvalues and confirm the trace and determinant identities.

**Solution.** The trace is 5 and the determinant is $4(1) - (-1)(2) = 6$, so the characteristic polynomial is $\lambda^2 - 5\lambda + 6 = (\lambda - 3)(\lambda - 2)$, giving eigenvalues 3 and 2. Check: $\lambda_1 + \lambda_2 = 5 = \operatorname{tr} A$ and $\lambda_1 \lambda_2 = 6 = \det A$. Both identities hold, as they must for any square matrix, since the characteristic polynomial's coefficients are $-\operatorname{tr} A$ and $\det A$.

### Question 15: Eigenvalues of $A^{-1}$ and $A^2$ (10 points)

**Problem.** A matrix $A$ has eigenvalues 4 and 2. Without computing $A$, give the eigenvalues of $A^{-1}$ and of $A^2$.

**Solution.** Eigenvalues transform under functions of the matrix while the eigenvectors stay fixed. If $A\boldsymbol{v} = \lambda \boldsymbol{v}$ then $A^{-1}\boldsymbol{v} = \frac{1}{\lambda}\boldsymbol{v}$ and $A^2\boldsymbol{v} = \lambda^2 \boldsymbol{v}$. So $A^{-1}$ has eigenvalues $\frac{1}{4}$ and $\frac{1}{2}$ (reciprocals), and $A^2$ has eigenvalues 16 and 4 (squares). This is why a tiny eigenvalue of $A$ becomes a huge one of $A^{-1}$ (ill-conditioning) and why powers of $A$ are dominated by the largest eigenvalue (power iteration).

## Question 16: The Rayleigh quotient (10 points)

**Problem.** For $A = \left[\begin{smallmatrix} 3 & 2 \\ 2 & 3 \end{smallmatrix}\right]$, find the maximum and minimum of $\boldsymbol{x}^\top A\boldsymbol{x}$ over unit vectors, and the directions where they occur.

**Solution.** For a symmetric matrix the Rayleigh quotient ranges between the smallest and largest eigenvalues. From $(3-\lambda)^2 - 4 = 0$, i.e. $3-\lambda = \pm 2$, the eigenvalues are $\lambda_{\max} = 5$ and $\lambda_{\min} = 1$. So over unit vectors $\boldsymbol{x}^\top A\boldsymbol{x}$ reaches a maximum of 5 at the eigenvector for 5, namely $\frac{1}{\sqrt{2}}(1,1)$, and a minimum of 1 at $\frac{1}{\sqrt{2}}(1,-1)$. This variational characterization is exactly the principle behind PCA, where the top eigenvector of the covariance is the direction of maximum variance.

## Question 17: Definiteness of a quadratic form (15 points)

**Problem.** Consider $f(x,y) = 3x^2 + 2xy + 3y^2$. (a) Write its symmetric matrix. (b) Find the eigenvalues. (c) Classify the form. (d) Give the direction of slowest increase on the unit circle.

**Solution.** *(a)* Splitting the cross term evenly, $f = \boldsymbol{x}^\top M\boldsymbol{x}$ with $M = \left[\begin{smallmatrix} 3 & 1 \\ 1 & 3 \end{smallmatrix}\right]$.

*(b)* From $(3-\lambda)^2 - 1 = 0$, i.e. $3-\lambda = \pm 1$, the eigenvalues are $\lambda_1 = 4$ and $\lambda_2 = 2$.

*(c)* Both eigenvalues are positive, so $M$ is positive definite and $f > 0$ for every nonzero $\boldsymbol{x}$: the form is *positive definite*, a bowl opening upward.

*(d)* On the unit circle the form ranges between its eigenvalues, reaching its smallest value $\lambda_2 = 2$ along the eigenvector for $\lambda = 2$. That eigenvector solves $(M - 2I)\boldsymbol{v} = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]\boldsymbol{v} = \boldsymbol{0}$, giving the direction $\frac{1}{\sqrt{2}}(1,-1)$. So $f$ grows slowest along $(1,-1)$ and fastest along $(1,1)$, the eigenvector for $\lambda = 4$.

## Question 18: Definiteness by leading minors

**Problem.** Is the quadratic form with matrix $M = \left[\begin{smallmatrix} 3 & -1 \\ -1 & 3 \end{smallmatrix}\right]$ positive definite?

**Solution.** The eigenvalues solve $(3-\lambda)^2 - 1 = 0$, i.e. $3-\lambda = \pm 1$, giving $\lambda = 4$ and $\lambda = 2$. Both are positive, so $M$ is positive definite and $\boldsymbol{x}^\top M\boldsymbol{x} > 0$ for every nonzero $\boldsymbol{x}$. (A quicker check: the leading principal minors are $3 > 0$ and $\det M = 9 - 1 = 8 > 0$, Sylvester's criterion.)

## Question 19: A singular value decomposition (10 points)

**Problem.** Find the singular values of $A = \left[\begin{smallmatrix} 0 & 3 \\ 2 & 0 \end{smallmatrix}\right]$.

**Solution.** The squared singular values are the eigenvalues of $A^\top A = \left[\begin{smallmatrix} 0 & 2 \\ 3 & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} 0 & 3 \\ 2 & 0 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 4 & 0 \\ 0 & 9 \end{smallmatrix}\right]$. This is diagonal with eigenvalues 4 and 9, so the singular values are $\sigma_1 = 3$ and $\sigma_2 = 2$. The matrix stretches one axis by 3 and the other by 2 (and swaps them), so it maps the unit circle to an ellipse with semi-axes 3 and 2.

## Question 20: SVD and the pseudoinverse (20 points)

**Problem.** Let $A = \left[\begin{smallmatrix} 2 & 1 \\ 4 & 2 \end{smallmatrix}\right]$. (a) Find the rank. (b) Find the singular values from $A^\top A$. (c) Form the pseudoinverse $A^\dagger$. (d) Verify $AA^\dagger A = A$.

**Solution.** *(a)* The second row $(4,2)$ is twice the first $(2,1)$, so the rows are dependent and $\operatorname{rank} A = 1$.

*(b)* Compute the Gram matrix:

$$A^\top A = \begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 2 & 1 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 20 & 10 \\ 10 & 5 \end{bmatrix}.$$

Its eigenvalues solve $\lambda^2 - 25\lambda = 0$ (trace 25, determinant $100 - 100 = 0$), so $\lambda = 25$ and 0. The singular values are $\sigma_1 = \sqrt{25} = 5$ and $\sigma_2 = 0$, consistent with rank 1.

*(c)* The single nonzero singular value has right singular vector along the row direction, $\boldsymbol{v}_1 = \frac{1}{\sqrt{5}}(2,1)$, and left singular vector along the column direction, $\boldsymbol{u}_1 = \frac{1}{\sqrt{5}}(1,2)$. The pseudoinverse inverts that one value:

$$A^\dagger = \frac{1}{\sigma_1}\boldsymbol{v}_1\boldsymbol{u}_1^\top = \frac{1}{5}\cdot\frac{1}{5}\begin{bmatrix}2\\1\end{bmatrix}\begin{bmatrix}1 & 2\end{bmatrix} = \frac{1}{25}\begin{bmatrix}2 & 4\\1 & 2\end{bmatrix}.$$

*(d)* Since $A^\dagger = \frac{1}{25}\left[\begin{smallmatrix}2&4\\1&2\end{smallmatrix}\right]$, first $A^\dagger A = \frac{1}{25}\left[\begin{smallmatrix}2&4\\1&2\end{smallmatrix}\right]\left[\begin{smallmatrix}2&1\\4&2\end{smallmatrix}\right] = \frac{1}{25}\left[\begin{smallmatrix}20&10\\10&5\end{smallmatrix}\right] = \frac{1}{5}\left[\begin{smallmatrix}4&2\\2&1\end{smallmatrix}\right]$, and then $A(A^\dagger A) = \left[\begin{smallmatrix}2&1\\4&2\end{smallmatrix}\right]\cdot\frac{1}{5}\left[\begin{smallmatrix}4&2\\2&1\end{smallmatrix}\right] = \frac{1}{5}\left[\begin{smallmatrix}10&5\\20&10\end{smallmatrix}\right] = \left[\begin{smallmatrix}2&1\\4&2\end{smallmatrix}\right] = A$. The identity holds. The naive formula $(A^\top A)^{-1}A^\top$ fails here because $A^\top A$ is singular, but the SVD pseudoinverse exists and is unique.

### Question 21: Low-rank approximation (10 points)

**Problem.** A data matrix has singular values $\sigma = (8, 6)$. (a) What fraction of the Frobenius energy does the best rank-1 approximation keep? (b) What is its Frobenius error?

**Solution.** *(a)* The total energy is $\sum \sigma_i^2 = 64 + 36 = 100$, and the rank-1 fit keeps $\sigma_1^2 = 64$, a fraction $64/100 = 0.64$, that is 64%. *(b)* By Eckart–Young the squared error is the dropped singular value squared, $\sigma_2^2 = 36$, so $\|A - A_1\|_F = \sqrt{36} = 6$. The two singular values are close (8 and 6), so dropping the second loses a substantial 36% of the energy; low-rank approximation pays off only when the spectrum decays sharply, which it does not here.

### Question 22: Principal component analysis (10 points)

**Problem.** A dataset has covariance $\boldsymbol{\Sigma} = \left[\begin{smallmatrix}10&6\\6&10\end{smallmatrix}\right]$. (a) Find the principal directions and their variances. (b) What fraction of variance lies along the first component?

**Solution.** *(a)* From $(10-\lambda)^2 - 36 = 0$, i.e. $10 - \lambda = \pm 6$, the eigenvalues are $\lambda_1 = 16$ and $\lambda_2 = 4$. The eigenvector for 16 solves $(\boldsymbol{\Sigma} - 16I)\boldsymbol{v} = \left[\begin{smallmatrix}-6&6\\6&-6\end{smallmatrix}\right]\boldsymbol{v} = \boldsymbol{0}$, giving the 45° direction $\frac{1}{\sqrt{2}}(1,1)$; the second is $\frac{1}{\sqrt{2}}(1,-1)$ with variance 4. *(b)* The first component captures $\dfrac{16}{16+4} = 0.8$, that is 80% of the total variance, along $(1,1)$. The strong positive covariance tilts the data cloud along the diagonal, exactly the axis PCA selects.

## 17.4 Probability and Bayes

### Question 23: Bayes' theorem and a spam filter (15 points)

**Problem.** A mail stream is 40% spam. A certain word appears in 80% of spam messages and 10% of legitimate ones. (a) What fraction of all messages contain the word? (b) Given a message contains the word, what is the probability it is spam? (c) Interpret the shift from the prior.

**Solution.** *(a)* By total probability, summing over spam and ham,

$$\mathbb{P}(\text{word}) = \mathbb{P}(\text{word} \mid \text{spam})\mathbb{P}(\text{spam}) + \mathbb{P}(\text{word} \mid \text{ham})\mathbb{P}(\text{ham}) = 0.8(0.4) + 0.1(0.6) = 0.32 + 0.06 = 0.38.$$

*(b)* By Bayes,

$$\mathbb{P}(\text{spam} \mid \text{word}) = \frac{0.8(0.4)}{0.38} = \frac{0.32}{0.38} \approx 0.842.$$

*(c)* The word raises the spam probability from the 40% base rate to about 84%, because it is eight times more likely under spam (0.8 vs 0.1). The likelihood ratio 8:1 multiplies the prior odds 0.4:0.6 = 2:3 to posterior odds 16:3, i.e. $\frac{16}{19} \approx 0.842$. This is exactly how a naive-Bayes filter accumulates evidence word by word.

### Question 24: Bayes with three sources (10 points)

**Problem.** Three machines make a part: $A$ (50% of output, 1% defective), $B$ (30%, 2%), $C$ (20%, 4%). A part is defective; find $\mathbb{P}(C \mid \text{def})$.

**Solution.** Total probability of a defect:

$$\mathbb{P}(\text{def}) = 0.5(0.01) + 0.3(0.02) + 0.2(0.04) = 0.005 + 0.006 + 0.008 = 0.019.$$

Then Bayes for machine $C$:

$$\mathbb{P}(C \mid \text{def}) = \frac{0.008}{0.019} \approx 0.421.$$

Although $C$ makes only 20% of the parts, its high defect rate makes it responsible for 42% of the defects, the largest share. Bayes weighs volume against defect rate.

### Question 25: A naive-Bayes update with two words (10 points)

**Problem.** Spam is 30% of mail. Word 1 appears in 60% of spam and 10% of ham; word 2 in 70% of spam and 20% of ham. A message contains both words. Assuming conditional independence, find $\mathbb{P}(\text{spam} \mid \text{both})$.

**Solution.** Naive Bayes multiplies the per-word likelihoods. The spam and ham scores are

$$\text{spam} : 0.3(0.6)(0.7) = 0.126, \qquad \text{ham} : 0.7(0.1)(0.2) = 0.014.$$

Normalizing, $\mathbb{P}(\text{spam} \mid \text{both}) = \dfrac{0.126}{0.126 + 0.014} = \dfrac{0.126}{0.140} = 0.9$. Two spam-indicative words push the posterior from the 30% prior to 90%; the conditional-independence assumption is what lets the likelihoods simply multiply.

### Question 26: Conditional probability with two dice (10 points)

**Problem.** Two fair dice are rolled. (a) Find $\mathbb{P}(\text{sum} = 8)$. (b) Find $\mathbb{P}(\text{first die} = 3 \mid \text{sum} = 8)$.

**Solution.** *(a)* The sum is 8 for the outcomes $(2,6), (3,5), (4,4), (5,3), (6,2)$, which is 5 of the 36 equally likely pairs, so $\mathbb{P}(\text{sum} = 8) = \frac{5}{36}$.

*(b)* Among those five outcomes, exactly one has first die 3, namely $(3,5)$. By the definition of conditional probability,

$$\mathbb{P}(\text{first} = 3 \mid \text{sum} = 8) = \frac{\mathbb{P}(\text{first} = 3 \text{ and sum} = 8)}{\mathbb{P}(\text{sum} = 8)} = \frac{1/36}{5/36} = \frac{1}{5}.$$

Conditioning on the sum restricts the sample space to those five equally likely outcomes, and the first die is equally likely to be any of $2, 3, 4, 5, 6$ there, giving $\frac{1}{5}$.

### Question 27: A Markov chain's stationary distribution (15 points)

**Problem.** A two-state chain has transition matrix (rows sum to 1) $P = \left[\begin{smallmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{smallmatrix}\right]$. Find the stationary distribution $\boldsymbol{\pi}$ with $\boldsymbol{\pi} P = \boldsymbol{\pi}$.

**Solution.** Writing $\boldsymbol{\pi} = (\pi_1, \pi_2)$ with $\pi_1 + \pi_2 = 1$, the balance equation for the first state is $0.8\pi_1 + 0.4\pi_2 = \pi_1$. Substitute $\pi_2 = 1 - \pi_1$:

$$0.8\pi_1 + 0.4(1 - \pi_1) = \pi_1 \implies 0.4\pi_1 + 0.4 = \pi_1 \implies 0.4 = 0.6\pi_1 \implies \pi_1 = \tfrac{2}{3}.$$

So $\boldsymbol{\pi} = (\frac{2}{3}, \frac{1}{3})$. A check: the first component of $\boldsymbol{\pi} P$ is $\frac{2}{3}(0.8) + \frac{1}{3}(0.4) = \frac{1.6}{3} + \frac{0.4}{3} = \frac{2}{3}$, and the second is $\frac{2}{3}(0.2) + \frac{1}{3}(0.6) = \frac{0.4}{3} + \frac{0.6}{3} = \frac{1}{3}$, so $\boldsymbol{\pi} P = \boldsymbol{\pi}$. The stationary distribution is the left eigenvector of $P$ for eigenvalue 1, the long-run fraction of time spent in each state, and it is exactly what PageRank computes on the web's link matrix.

## 17.5 Estimation, distributions, and covariance

## Question 28: Expectation and variance (15 points)

**Problem.** A fair four-sided die shows $\{2, 4, 6, 8\}$ with equal probability. (a) Find $\mathbb{E}[X]$. (b) Find $\mathrm{Var}(X)$. (c) Let $Y = 3X - 1$; find $\mathbb{E}[Y]$ and $\mathrm{Var}(Y)$.

**Solution.** *(a)* Each outcome has probability $\frac{1}{4}$, so

$$\mathbb{E}[X] = \tfrac{1}{4}(2 + 4 + 6 + 8) = \tfrac{20}{4} = 5.$$

*(b)* The second moment is $\mathbb{E}[X^2] = \frac{1}{4}(4 + 16 + 36 + 64) = \frac{120}{4} = 30$, so

$$\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = 30 - 25 = 5.$$

*(c)* An affine transform scales the mean affinely and the variance by the squared multiplier:

$$\mathbb{E}[Y] = 3\mathbb{E}[X] - 1 = 3(5) - 1 = 14, \qquad \mathrm{Var}(Y) = 3^2\,\mathrm{Var}(X) = 9(5) = 45.$$

The constant $-1$ shifts the center but never the spread, and the factor 3 triples the spread, so the variance grows ninefold. The standard deviation of $Y$ is $\sqrt{45} \approx 6.7$, three times that of $X$ ($\sqrt{5} \approx 2.24$), as scaling by 3 predicts.

## Question 29: Binomial mean and variance (10 points)

**Problem.** Let $X \sim \mathrm{Binomial}(12, \frac{1}{2})$. Find $\mathbb{E}[X]$ and $\mathrm{Var}(X)$.

**Solution.** A binomial sums $n$ independent Bernoulli trials, so $\mathbb{E}[X] = np = 12(\frac{1}{2}) = 6$ and $\mathrm{Var}(X) = np(1-p) = 12(\frac{1}{2})(\frac{1}{2}) = 3$, giving standard deviation $\sqrt{3} \approx 1.73$. On average 6 of 12 fair flips are heads, give or take about 1.7. The variance is largest at $p = \frac{1}{2}$, the most uncertain bias.

## Question 30: Maximum likelihood for an exponential (10 points)

**Problem.** Waiting times $\{2, 4, 6\}$ are modeled as iid exponential with rate $\lambda$. Find the MLE.

**Solution.** The exponential density is $\lambda e^{-\lambda x}$, so the log-likelihood for $n$ samples is $\ell(\lambda) = n \ln \lambda - \lambda \sum_i x_i$. Differentiating, $\ell'(\lambda) = \dfrac{n}{\lambda} - \sum_i x_i = 0$ gives $\hat{\lambda} = \dfrac{n}{\sum_i x_i} = \dfrac{1}{\bar{x}}$. With $\bar{x} = \frac{2+4+6}{3} = 4$,

$$\hat{\lambda} = \frac{1}{4} = 0.25.$$

The MLE rate is the reciprocal of the sample mean, sensible because the exponential's mean is $1/\lambda$; matching the model mean to the data mean recovers $\lambda$.

## Question 31: Multinomial maximum likelihood (10 points)

**Problem.** A three-category variable is observed 100 times with counts $(20, 30, 50)$. Find the MLE of the category probabilities.

**Solution.** Maximizing the multinomial log-likelihood $\sum_k m_k \ln p_k$ subject to $\sum_k p_k = 1$ (with a Lagrange multiplier) gives each probability as its observed frequency, $\hat{p}_k = m_k / N$:

$$\hat{\boldsymbol{p}} = \left(\tfrac{20}{100}, \tfrac{30}{100}, \tfrac{50}{100}\right) = (0.2, 0.3, 0.5).$$

The constraint is what ties the probabilities together so they sum to one; without it each $p_k$ would run to its unconstrained maximum. As for the coin, the MLE is the empirical frequency.

## Question 32: A Gaussian under an affine map (10 points)

**Problem.** Let $X \sim \mathcal{N}(1, 4)$ and $Y = 3X - 2$. Find the distribution of $Y$.

**Solution.** An affine function of a Gaussian is Gaussian. The mean maps affinely, $\mathbb{E}[Y] = 3(1) - 2 = 1$, and the variance scales by the squared multiplier, $\mathrm{Var}(Y) = 3^2 \mathrm{Var}(X) = 9(4) = 36$. So $Y \sim \mathcal{N}(1, 36)$, with standard deviation 6. The constant $-2$ shifts the center without touching the spread, while the factor 3 triples the standard deviation and so multiplies the variance by nine.

## Question 33: Covariance to correlation (10 points)

**Problem.** Two variables have covariance matrix $\left[\begin{smallmatrix} 16 & 8 \\ 8 & 9 \end{smallmatrix}\right]$. Find the correlation coefficient.

**Solution.** The variances are $\sigma_X^2 = 16$ and $\sigma_Y^2 = 9$, so $\sigma_X = 4$ and $\sigma_Y = 3$. The covariance is the off-diagonal 8, so

$$\rho = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{8}{4 \cdot 3} = \frac{8}{12} = \frac{2}{3} \approx 0.667.$$

A moderate-to-strong positive correlation. The correlation rescales the unit-laden covariance into a pure number in $[-1, 1]$, comparable across problems regardless of the variables' scales.

## Question 34: Sample covariance (10 points)

**Problem.** For the points $(1, 1), (2, 2), (3, 4)$, compute the sample covariance matrix (dividing by $n = 3$).

**Solution.** The mean is $(\bar{x}, \bar{y}) = (2, \frac{7}{3})$, so the centered points are $(-1, -\frac{4}{3}), (0, -\frac{1}{3}), (1, \frac{5}{3})$. Then

$$\mathrm{Var}(x) = \tfrac{1+0+1}{3} = \tfrac{2}{3}, \quad \mathrm{Var}(y) = \tfrac{(4/3)^2+(1/3)^2+(5/3)^2}{3} = \tfrac{14}{9}, \quad \mathrm{Cov} = \tfrac{(-1)(-\frac{4}{3})+0+(1)(\frac{5}{3})}{3} = 1.$$

So $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 2/3 & 1 \\ 1 & 14/9 \end{smallmatrix}\right]$. The positive off-diagonal confirms $x$ and $y$ rise together, as the upward drift of the three points shows.

CHAPTER 18

# Practice Final, Worked in Full

This practice final spans the whole course, from the spectral theorem and the SVD through maximum likelihood, optimization, the support vector machine, and PCA. Each question carries its point value and is solved completely, with every intermediate quantity shown and checked. Attempt each closed-book under time before reading the solution; the goal is to connect the threads, since the final rewards seeing that least squares, the Gaussian, and PCA are one circle of ideas.

## 18.1 Linear algebra and matrix decompositions

### Question 1: The spectral theorem in action (15 points)

**Problem.** Let $A = \left[\begin{smallmatrix} 4 & 1 \\ 1 & 4 \end{smallmatrix}\right]$. (a) Find its eigenvalues and orthonormal eigenvectors. (b) Write the spectral decomposition $A = \lambda_1 \boldsymbol{q}_1 \boldsymbol{q}_1^\top + \lambda_2 \boldsymbol{q}_2 \boldsymbol{q}_2^\top$. (c) Use it to state $\sqrt{A}$, the symmetric positive-definite square root.

**Solution.** *(a)* From $(4-\lambda)^2 - 1 = 0$, $\lambda_1 = 5$ and $\lambda_2 = 3$. For $\lambda = 5$, $(A - 5I) = \left[\begin{smallmatrix} -1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$ gives $\boldsymbol{q}_1 = \frac{1}{\sqrt{2}}(1,1)$; for $\lambda = 3$, $\boldsymbol{q}_2 = \frac{1}{\sqrt{2}}(1,-1)$, and $\boldsymbol{q}_1^\top \boldsymbol{q}_2 = 0$.

*(b)* The outer products are $\boldsymbol{q}_1 \boldsymbol{q}_1^\top = \frac{1}{2}\left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$ and $\boldsymbol{q}_2 \boldsymbol{q}_2^\top = \frac{1}{2}\left[\begin{smallmatrix} 1 & -1 \\ -1 & 1 \end{smallmatrix}\right]$, so

$$A = 5 \cdot \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + 3 \cdot \tfrac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \tfrac{1}{2}\begin{bmatrix} 8 & 2 \\ 2 & 8 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}. \checkmark$$

*(c)* A function of a symmetric matrix acts on its eigenvalues: replace each $\lambda_i$ by $\sqrt{\lambda_i}$ in the spectral sum. Thus

$$\sqrt{A} = \sqrt{5}\,\boldsymbol{q}_1 \boldsymbol{q}_1^\top + \sqrt{3}\,\boldsymbol{q}_2 \boldsymbol{q}_2^\top = \tfrac{1}{2}\begin{bmatrix} \sqrt{5}+\sqrt{3} & \sqrt{5}-\sqrt{3} \\ \sqrt{5}-\sqrt{3} & \sqrt{5}+\sqrt{3} \end{bmatrix}.$$

One checks $(\sqrt{A})^2 = A$ because the projectors are orthogonal and idempotent. Matrix square roots like this are what whiten data and define the geometry of the Gaussian.

### Question 2: Diagonalizing a $2 \times 2$ (10 points)

**Problem.** Diagonalize $A = \left[\begin{smallmatrix} 2 & 2 \\ 1 & 3 \end{smallmatrix}\right]$.

**Solution.** The characteristic polynomial is $\lambda^2 - 5\lambda + 4 = (\lambda - 4)(\lambda - 1)$ (trace 5, determinant $6 - 2 = 4$), so the eigenvalues are 4 and 1. For $\lambda = 4$, $(A - 4I) = \left[\begin{smallmatrix} -2 & 2 \\ 1 & -1 \end{smallmatrix}\right]$ gives $\boldsymbol{v}_1 = (1,1)$; for $\lambda = 1$, $\left[\begin{smallmatrix} 1 & 2 \\ 1 & 2 \end{smallmatrix}\right]$ gives $\boldsymbol{v}_2 = (2,-1)$. So $A = PDP^{-1}$ with $P = \left[\begin{smallmatrix} 1 & 2 \\ 1 & -1 \end{smallmatrix}\right]$, $D = \operatorname{diag}(4,1)$, and $P^{-1} = \frac{1}{3}\left[\begin{smallmatrix} 1 & 2 \\ 1 & -1 \end{smallmatrix}\right]$ (since $\det P = -3$). The matrix is not symmetric, so the eigenvectors are independent but not orthogonal.

### Question 3: Power iteration (10 points)

**Problem.** Run three steps of power iteration on $A = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ from $\boldsymbol{x}_0 = (1,0)$, and say what the Rayleigh quotient converges to.

**Solution.** Apply $A$ repeatedly: $(1,0) \to (2,1) \to (5,4) \to (14,13)$. The Rayleigh quotient $\boldsymbol{x}^\top A\boldsymbol{x}/\boldsymbol{x}^\top\boldsymbol{x}$ runs $2.0 \to 2.8 \to 2.976 \to \cdots$, climbing toward 3, the dominant eigenvalue (the eigenvalues are 3 and 1). The direction

tilts toward $(1,1)$, the top eigenvector, and the error shrinks by the ratio $|\lambda_2/\lambda_1| = 1/3$ each step. This is the engine behind PageRank and large-scale PCA.

### Question 4: Low-rank approximation (15 points)

**Problem.** A matrix has singular values $\sigma = (10, 6, 2)$. (a) State the best rank-2 approximation's Frobenius error. (b) What fraction of the Frobenius energy does it retain? (c) By Eckart–Young, could any rank-2 matrix do better?

**Solution.** *(a)* The best rank-2 truncation drops only the smallest singular value, so the squared error is $\sigma_3^2 = 4$ and $\|A - A_2\|_F = \sqrt{4} = 2$.

*(b)* The total energy is $\sum \sigma_i^2 = 100 + 36 + 4 = 140$, and the rank-2 fit keeps $100 + 36 = 136$, a fraction $136/140 \approx 0.971$, about 97%.

*(c)* No. The Eckart–Young theorem states that the truncated SVD is the optimal low-rank approximation in both the Frobenius and spectral norms, so no rank-2 matrix can beat the error 2. The discarded energy $\sigma_3^2$ is the unavoidable price of dropping to rank 2.

### Question 5: Low-rank energy (10 points)

**Problem.** A matrix has singular values $\sigma = (8, 4, 2)$. What fraction of the Frobenius energy do the best rank-1 and rank-2 approximations retain?

**Solution.** The total energy is $\sum \sigma_i^2 = 64 + 16 + 4 = 84$. The rank-1 fit keeps $\sigma_1^2 = 64$, a fraction $64/84 \approx 0.762$ (76%); the rank-2 fit keeps $64 + 16 = 80$, a fraction $80/84 \approx 0.952$ (95%). By Eckart–Young the truncated SVD is optimal, so no other rank-2 matrix does better, and the discarded $\sigma_3^2 = 4$ is the unavoidable squared error. The sharp drop after $\sigma_1$ is what makes the low-rank approximation faithful.

### Question 6: Condition number (5 points)

**Problem.** A matrix has singular values $\sigma = (6, 2)$. Give its 2-norm condition number and the worst-case error amplification when solving a linear system.

**Solution.** The condition number is $\kappa_2 = \sigma_{\max}/\sigma_{\min} = 6/2 = 3$. Solving $A\boldsymbol{x} = \boldsymbol{b}$ can amplify a relative error in $\boldsymbol{b}$ by at most $\kappa_2$: a 1% data error becomes at most a 3% solution error. A condition number near 1 is benign; a large one (tiny $\sigma_{\min}$) signals a nearly singular, sensitive problem, the numerical reason regularization lifts the small singular values.

### Question 7: Three norms (5 points)

**Problem.** For $\boldsymbol{x} = (2, 3)$, compute $\|\boldsymbol{x}\|_1$, $\|\boldsymbol{x}\|_2$, and $\|\boldsymbol{x}\|_\infty$.

**Solution.** $\|\boldsymbol{x}\|_1 = 2 + 3 = 5$, $\|\boldsymbol{x}\|_2 = \sqrt{4 + 9} = \sqrt{13} \approx 3.61$, and $\|\boldsymbol{x}\|_\infty = \max(2, 3) = 3$. The ordering $\|\cdot\|_\infty \le \|\cdot\|_2 \le \|\cdot\|_1$ holds, as always. These are the norms behind worst-case ($\ell_\infty$), ridge ($\ell_2$), and lasso/sparsity ($\ell_1$) reasoning.

## 18.2 Least squares, regression, and PCA

### Question 8: Ridge regression (15 points)

**Problem.** Ridge regression solves $(A^\top A + \lambda I)\boldsymbol{\beta} = A^\top \boldsymbol{y}$. With $A^\top A = \left[\begin{smallmatrix} 4 & 0 \\ 0 & 4 \end{smallmatrix}\right]$, $A^\top \boldsymbol{y} = (8, 8)$, compare the ordinary least-squares solution with the ridge solution at $\lambda = 4$, and explain the effect.

**Solution.** Ordinary least squares solves $A^\top A\boldsymbol{\beta} = A^\top \boldsymbol{y}$, i.e. $4\boldsymbol{\beta} = (8, 8)$, giving $\boldsymbol{\beta}_{\text{OLS}} = (2, 2)$. Ridge adds $\lambda I = 4I$:

$$(A^\top A + 4I)\boldsymbol{\beta} = (8I)\boldsymbol{\beta} = (8, 8) \implies \boldsymbol{\beta}_{\text{ridge}} = (1, 1).$$

The penalty halved the coefficients, from $(2, 2)$ to $(1, 1)$. Ridge lifts every eigenvalue of $A^\top A$ by $\lambda$ (here $4 \to 8$), which shrinks the estimate toward zero and cures ill-conditioning by pushing small eigenvalues away from zero. This

trades a little bias for a large drop in variance, and corresponds to a Gaussian prior on $\boldsymbol{\beta}$, the MAP estimate of Chapter 2. It is the bridge from least squares to regularized learning.

### Question 9: Gradient of a quadratic form (10 points)

**Problem.** For the quadratic form $f(\boldsymbol{x}) = \boldsymbol{x}^\top A\boldsymbol{x}$ with the symmetric $A = \left[\begin{smallmatrix} 3 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, find $\nabla f$ at $(1, 1)$.

**Solution.** For symmetric $A$, $\nabla(\boldsymbol{x}^\top A\boldsymbol{x}) = 2A\boldsymbol{x}$. At $(1, 1)$,

$$\nabla f = 2\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = 2\begin{bmatrix} 4 \\ 3 \end{bmatrix} = (8, 6).$$

This identity is the matrix-calculus engine behind the normal equations: differentiating $\|A\boldsymbol{x} - \boldsymbol{b}\|^2$ uses exactly $\nabla(\boldsymbol{x}^\top A^\top A\boldsymbol{x}) = 2A^\top A\boldsymbol{x}$, and setting it to zero gives $A^\top A\boldsymbol{x} = A^\top \boldsymbol{b}$.

### Question 10: Principal component analysis (15 points)

**Problem.** A dataset has covariance $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 6 & 2 \\ 2 & 3 \end{smallmatrix}\right]$. (a) Find the principal directions and their variances. (b) What fraction of variance does the first component capture? (c) If you project onto the first component, what is the average squared reconstruction error?

**Solution.** *(a)* The eigenvalues solve $\lambda^2 - 9\lambda + 14 = 0$ (trace 9, determinant $6 \cdot 3 - 2 \cdot 2 = 14$), giving $\lambda_1 = 7$ and $\lambda_2 = 2$. For $\lambda = 7$, $(\boldsymbol{\Sigma} - 7I) = \left[\begin{smallmatrix} -1 & 2 \\ 2 & -4 \end{smallmatrix}\right]$ gives the direction $\boldsymbol{v}_1 = \frac{1}{\sqrt{5}}(2, 1)$; the second principal direction is the orthogonal $\boldsymbol{v}_2 = \frac{1}{\sqrt{5}}(-1, 2)$, with variance 2.

*(b)* The first component captures $\dfrac{\lambda_1}{\lambda_1 + \lambda_2} = \dfrac{7}{9} \approx 0.778$, about 78% of the total variance, along $(2, 1)$.

*(c)* Projecting onto the first component discards the second, so the average squared reconstruction error equals the discarded eigenvalue, $\lambda_2 = 2$. This is the Eckart–Young statement for PCA: dropping a component costs exactly its variance, the same $\lambda_2$ that the per-point error $y_2^2$ averages to. With 7 of the 9 total variance retained, the one-dimensional summary keeps most of the structure while halving the storage.

### Question 11: PCA reconstruction error (10 points)

**Problem.** With top principal component $\boldsymbol{v}_1 = \frac{1}{\sqrt{2}}(1, 1)$, project $\boldsymbol{x} = (2, 1)$ onto $\boldsymbol{v}_1$, reconstruct, and give the squared reconstruction error.

**Solution.** The first coordinate is $y_1 = \boldsymbol{v}_1^\top \boldsymbol{x} = \frac{3}{\sqrt{2}}$, and the reconstruction is $\hat{\boldsymbol{x}} = y_1\boldsymbol{v}_1 = \frac{3}{2}(1, 1) = (1.5, 1.5)$. The error is $\boldsymbol{x} - \hat{\boldsymbol{x}} = (0.5, -0.5)$ with squared length $0.25 + 0.25 = 0.5$. This equals the squared discarded coordinate $y_2^2$ with $y_2 = \boldsymbol{v}_2^\top \boldsymbol{x} = \frac{1}{\sqrt{2}}$, and averaged over a dataset it equals the discarded eigenvalue $\lambda_2$, the Eckart–Young statement for PCA.

## 18.3 Probability, distributions, and learning theory

### Question 12: Maximum likelihood and MAP (15 points)

**Problem.** A coin shows 8 heads in 12 flips. (a) Give the MLE of the bias. (b) With a Beta$(3, 3)$ prior, give the posterior, its mean, and the MAP estimate. (c) Explain the ordering of the three numbers.

**Solution.** *(a)* The MLE is the sample frequency $\hat{\mu}_{\text{MLE}} = \frac{8}{12} = \frac{2}{3} \approx 0.667$.

*(b)* The Beta is conjugate, so the posterior adds the counts as pseudo-observations: $\text{Beta}(3 + 8,\ 3 + 4) = \text{Beta}(11, 7)$. Its mean is $\frac{11}{18} \approx 0.611$, and the MAP (the mode) is $\frac{a' - 1}{a' + b' - 2} = \frac{10}{16} = 0.625$.

*(c)* The prior Beta$(3, 3)$ is centered at 0.5, so it pulls the estimate down from the data's 0.667 toward 0.5. The posterior mean 0.611 is pulled most (it averages over the whole posterior), the MAP 0.625 less so (it sits at the peak), and the MLE 0.667 ignores the prior entirely. All three converge to $m/n$ as data accumulates and the prior's fixed pseudo-counts become negligible.

## Question 13: A medical test (10 points)

**Problem.** A disease affects 1% of people. A test is 95% sensitive and 90% specific. Given a positive result, what is the probability of disease?

**Solution.** The probability of testing positive is, by total probability,

$$\mathbb{P}(+) = 0.95(0.01) + 0.10(0.99) = 0.0095 + 0.099 = 0.1085,$$

using $\mathbb{P}(+ \mid \text{healthy}) = 1 - 0.90 = 0.10$. By Bayes,

$$\mathbb{P}(D \mid +) = \frac{0.0095}{0.1085} \approx 0.088.$$

Even after a positive on a fairly accurate test, the patient is only about 9% likely to be sick, because the disease is rare: the $0.99 \times 0.10$ false alarms swamp the $0.01 \times 0.95$ true detections. This base-rate effect is the central caution of screening.

## Question 14: A sum of independent Gaussians (5 points)

**Problem.** Let $X \sim \mathcal{N}(2, 3)$ and $Y \sim \mathcal{N}(1, 5)$ be independent. Find the distribution of $X + Y$.

**Solution.** Independent Gaussians add to a Gaussian, with means and variances each summing: $X + Y \sim \mathcal{N}(2+1,\ 3+5) = \mathcal{N}(3, 8)$, standard deviation $\sqrt{8} \approx 2.83$. The spreads combine in quadrature, not by adding standard deviations; independence is what kills the cross-covariance term.

## Question 15: A Markov stationary distribution (10 points)

**Problem.** Find the stationary distribution of the chain with transition matrix (rows sum to 1) $P = \left[\begin{smallmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{smallmatrix}\right]$.

**Solution.** The balance equation $0.9\pi_1 + 0.3\pi_2 = \pi_1$ with $\pi_2 = 1 - \pi_1$ gives $0.9\pi_1 + 0.3(1 - \pi_1) = \pi_1$, i.e. $0.3 = 0.4\pi_1$, so $\pi_1 = 0.75$ and $\pi_2 = 0.25$. Thus $\boldsymbol{\pi} = (\frac{3}{4}, \frac{1}{4})$, the left eigenvector of $P$ for eigenvalue 1 and the long-run fraction of time in each state. This is the computation PageRank runs on the web's link matrix.

## Question 16: Covariance to correlation (5 points)

**Problem.** Two variables have covariance matrix $\left[\begin{smallmatrix} 25 & 12 \\ 12 & 16 \end{smallmatrix}\right]$. Find the correlation coefficient.

**Solution.** The standard deviations are $\sigma_X = \sqrt{25} = 5$ and $\sigma_Y = \sqrt{16} = 4$, so

$$\rho = \frac{\operatorname{Cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{12}{5 \cdot 4} = \frac{12}{20} = 0.6,$$

a moderate positive linear relationship. The correlation strips the units out of the covariance, landing in $[-1, 1]$ regardless of the variables' scales.

## Question 17: Entropy (5 points)

**Problem.** Compute the entropy of the distribution $(0.6, 0.4)$ in bits.

**Solution.** $H = -0.6 \log_2 0.6 - 0.4 \log_2 0.4 = -0.6(-0.737) - 0.4(-1.322) \approx 0.442 + 0.529 = 0.971$ bits. This is just below the maximum of 1 bit (reached at 0.5/0.5): the slight imbalance toward the first outcome makes the variable a touch more predictable. Binary entropy peaks at $p = \frac{1}{2}$ and falls to 0 at the certain extremes, mirroring the Bernoulli variance $p(1 - p)$.

## Question 18: A Kullback–Leibler divergence (10 points)

**Problem.** Compute $\mathrm{KL}(p \,\|\, q)$ in nats for the distributions $p = (0.5, 0.5)$ and $q = (0.25, 0.75)$.

**Solution.** The KL divergence is $\sum_k p_k \ln \dfrac{p_k}{q_k}$:

$$\mathrm{KL}(p\|q) = 0.5 \ln \frac{0.5}{0.25} + 0.5 \ln \frac{0.5}{0.75} = 0.5 \ln 2 + 0.5 \ln \tfrac{2}{3} \approx 0.5(0.693) + 0.5(-0.405) = 0.144.$$

It is positive (zero only when $p = q$) and asymmetric: $\mathrm{KL}(q\|p)$ differs. It measures the extra nats to encode samples from $p$ with a code built for $q$, the quantity maximum likelihood implicitly minimizes between data and model and that the Chernoff exponent uses.

## Question 19: A concentration bound (10 points)

**Problem.** A bounded loss in $[0, 1]$ is averaged over $n = 200$ independent test points. Use Hoeffding to bound the probability the empirical average is off by at least $\varepsilon = 0.1$ from the true mean.

**Solution.** Hoeffding gives $\mathbb{P}(|\bar{X} - \mu| \geq \varepsilon) \leq 2e^{-2n\varepsilon^2}$:

$$2e^{-2(200)(0.1)^2} = 2e^{-4} \approx 2(0.0183) = 0.037.$$

So with about 96% confidence the test estimate is within 0.1 of the truth. This is the generalization guarantee in miniature: a held-out average of 200 points pins the true risk to a tight band, and the band shrinks exponentially in $n$.

## Question 20: A Hoeffding bound (10 points)

**Problem.** A bounded loss in $[0, 1]$ is averaged over $n = 200$ independent points. Bound the probability the empirical average is off by at least $\varepsilon = 0.1$.

**Solution.** Hoeffding's inequality gives $\mathbb{P}(|\bar{X} - \mu| \geq \varepsilon) \leq 2e^{-2n\varepsilon^2} = 2e^{-2(200)(0.01)} = 2e^{-4} \approx 0.037$. So with about 96% confidence the held-out estimate is within 0.1 of the true risk. The bound decays exponentially in $n$, the quantitative form of the law of large numbers and the basis of generalization guarantees.

## Question 21: VC dimension of rectangles (10 points)

**Problem.** Find the VC dimension of axis-aligned rectangles in the plane (a point is labeled 1 if it lies inside the rectangle).

**Solution.** Place four points as a diamond: one topmost, one bottommost, one leftmost, one rightmost. Any subset can be realized by taking the axis-aligned bounding box of the points to include, which excludes the rest, so all $2^4 = 16$ labelings are achievable and $d_{\mathrm{VC}} \geq 4$. Five points cannot be shattered: among any five, take the one that is not extreme in any of the four directions; any rectangle containing the other four extremes must contain the whole bounding box, hence this interior point too, so the labeling excluding only it is impossible. Therefore $d_{\mathrm{VC}} = 4$, matching the four free parameters (the rectangle's four edges).

## Question 22: The bias–variance decomposition (10 points)

**Problem.** An estimator has squared bias 0.09, variance 0.04, and the data carries irreducible noise of variance 0.05. (a) Give the expected test error. (b) Describe how it would change if the model were made more flexible.

**Solution.** *(a)* The expected squared error splits into three nonnegative pieces:

$$\mathrm{MSE} = \mathrm{bias}^2 + \mathrm{variance} + \mathrm{noise} = 0.09 + 0.04 + 0.05 = 0.18.$$

*(b)* A more flexible model fits the training data more closely, lowering the bias (here 0.09) but raising the variance (here 0.04), since it chases noise. The noise floor 0.05 is irreducible: no model beats it. The test error is minimized at the flexibility where the falling bias and rising variance balance, the heart of the bias–variance tradeoff and the reason the best model is neither the simplest nor the most complex.

## 18.4 Vector calculus and optimization

### Question 23: Newton's method (10 points)

**Problem.** Use Newton's method on $f(x) = x^2 - 5$ from $x_0 = 2$ to approximate $\sqrt{5}$. Trace three iterations and comment on the convergence rate.

**Solution.** With $f'(x) = 2x$, the update $x_{t+1} = x_t - \dfrac{x_t^2 - 5}{2x_t} = \frac{1}{2}\Big(x_t + \dfrac{5}{x_t}\Big)$. From $x_0 = 2$:

$$x_1 = \tfrac{1}{2}(2 + 2.5) = 2.25, \quad x_2 = \tfrac{1}{2}\big(2.25 + \tfrac{5}{2.25}\big) = 2.23611, \quad x_3 = 2.236068,$$

already matching $\sqrt{5} = 2.2360679\ldots$ to six digits. The number of correct digits roughly doubles each step (quadratic convergence) once near the root, because Newton models $f$ by its tangent and the error obeys $e_{t+1} \approx c\, e_t^2$. Compared with gradient descent's geometric (linear) rate, Newton is dramatically faster near a solution, at the cost of needing the second derivative.

### Question 24: Gradient-descent step size (10 points)

**Problem.** For $f(x) = \frac{1}{2}x^2$, gradient descent updates $x \leftarrow x - \alpha x$. For which step sizes $\alpha$ does it converge, and what happens at $\alpha = 0.5$ versus $\alpha = 2.5$?

**Solution.** The update is $x_{t+1} = (1 - \alpha)x_t$, so $x_t = (1 - \alpha)^t x_0$, which decays to zero exactly when $|1 - \alpha| < 1$, i.e. $0 < \alpha < 2$. At $\alpha = 0.5$ the factor is $0.5$, so each step halves the iterate and it converges smoothly. At $\alpha = 2.5$ the factor is $1 - 2.5 = -1.5$, whose magnitude exceeds 1, so the iterate *diverges*, oscillating with growing amplitude. The stable bound $\alpha < 2/L$ with curvature $L = f'' = 1$ is the quadratic prototype of every learning-rate limit; overshooting it is the classic "rate too large" failure.

### Question 25: One backpropagation step (15 points)

**Problem.** A tiny network computes $z = wx$, $a = \sigma(z)$, and loss $L = \frac{1}{2}(a - y)^2$, with $\sigma$ the logistic sigmoid. For $w = 0.5$, $x = 2$, $y = 1$, compute the forward pass and then $\partial L / \partial w$ by backpropagation.

**Solution.** *Forward.* $z = wx = 0.5(2) = 1$, then $a = \sigma(1) = \dfrac{1}{1 + e^{-1}} \approx 0.731$, and $L = \frac{1}{2}(0.731 - 1)^2 = \frac{1}{2}(0.0724) \approx 0.0362$. *Backward.* Chain the three local derivatives from the loss back to $w$:

$$\frac{\partial L}{\partial a} = a - y = -0.269, \quad \frac{\partial a}{\partial z} = \sigma(z)(1 - \sigma(z)) = 0.731(0.269) = 0.197, \quad \frac{\partial z}{\partial w} = x = 2.$$

Multiplying,

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w} = (-0.269)(0.197)(2) \approx -0.106.$$

The negative gradient means increasing $w$ lowers the loss, sensible since the target 1 exceeds the prediction 0.731, so the network should push its output up. Backpropagation is exactly this product of local derivatives, computed right to left.

### Question 26: A Jacobian (10 points)

**Problem.** For $\boldsymbol{F}(x, y) = (x + y^2,\ x^2 - y)$, compute the Jacobian at $(1, 2)$ and its determinant.

**Solution.** The Jacobian stacks the gradients of each output as rows:

$$\boldsymbol{J} = \begin{bmatrix} \partial(x + y^2)/\partial x & \partial(x + y^2)/\partial y \\ \partial(x^2 - y)/\partial x & \partial(x^2 - y)/\partial y \end{bmatrix} = \begin{bmatrix} 1 & 2y \\ 2x & -1 \end{bmatrix}.$$

At $(1, 2)$ this is $\left[\begin{smallmatrix} 1 & 4 \\ 2 & -1 \end{smallmatrix}\right]$, with determinant $1(-1) - 4(2) = -9$. The Jacobian is the best linear approximation of $\boldsymbol{F}$ near the point, and its determinant $-9$ is the local area-scaling factor (the negative sign flags an orientation reversal).

## Question 27: Constrained optimization by Lagrange multipliers (10 points)

**Problem.** Minimize $f(x, y) = x^2 + 2y^2$ subject to $x + y = 3$.

**Solution.** At the constrained minimum $\nabla f$ is parallel to the constraint gradient: $(2x,\ 4y) = \lambda(1, 1)$. So $2x = \lambda$ and $4y = \lambda$, hence $2x = 4y$, i.e. $x = 2y$. Substituting into $x + y = 3$ gives $2y + y = 3$, so $y = 1$, $x = 2$, and $\lambda = 4$. The minimum value is $f(2, 1) = 4 + 2 = 6$. The asymmetry $x = 2y$ comes from the heavier penalty on $y$ (coefficient 2): the optimizer leans toward spending its budget on the cheaper variable $x$. The multiplier $\lambda = 4$ is the rate at which the optimum would change if the constraint level 3 were relaxed.

## Question 28: A Lagrange multiplier (10 points)

**Problem.** Minimize $x^2 + y^2$ subject to $3x + 4y = 25$.

**Solution.** At the constrained minimum $\nabla(x^2 + y^2) = \lambda\nabla(3x + 4y)$, i.e. $(2x, 2y) = \lambda(3, 4)$, so $x = \frac{3\lambda}{2}$, $y = 2\lambda$. Substituting into the constraint, $3(\frac{3\lambda}{2}) + 4(2\lambda) = \frac{25\lambda}{2} = 25$, so $\lambda = 2$, giving $(x, y) = (3, 4)$ and minimum value $9 + 16 = 25$. Geometrically $(3, 4)$ is the point of the line closest to the origin, and $\sqrt{25} = 5$ is that distance; the squared distance from the origin to $3x + 4y = 25$ is $25^2/(3^2 + 4^2) = 625/25 = 25$.

## Question 29: KKT with two active constraints (15 points)

**Problem.** Minimize $x^2 + y^2$ subject to $x \geq 1$ and $y \geq 2$.

**Solution.** The unconstrained minimum is $(0, 0)$, which violates both constraints, so test them active. With $x = 1$ and $y = 2$ both binding, the candidate is $(1, 2)$ with value $1^2 + 2^2 = 5$. Verify KKT: writing the constraints as $g_1 = 1 - x \leq 0$, $g_2 = 2 - y \leq 0$, stationarity $\nabla f = \mu_1\nabla(-g_1) + \mu_2\nabla(-g_2)$ reads $(2x, 2y) = (\mu_1, \mu_2)$, so $\mu_1 = 2x = 2$ and $\mu_2 = 2y = 4$, both nonnegative. Primal feasibility ($x = 1 \geq 1$, $y = 2 \geq 2$), dual feasibility ($\mu_i \geq 0$), and complementary slackness (both constraints tight) all hold, so for this convex problem $(1, 2)$ is the global minimum. Each multiplier is the shadow price of its constraint: tightening $x \geq 1$ to $x \geq 1+\epsilon$ would raise the optimum at rate $\mu_1 = 2$.

## Question 30: A constrained minimum with an active constraint (10 points)

**Problem.** Minimize $x^2 + y^2$ subject to $x + y \geq 4$. Identify whether the constraint is active and give the optimal multiplier.

**Solution.** The unconstrained minimum is $(0, 0)$, but $0 + 0 = 0 < 4$ violates $x + y \geq 4$, so the constraint binds and the optimum lies on the line $x + y = 4$. Minimizing $x^2 + y^2$ on that line (the closest point to the origin) gives $x = y = 2$ by symmetry, with value $f(2, 2) = 8$. The KKT stationarity $\nabla f = \mu\nabla(x + y)$ reads $(2x, 2y) = \mu(1, 1)$, so $\mu = 2x = 4 > 0$. The positive multiplier and the tight constraint satisfy complementary slackness, certifying the minimum for this convex problem. The multiplier $\mu = 4$ is the shadow price: tightening the requirement to $x + y \geq 5$ would raise the optimal cost at rate 4 per unit.

## Question 31: An inactive constraint (10 points)

**Problem.** Minimize $(x - 2)^2 + (y - 1)^2$ subject to $x + y \leq 8$.

**Solution.** Check the unconstrained minimum first: it is $(2, 1)$, where the objective is 0. Test feasibility: $2 + 1 = 3 \leq 8$, satisfied with room to spare, so the constraint is *inactive*. By KKT complementary slackness an inactive constraint forces its multiplier to zero, and the problem reduces to the unconstrained one. The minimum is $(2, 1)$ with value 0. Always test whether the unconstrained optimum is already feasible; only a binding constraint bends the solution (Chapter 9).

### Question 32: A logistic-regression gradient (10 points)

**Problem.** A logistic model has weights $\boldsymbol{\theta} = \mathbf{0}$. For the single example $\boldsymbol{x} = (2, 1)$ with label $y = 0$, compute the gradient of the cross-entropy loss.

**Solution.** The predicted probability is $p = \sigma(\boldsymbol{\theta}^\top \boldsymbol{x}) = \sigma(0) = 0.5$. The one-example cross-entropy gradient is $(p - y)\boldsymbol{x}$:

$$\nabla \ell = (p - y)\boldsymbol{x} = (0.5 - 0)(2, 1) = (1,\ 0.5).$$

A descent step $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla \ell$ moves the weights along $(-1, -0.5)$, lowering $\boldsymbol{\theta}^\top \boldsymbol{x}$ and so lowering the predicted probability on this negative example, exactly as desired. The "prediction minus target" form is identical to linear regression's residual.

## 18.5 Supervised and unsupervised learning

### Question 33: A support vector machine (20 points)

**Problem.** Two points: $\boldsymbol{x}_1 = (0, 0)$ with $y_1 = -1$ and $\boldsymbol{x}_2 = (2, 2)$ with $y_2 = +1$. (a) Find the maximum-margin weights $\boldsymbol{w}$ and bias $b$. (b) Give the margin width. (c) Write the decision function and classify $\boldsymbol{x} = (2, 0)$.

**Solution.** *(a)* By symmetry the boundary bisects the segment and $\boldsymbol{w}$ is parallel to $\boldsymbol{x}_2 - \boldsymbol{x}_1 = (2, 2)$, so $\boldsymbol{w} = c(1, 1)$. The support-vector conditions $\boldsymbol{w}^\top \boldsymbol{x}_i + b = y_i$ read $b = -1$ (from $\boldsymbol{x}_1 = (0, 0)$) and $c(2 + 2) + b = 1$, i.e. $4c - 1 = 1$, so $c = \frac{1}{2}$. Thus $\boldsymbol{w} = (\frac{1}{2}, \frac{1}{2})$, $b = -1$.

*(b)* The margin width is $\dfrac{2}{\|\boldsymbol{w}\|} = \dfrac{2}{\sqrt{1/4 + 1/4}} = \dfrac{2}{1/\sqrt{2}} = 2\sqrt{2} \approx 2.83$, the distance between the two points (which are both support vectors).

*(c)* The decision function is $f(\boldsymbol{x}) = \frac{1}{2}x_1 + \frac{1}{2}x_2 - 1$. At $\boldsymbol{x} = (2, 0)$, $f = \frac{1}{2}(2) + \frac{1}{2}(0) - 1 = 0$, exactly on the boundary, so the point is on the decision line itself (the perpendicular bisector $x_1 + x_2 = 2$), an ambiguous case sitting at margin value 0.

### Question 34: Hinge loss (5 points)

**Problem.** A trained SVM has $\boldsymbol{w} = (1, 0)$, $b = -1$, so $f(\boldsymbol{x}) = x_1 - 1$. Compute the hinge loss $\max(0, 1 - y\, f(\boldsymbol{x}))$ at the point $\boldsymbol{x} = (0.5, 0)$ with label $y = +1$.

**Solution.** The score is $f = 0.5 - 1 = -0.5$, so the margin is $y\, f = -0.5$ and the hinge loss is $\max(0, 1 - (-0.5)) = 1.5$. The point is misclassified ($f < 0$ for a $+1$ label), so it pays a large loss and is a support vector. Only points with margin below 1 contribute, which is the source of the SVM's sparsity.

### Question 35: The kernel trick (10 points)

**Problem.** For the kernel $k(\boldsymbol{x}, \boldsymbol{z}) = (1 + \boldsymbol{x}^\top \boldsymbol{z})^2$ in two dimensions, (a) evaluate $k$ for $\boldsymbol{x} = (1, 2)$, $\boldsymbol{z} = (1, 0)$, and (b) explain what feature space it corresponds to.

**Solution.** *(a)* The inner product is $\boldsymbol{x}^\top \boldsymbol{z} = 1(1) + 2(0) = 1$, so $k = (1 + 1)^2 = 4$.

*(b)* Expanding $(1 + \boldsymbol{x}^\top \boldsymbol{z})^2 = 1 + 2\boldsymbol{x}^\top \boldsymbol{z} + (\boldsymbol{x}^\top \boldsymbol{z})^2$ shows the kernel is an inner product $\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{z})$ for the feature map containing a constant, the linear terms, and all quadratic terms:

$$\boldsymbol{\phi}(\boldsymbol{x}) = (1,\ \sqrt{2}\, x_1,\ \sqrt{2}\, x_2,\ x_1^2,\ \sqrt{2}\, x_1 x_2,\ x_2^2).$$

So the degree-2 polynomial kernel computes a dot product in this six-dimensional space using only the original two-dimensional inner product. The SVM thereby fits a quadratic boundary (conics: ellipses, parabolas, hyperbolas) at the cost of a linear one, never forming $\boldsymbol{\phi}$ explicitly.

## Question 36: A polynomial kernel (10 points)

**Problem.** For the kernel $k(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^\top \boldsymbol{z})^2$ in two dimensions, evaluate $k$ for $\boldsymbol{x} = (2, 1)$, $\boldsymbol{z} = (1, 1)$, and name the feature space it corresponds to.

**Solution.** The inner product is $\boldsymbol{x}^\top \boldsymbol{z} = 2(1) + 1(1) = 3$, so $k = 3^2 = 9$. Expanding $(\boldsymbol{x}^\top \boldsymbol{z})^2$ shows $k = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{z})$ for the quadratic feature map $\boldsymbol{\phi}(\boldsymbol{x}) = (x_1^2,\ \sqrt{2}\, x_1 x_2,\ x_2^2)$. The kernel computes a dot product in that three-dimensional space using only the original two-dimensional inner product, so the SVM fits a quadratic boundary at the cost of a linear one, never forming $\boldsymbol{\phi}$.

## Question 37: A perceptron update (10 points)

**Problem.** A perceptron starts at $\boldsymbol{w} = \boldsymbol{0}$, $b = 0$. The point $\boldsymbol{x} = (1, 2)$ has label $y = +1$. Perform one update and confirm the point is then correctly classified.

**Solution.** The score $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$ is not positive, so the positive point is misclassified and triggers an update. The rule adds the signed point: $\boldsymbol{w} \leftarrow \boldsymbol{w} + y\boldsymbol{x} = (1, 2)$ and $b \leftarrow b + y = 1$. The new score is $(1)(1) + (2)(2) + 1 = 6 > 0$, now correct. Each correction tilts the boundary toward the offending point, and on separable data only finitely many corrections occur.

## Question 38: The softmax (5 points)

**Problem.** Compute the softmax of the logits $\boldsymbol{z} = (2, 0, 1)$.

**Solution.** Exponentiate and normalize: $e^2 = 7.389$, $e^0 = 1$, $e^1 = 2.718$, summing to 11.107, so

$$\boldsymbol{p} = \left( \tfrac{7.389}{11.107}, \tfrac{1}{11.107}, \tfrac{2.718}{11.107} \right) \approx (0.665,\ 0.090,\ 0.245).$$

The probabilities are positive and sum to 1, and the largest logit gets the largest share. Softmax is the multiclass generalization of the sigmoid, sharpening differences through the exponential.

## Question 39: One round of $k$-means (10 points)

**Problem.** Cluster $\{2, 3, 12, 14\}$ with $k = 2$, starting from centroids $\mu_1 = 0$, $\mu_2 = 10$. Perform one assignment and update, and state the within-cluster sum of squares.

**Solution.** *Assignment.* Distances to $\mu_1 = 0$ are $2, 3, 12, 14$; to $\mu_2 = 10$ they are $8, 7, 2, 4$. So $2, 3$ join cluster 1 and $12, 14$ join cluster 2. *Update.* $\mu_1 = \frac{2+3}{2} = 2.5$ and $\mu_2 = \frac{12+14}{2} = 13$. A reassignment would not change the clusters, so this is the converged solution. The objective is

$$J = (2 - 2.5)^2 + (3 - 2.5)^2 + (12 - 13)^2 + (14 - 13)^2 = 0.25 + 0.25 + 1 + 1 = 2.5.$$

The algorithm recovered the two obvious groups in one round, and $J = 2.5$ measures their total tightness.

## Question 40: $k$-means on a second dataset

**Problem.** Cluster $\{2, 3, 11, 12\}$ with $k = 2$ from centroids $\mu_1 = 0$, $\mu_2 = 10$. Perform one assignment and update, and give the objective.

**Solution.** *Assignment.* Distances to 0 are $2, 3, 11, 12$; to 10 they are $8, 7, 1, 2$. So $2, 3$ join cluster 1 and $11, 12$ join cluster 2. *Update.* $\mu_1 = \frac{2+3}{2} = 2.5$, $\mu_2 = \frac{11+12}{2} = 11.5$. A reassignment would not change the clusters, so this is converged. The objective is $(2 - 2.5)^2 + (3 - 2.5)^2 + (11 - 11.5)^2 + (12 - 11.5)^2 = 4(0.25) = 1$.

## Question 41: A GMM responsibility (10 points)

**Problem.** A mixture has two equally weighted components $\mathcal{N}(0,1)$ and $\mathcal{N}(3,1)$. Compute the responsibility of the first component for the point $x = 1$.

**Solution.** The responsibility is the posterior $\gamma_1 = \dfrac{\frac{1}{2}\mathcal{N}(1;0,1)}{\frac{1}{2}\mathcal{N}(1;0,1) + \frac{1}{2}\mathcal{N}(1;3,1)}$. The densities depend on $x$ through $e^{-(x-\mu)^2/2}$: for $\mu = 0$ the exponent is $-\frac{1}{2}$, for $\mu = 3$ it is $-2$. So

$$\gamma_1 = \frac{e^{-1/2}}{e^{-1/2} + e^{-2}} = \frac{0.6065}{0.6065 + 0.1353} \approx 0.818.$$

The point $x = 1$ is closer to component 1, so it is assigned there with about 82% confidence, a soft assignment. This is the E-step of EM; the M-step then re-estimates the means with these responsibilities as weights.

PART IX

# Appendices

APPENDIX A

# Linear Algebra Reference

**This appendix**

A compact refresher on the linear algebra used throughout the book: vector spaces and the four subspaces, rank, orthogonality and projection, the matrix factorizations, norms, and the families of special matrices. Statements are collected for lookup; the derivations live in Chapters 1 to 4.

## A.1 Vector spaces, span, and basis

A *subspace* of $\mathbb{R}^n$ is a nonempty set closed under addition and scalar multiplication. The *span* of vectors is the set of all their linear combinations, the smallest subspace containing them. A *basis* is a linearly independent spanning set; every basis of a subspace has the same size, its *dimension*. For $A \in \mathbb{R}^{m\times n}$ the four fundamental subspaces and their dimensions are

$$\mathrm{Col}(A) \subseteq \mathbb{R}^m \ (\dim r), \quad \mathrm{Nul}(A) \subseteq \mathbb{R}^n \ (\dim n{-}r), \quad \mathrm{Row}(A) \subseteq \mathbb{R}^n \ (\dim r), \quad \mathrm{Nul}(A^\top) \subseteq \mathbb{R}^m \ (\dim m{-}r),$$

where $r = \mathrm{rank}(A)$. The SVD supplies orthonormal bases for all four (Section 4.5.1). Orthogonality pairs them: $\mathrm{Nul}(A) = \mathrm{Row}(A)^\perp$ and $\mathrm{Nul}(A^\top) = \mathrm{Col}(A)^\perp$.

**Rank–nullity theorem**

For $A \in \mathbb{R}^{m\times n}$, $\mathrm{rank}(A) + \dim \mathrm{Nul}(A) = n$. Columns either contribute an independent output direction (rank) or a relation among inputs (null space); together they account for all $n$ columns.

## A.2 Rank facts

For conformable matrices: $\mathrm{rank}(A) = \mathrm{rank}(A^\top) = \mathrm{rank}(A^\top A) = \mathrm{rank}(AA^\top)$ (Exercises 1.3 and 1.4); $\mathrm{rank}(AB) \le \min\{\mathrm{rank}(A), \mathrm{rank}(B)\}$; $\mathrm{rank}(A) \le \min\{m, n\}$, with *full rank* meaning equality. $A \in \mathbb{R}^{n\times n}$ is invertible iff $\mathrm{rank}(A) = n$ iff $\det A \neq 0$ iff 0 is not an eigenvalue (Table 1.3).

## A.3 Inner products, norms, orthogonality

The Euclidean inner product is $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \boldsymbol{x}^\top \boldsymbol{y}$, with $\|\boldsymbol{x}\|_2 = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$ and angle $\cos\vartheta = \langle \boldsymbol{x}, \boldsymbol{y}\rangle / (\|\boldsymbol{x}\|\|\boldsymbol{y}\|)$. Cauchy–Schwarz: $|\langle \boldsymbol{x}, \boldsymbol{y}\rangle| \le \|\boldsymbol{x}\|\|\boldsymbol{y}\|$ (Theorem 1.3). Vectors are *orthogonal* when $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = 0$, *orthonormal* when additionally of unit length. A square $Q$ with orthonormal columns is *orthogonal*: $Q^\top Q = QQ^\top = I$, $Q^{-1} = Q^\top$, and $Q$ preserves lengths and angles.

| Norm | Vector $\|\boldsymbol{x}\|_p$ | Matrix |
|---|---|---|
| $\ell_1$ | $\sum_i \|x_i\|$ | max absolute column sum |
| $\ell_2$ | $\left(\sum_i x_i^2\right)^{1/2}$ | spectral norm $\sigma_1$ |
| $\ell_\infty$ | $\max_i \|x_i\|$ | max absolute row sum |
| Frobenius | — | $\left(\sum_{ij} A_{ij}^2\right)^{1/2} = \left(\sum_k \sigma_k^2\right)^{1/2}$ |

**Table A.1.** Common vector and matrix norms. The matrix 2-norm is the largest singular value; the Frobenius norm is the Euclidean norm of the flattened matrix.

## A.4 Matrix factorizations

| Factorization | Form | Applies to | Used for |
|---|---|---|---|
| LU | $A = LU$ | square | solving $A\boldsymbol{x} = \boldsymbol{b}$, det |
| Cholesky | $A = LL^\top$ | symmetric PD | fast solves, sampling Gaussians |
| QR | $A = QR$ | full column rank | least squares (Section 2.11.1) |
| Eigen | $A = Q\Lambda Q^\top$ | symmetric | spectral theorem (Theorem 3.16) |
| Eigen | $A = PDP^{-1}$ | diagonalizable | powers, dynamics (Section 3.5) |
| SVD | $A = U\Sigma V^\top$ | *any* matrix | rank, norms, pseudoinverse (Chapter 4) |

**Table A.2.** The standard factorizations. The SVD is the universal one; the others specialize to structured matrices but are cheaper when they apply.

## A.5 Special matrices

- **Symmetric** ($A^\top = A$): real eigenvalues, orthonormal eigenbasis (Theorem 3.16).
- **Orthogonal** ($Q^\top Q = I$): eigenvalues on the unit circle, length-preserving.
- **Positive (semi)definite** ($\boldsymbol{x}^\top A\boldsymbol{x} > 0$, resp. $\geq 0$): symmetric with positive (nonnegative) eigenvalues (Proposition 3.24); admits a Cholesky factor; the shape of every covariance matrix and minimizing Hessian.
- **Idempotent** ($P^2 = P$): a projection; eigenvalues $0, 1$; $\operatorname{tr} P = \operatorname{rank} P$. Symmetric idempotents are orthogonal projections (Definition 2.4).
- **Diagonal / triangular**: eigenvalues are the diagonal entries; determinant is their product.

## A.6 Determinant and trace identities

$\det(AB) = \det A \det B$, $\det(A^\top) = \det A$, $\det(cA) = c^n \det A$, $\det(A^{-1}) = 1/\det A$, $\det A = \prod_i \lambda_i$. $\operatorname{tr}(A+B) = \operatorname{tr} A + \operatorname{tr} B$, $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ (Eq. (1.11)), $\operatorname{tr} A = \sum_i \lambda_i$. For a $2 \times 2$ matrix, $\det\left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right] = ad - bc$ and the inverse is $\frac{1}{ad-bc}\left[\begin{smallmatrix} d & -b \\ -c & a \end{smallmatrix}\right]$ (Eq. (1.10)).

APPENDIX B

# Probability and Distributions Reference

**This appendix**

The probability facts used in Chapters 5 to 7: the rules, the moment identities, the distribution catalogue with means and variances, conjugate pairs, and the concentration inequalities, gathered for lookup.

## B.1 Rules

**Sum rule:** $p(x) = \sum_y p(x, y)$ (or $\int p(x, y)\, dy$). **Product rule:** $p(x, y) = p(x \mid y)p(y)$. **Bayes:** $p(\theta \mid \mathcal{D}) = \dfrac{p(\mathcal{D} \mid \theta)p(\theta)}{p(\mathcal{D})} \propto p(\mathcal{D} \mid \theta)p(\theta)$ (Eq. (5.7)). **Independence:** $p(x, y) = p(x)p(y)$. **Total probability:** $p(x) = \sum_k p(x \mid y_k)p(y_k)$.

## B.2 Expectation, variance, covariance

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y] \text{ (always)}, \qquad \mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2,$$
$$\mathrm{Var}(aX + b) = a^2 \mathrm{Var}(X), \qquad \mathrm{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y],$$
$$\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\,\mathrm{Cov}(X, Y), \qquad \mathrm{Cov}(\boldsymbol{x}) = \mathbb{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top] \succeq 0.$$

For independent $X, Y$: $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$ and $\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y)$. Affine maps of random vectors: $\mathbb{E}[A\boldsymbol{x} + \boldsymbol{b}] = A\boldsymbol{\mu} + \boldsymbol{b}$ and $\mathrm{Cov}(A\boldsymbol{x} + \boldsymbol{b}) = A\boldsymbol{\Sigma}A^\top$ (Eq. (6.12)). Zero covariance is implied by independence but does not imply it, except for jointly Gaussian variables (Proposition 6.15).

## B.3 Distribution catalogue

## B.4 Estimation

**MLE:** $\hat{\theta}_{\mathrm{MLE}} = \arg\max_\theta p(\mathcal{D} \mid \theta)$, usually via the log-likelihood. **MAP:** $\hat{\theta}_{\mathrm{MAP}} = \arg\max_\theta \big[\log p(\mathcal{D} \mid \theta) + \log p(\theta)\big]$, the MLE plus a prior penalty (regularization, Remark 5.12). Bernoulli MLE $\hat{\mu} = m/n$; multinomial MLE $\hat{\mu}_k = m_k/N$; Gaussian MLE the sample mean and covariance (Eq. (6.9)). The Beta–Bernoulli and Dirichlet–multinomial posteriors add observed counts to the prior pseudo-counts.

## B.5 Key inequalities and limits

- **Jensen:** for convex $\varphi$, $\varphi(\mathbb{E}[X]) \le \mathbb{E}[\varphi(X)]$ (reversed for concave); underlies the EM lower bound.
- **Markov:** $\mathbb{P}(X \ge a) \le \mathbb{E}[X]/a$ for $X \ge 0$.
- **Chebyshev:** $\mathbb{P}(|X - \mu| \ge a) \le \sigma^2/a^2$.
- **Chernoff / Hoeffding:** exponential tails for sums of independent (bounded) variables (Table 7.1).
- **Law of large numbers:** the sample mean of iid variables converges to the true mean.

| **Distribution** | **pmf / pdf** | **Mean** | **Variance** | **Conjugate** |
|---|---|---|---|---|
| Bernoulli($\mu$) | $\mu^x(1-\mu)^{1-x}$ | $\mu$ | $\mu(1-\mu)$ | Beta |
| Binomial($n, \mu$) | $\binom{n}{x}\mu^x(1-\mu)^{n-x}$ | $n\mu$ | $n\mu(1-\mu)$ | Beta |
| Categorical($\boldsymbol{\mu}$) | $\prod_k \mu_k^{x_k}$ | $\boldsymbol{\mu}$ | — | Dirichlet |
| Multinomial($N, \boldsymbol{\mu}$) | $\binom{N}{\boldsymbol{m}}\prod_k \mu_k^{m_k}$ | $N\boldsymbol{\mu}$ | $N\mu_k(1-\mu_k)$ | Dirichlet |
| Beta($a, b$) | $\frac{\mu^{a-1}(1-\mu)^{b-1}}{B(a,b)}$ | $\frac{a}{a+b}$ | $\frac{ab}{(a+b)^2(a+b+1)}$ | — |
| Dirichlet($\boldsymbol{\alpha}$) | $\frac{1}{B(\boldsymbol{\alpha})}\prod_k \mu_k^{\alpha_k-1}$ | $\frac{\boldsymbol{\alpha}}{\alpha_0}$ | — | — |
| $\mathcal{N}(\mu, \sigma^2)$ | $\frac{1}{\sigma\sqrt{2\pi}}e^{-(x-\mu)^2/2\sigma^2}$ | $\mu$ | $\sigma^2$ | Gaussian |
| $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ | $\frac{e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}}{(2\pi)^{d/2}\lvert\boldsymbol{\Sigma}\rvert^{1/2}}$ | $\boldsymbol{\mu}$ | $\boldsymbol{\Sigma}$ | Gaussian |

**Table B.1.** The distributions of the course, with first two moments and conjugate priors. $B(\cdot)$ is the (multivariate) Beta function; $\alpha_0 = \sum_k \alpha_k$. See Tables 5.2 and 6.2.

- **Central limit theorem:** the normalized sum of many independent finite-variance variables converges to a Gaussian, the reason the Gaussian is ubiquitous (Chapter 6).

APPENDIX C

# Matrix Calculus Identities

**This appendix**

The gradient, Jacobian, and trace identities used in Chapters 2, 8 and 10. We use *denominator layout*: the derivative of a scalar by a column vector is a column vector of the same shape. Throughout, $\boldsymbol{a}, \boldsymbol{b}$ are constant vectors, $A, B$ constant matrices, and $\boldsymbol{x}$ the variable.

## C.1 Scalar by vector

| **Function** $f(\boldsymbol{x})$ | **Gradient** $\nabla_{\boldsymbol{x}} f$ | **Note** |
|---|---|---|
| $\boldsymbol{a}^\top \boldsymbol{x} = \boldsymbol{x}^\top \boldsymbol{a}$ | $\boldsymbol{a}$ | linear |
| $\boldsymbol{x}^\top \boldsymbol{x} = \|\boldsymbol{x}\|^2$ | $2\boldsymbol{x}$ | |
| $\boldsymbol{x}^\top A \boldsymbol{x}$ | $(A + A^\top)\boldsymbol{x}$ | $= 2A\boldsymbol{x}$ if $A$ symmetric |
| $\|A\boldsymbol{x} - \boldsymbol{b}\|_2^2$ | $2A^\top(A\boldsymbol{x} - \boldsymbol{b})$ | least squares (Eq. (2.4)) |
| $\|A\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|^2$ | $2A^\top(A\boldsymbol{x} - \boldsymbol{b}) + 2\lambda\boldsymbol{x}$ | ridge (Eq. (2.15)) |
| $\log\det X$ (in $X$) | $X^{-\top}$ | for invertible $X$ |

**Table C.1.** Gradients of the scalar functions that appear in the loss derivations.

## C.2 Hessians

$\nabla^2(\boldsymbol{a}^\top \boldsymbol{x}) = 0$; $\nabla^2(\boldsymbol{x}^\top A \boldsymbol{x}) = A + A^\top$ ($= 2A$ symmetric); $\nabla^2 \tfrac{1}{2}\|A\boldsymbol{x} - \boldsymbol{b}\|^2 = A^\top A$; $\nabla^2\big[\tfrac{1}{2}\|A\boldsymbol{x} - \boldsymbol{b}\|^2 + \tfrac{\lambda}{2}\|\boldsymbol{x}\|^2\big] = A^\top A + \lambda I$. A function is convex exactly where its Hessian is positive semidefinite (Theorem 8.9).

## C.3 Vector by vector (Jacobians) and the chain rule

For $F : \mathbb{R}^n \to \mathbb{R}^m$, the Jacobian $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ has $\boldsymbol{J}_{ij} = \partial F_i / \partial x_j$. Linear maps: $\partial(A\boldsymbol{x})/\partial \boldsymbol{x} = A$. Chain rule for $\boldsymbol{y} = f(g(\boldsymbol{x}))$:

$$\boldsymbol{J}_{f \circ g}(\boldsymbol{x}) = \boldsymbol{J}_f\big(g(\boldsymbol{x})\big)\, \boldsymbol{J}_g(\boldsymbol{x}),$$

the product of Jacobians, applied right-to-left in backpropagation (Section 8.3). Elementwise activation $\boldsymbol{a} = \sigma(\boldsymbol{z})$ has diagonal Jacobian $\operatorname{diag}(\sigma'(\boldsymbol{z}))$, so the backprop error is $\boldsymbol{\delta} = \partial L/\partial \boldsymbol{z} = (\partial L/\partial \boldsymbol{a}) \odot \sigma'(\boldsymbol{z})$.

## C.4 Trace and determinant derivatives

The trace linearizes matrix expressions so they can be differentiated. Useful identities (derivatives with respect to $X$):

$$\nabla_X \operatorname{tr}(AX) = A^\top, \quad \nabla_X \operatorname{tr}(X^\top A) = A, \quad \nabla_X \operatorname{tr}(X^\top A X) = (A + A^\top)X, \quad \nabla_X \log\det X = X^{-\top}.$$

Cyclic property $\operatorname{tr}(ABC) = \operatorname{tr}(BCA) = \operatorname{tr}(CAB)$ is what lets the squared norm be written $\|A\boldsymbol{x} - \boldsymbol{b}\|^2 = \operatorname{tr}\big((A\boldsymbol{x} - \boldsymbol{b})(A\boldsymbol{x} - \boldsymbol{b})^\top\big)$ and then differentiated termwise (Section 8.7).

## C.5 Gradients of the common machine-learning losses

| **Model** | **Loss** | **Gradient in $\boldsymbol{\theta}$** |
|---|---|---|
| Linear regression | $\frac{1}{2m}\|X\boldsymbol{\theta} - \boldsymbol{y}\|^2$ | $\frac{1}{m}X^\top(X\boldsymbol{\theta} - \boldsymbol{y})$ |
| Ridge regression | $+\,\frac{\lambda}{2}\|\boldsymbol{\theta}\|^2$ | $+\,\lambda\boldsymbol{\theta}$ |
| Logistic regression | cross-entropy | $\frac{1}{m}X^\top(\boldsymbol{p} - \boldsymbol{y})$, $\boldsymbol{p} = \sigma(X\boldsymbol{\theta})$ |
| Soft-margin SVM | hinge $+\,\frac{1}{2C}\|\boldsymbol{w}\|^2$ | subgradient on margin violators |

**Table C.2.** The gradients that drive training. Linear and logistic regression share the form $\frac{1}{m}X^\top(\text{prediction} - \text{target})$, the residual pushed back through the design matrix.

APPENDIX D

# Software Setup and the Coding Assignments

**This appendix**

The computing environment for the course and a topic-by-topic guide to the coding work: which library routines implement each piece of mathematics, and how the chapters map onto code. The aim is to connect the equations to the few lines that realize them.

## D.1 Environment

The assignments use Python with the scientific stack: `numpy` for arrays and linear algebra [10], `scipy` for optimization and statistics, `matplotlib` for plotting, and `scikit-learn` for reference implementations of the models [17]. A managed environment (for example through `conda`) keeps versions consistent, and the notebooks run in Jupyter. Two habits make results reproducible: fix the random seed at the top of each notebook, and prefer vectorized array operations over Python loops, both for speed and because the vectorized form usually mirrors the mathematics more closely.

## D.2 From mathematics to library calls

| Operation | Routine | Chapter |
|---|---|---|
| matrix product, transpose | `A @ B`, `A.T` | Chapter 1 |
| solve $A\boldsymbol{x} = \boldsymbol{b}$ | `np.linalg.solve` | Chapter 1 |
| least squares | `np.linalg.lstsq` | Chapter 2 |
| pseudoinverse | `np.linalg.pinv` | Chapters 2 and 4 |
| eigendecomposition (symmetric) | `np.linalg.eigh` | Chapter 3 |
| SVD | `np.linalg.svd` | Chapter 4 |
| multivariate normal | `scipy.stats.multivariate_normal` | Chapter 6 |
| gradient-based fit | `scipy.optimize.minimize` | Chapter 8 |
| $k$-means, GMM, PCA, SVM | `sklearn` estimators | Chapters 10 and 11 |

**Table D.1.** The mathematics of each chapter and the routine that realizes it. The library calls are for checking work and scaling up; the assignments ask you to implement the core of each from the equations first.

## D.3 The coding assignments

Each assignment implements one chapter's central algorithm from scratch, then validates against the library reference and a held-out set.

**Vectorization and the linear model**
Load a small image dataset, flatten each image to a vector (Fig. 1.2), assemble the design matrix, and fit a linear model. The lesson is the data-to-matrix pipeline and why vectorized code is both faster and clearer.

**Least squares and the pseudoinverse**
Solve a regression both by the normal equations and by the $QR$ route (Section 2.11.1), compare their numerical behavior on an ill-conditioned design, and add ridge regularization to see the conditioning improve.

**SVD and low-rank approximation**
Compute the SVD of an image, reconstruct it at increasing ranks, and plot the error against the discarded singular values to witness Eckart–Young (Theorem 4.6) and the scree elbow.

**Gradient descent and logistic regression**
Implement batch and stochastic gradient descent for the convex cross-entropy loss, tune the learning rate, and watch the decision boundary form. Compare convergence with and without momentum.

**Clustering and dimensionality reduction**
Implement Lloyd's algorithm and EM for a Gaussian mixture, compare hard and soft assignments (Fig. 11.2), then run PCA via the SVD and visualize the data in its top two components.

**The support vector machine**
Set up the dual quadratic program, solve it (a small SMO loop or a QP solver), recover the support vectors through complementary slackness, and swap in an RBF kernel for a nonlinear boundary.

## D.4 Debugging the mathematics

Three checks catch most errors. *Shapes:* every product and gradient should have the dimensions the equations predict (Exercise 8.5). *Gradients:* verify an analytic gradient against a finite-difference estimate $\big(f(\boldsymbol{x}+\varepsilon\boldsymbol{e}_i) - f(\boldsymbol{x}-\varepsilon\boldsymbol{e}_i)\big)/2\varepsilon$ before trusting it. *Sanity values:* test on a tiny hand-solved instance, like the worked examples in the chapters, where the right answer is known in closed form.

APPENDIX E

# Practice Final, with Solutions

**This appendix**

A full practice final, assembled in the format of the course examinations, with complete solutions. The seven questions sweep the whole book: eigenvalues and spectra, the SVD, spectral decomposition and the pseudoinverse, Bayesian reasoning, logistic regression, support vector machines, and convexity with the KKT conditions. Each solution cross-references the chapter that develops it. Many of the per-chapter exercises are drawn from the same examinations; this appendix shows how the pieces fit into one paper.

## Question 1: Eigenvalues, inverses, and spectra (25 pts)

**(a)** Do $AB$ and $BA$ have the same eigenvalues? **(b)** Must a real invertible $n \times n$ matrix have a real eigenvalue? Answer for $n = 3$ and $n = 4$. **(c)** If $\lambda \neq 0$ is an eigenvalue of invertible $A$, is $1/\lambda$ an eigenvalue of $A^{-1}$? **(d)** Do $A$ and $A^{-1}$ share eigenvectors? **(e)** If $A^2 = I$ and $-1$ is not an eigenvalue of $A$, must $A = I$?

*Solution.* **(a)** Yes. If $\lambda \neq 0$ and $BA\boldsymbol{v} = \lambda \boldsymbol{v}$, then $AB(A\boldsymbol{v}) = \lambda(A\boldsymbol{v})$ with $A\boldsymbol{v} \neq \mathbf{0}$, so $\lambda$ is an eigenvalue of $AB$; the argument is symmetric, and $\lambda = 0$ occurs for one iff for the other since $\det(AB) = \det(BA)$. (For square $A, B$ the characteristic polynomials are in fact equal.) **(b)** The characteristic polynomial is real of degree $n$. For odd $n = 3$ it has a real root, so yes. For even $n = 4$ it need not: a block of two $2 \times 2$ rotations is real, invertible, with only complex eigenvalues (Exercise 3.5). **(c)** Yes: from $A\boldsymbol{v} = \lambda \boldsymbol{v}$, apply $A^{-1}$ to get $A^{-1}\boldsymbol{v} = \frac{1}{\lambda}\boldsymbol{v}$. **(d)** Yes, the same $\boldsymbol{v}$ works for both, with reciprocal eigenvalues. **(e)** Yes. $A^2 = I$ forces every eigenvalue to satisfy $\lambda^2 = 1$, so $\lambda \in \{+1, -1\}$; excluding $-1$ leaves only $\lambda = 1$. Since $A^2 = I$ means $A$ is diagonalizable (it satisfies the squarefree polynomial $t^2 - 1$), $A = QIQ^{-1} = I$. See Chapter 3. □

## Question 2: Singular value decomposition (20 pts)

Find an SVD of $A = \left[\begin{smallmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, then use it to find the $\boldsymbol{x}$ that best solves $A\boldsymbol{x} = \boldsymbol{b}$ for a given $\boldsymbol{b}$.

*Solution.* **Right singular vectors.** $A^\top A = \left[\begin{smallmatrix} 2 & -1 \\ -1 & 2 \end{smallmatrix}\right]$ has eigenvalues $\lambda_1 = 3$, $\lambda_2 = 1$ with unit eigenvectors $\boldsymbol{v}_1 = \frac{1}{\sqrt{2}}(1, -1)^\top$, $\boldsymbol{v}_2 = \frac{1}{\sqrt{2}}(1, 1)^\top$. So the singular values are $\sigma_1 = \sqrt{3}$, $\sigma_2 = 1$. **Left singular vectors** $\boldsymbol{u}_i = A\boldsymbol{v}_i/\sigma_i$:

$$\boldsymbol{u}_1 = \frac{A\boldsymbol{v}_1}{\sqrt{3}} = \frac{1}{\sqrt{6}}\begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{u}_2 = \frac{A\boldsymbol{v}_2}{1} = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix},$$

both unit and orthogonal. A third column $\boldsymbol{u}_3$, any unit vector orthogonal to both (spanning $\mathrm{Nul}(A^\top)$), completes $U \in \mathbb{R}^{3\times 3}$, with $\Sigma = \left[\begin{smallmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$ and $V = [\boldsymbol{v}_1\ \boldsymbol{v}_2]$. **Best solution.** The least-squares solution is $\boldsymbol{x}^\star = A^\dagger \boldsymbol{b}$ with $A^\dagger = V\Sigma^\dagger U^\top$, $\Sigma^\dagger = \left[\begin{smallmatrix} 1/\sqrt{3} & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right]$; concretely $\boldsymbol{x}^\star = \sum_{i=1}^{2} \frac{\boldsymbol{u}_i^\top \boldsymbol{b}}{\sigma_i}\boldsymbol{v}_i$, the projection of $\boldsymbol{b}$ onto $\mathrm{Col}(A)$ expressed in the right singular basis (Chapter 4, Section 4.8). □

## Question 3: Spectral decomposition and the pseudoinverse (20 pts)

**(a)** Diagonalize $A = \left[\begin{smallmatrix} 0 & 1 \\ 3 & 2 \end{smallmatrix}\right]$ over $\mathbb{R}$. **(b)** Find the Moore–Penrose pseudoinverse of $A = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$.

*Solution.* **(a)** Characteristic polynomial $\det(A - \lambda I) = \lambda^2 - 2\lambda - 3 = (\lambda - 3)(\lambda + 1)$, so $\lambda_1 = 3$, $\lambda_2 = -1$. For $\lambda = 3$, $(A - 3I)\boldsymbol{v} = \mathbf{0}$ gives $\boldsymbol{v}_1 = (1, 3)^\top$; for $\lambda = -1$, $\boldsymbol{v}_2 = (1, -1)^\top$. Hence

$$A = PDP^{-1}, \quad P = \begin{bmatrix} 1 & 1 \\ 3 & -1 \end{bmatrix}, \quad D = \begin{bmatrix} 3 & 0 \\ 0 & -1 \end{bmatrix}, \quad P^{-1} = \frac{1}{-4}\begin{bmatrix} -1 & -1 \\ -3 & 1 \end{bmatrix}.$$

$A$ is not symmetric, so $P$ is not orthogonal ($\boldsymbol{v}_1^\top \boldsymbol{v}_2 = 1 - 3 = -2 \neq 0$); the eigenvectors are independent but not perpendicular (Section 3.5). **(b)** $A = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$ has rank 1 with SVD singular value $\sigma_1 = 1$, $\boldsymbol{u}_1 = (1, 0)^\top$, $\boldsymbol{v}_1 = (0, 1)^\top$. By Eq. (4.7), $A^\dagger = \frac{1}{\sigma_1}\boldsymbol{v}_1\boldsymbol{u}_1^\top = (0, 1)^\top(1, 0) = \left[\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right]$. One checks $AA^\dagger A = A$. (This is the defective matrix of Example 3.12: it has no inverse, but the pseudoinverse always exists.) □

## Question 4: Bayes' theorem and Monty Hall (20 pts)

**(a)** Three doors, one car. You pick door 1; the host opens door 2 on a goat. Should you switch? **(b)** Repeat with four doors (one car, three goats), host opens door 2.

*Solution.* **(a)** With uniform prior $\mathbb{P}(H_i) = \frac{1}{3}$ and host likelihoods $\mathbb{P}(E \mid H_1) = \frac{1}{2}$, $\mathbb{P}(E \mid H_2) = 0$, $\mathbb{P}(E \mid H_3) = 1$, the evidence is $\mathbb{P}(E) = \frac{1}{2}$ and Bayes gives $\mathbb{P}(H_1 \mid E) = \frac{1}{3}$, $\mathbb{P}(H_3 \mid E) = \frac{2}{3}$. **Switch** (Section 5.8.1). **(b)** Now $\mathbb{P}(H_i) = \frac{1}{4}$; likelihoods of opening door 2 are $\frac{1}{3}, 0, \frac{1}{2}, \frac{1}{2}$, so $\mathbb{P}(E) = \frac{1}{3}$ and $\mathbb{P}(H_1 \mid E) = \frac{1}{4}$, $\mathbb{P}(H_3 \mid E) = \mathbb{P}(H_4 \mid E) = \frac{3}{8}$. Either unopened door beats staying, so **switch** (Exercise 5.4). The host's informed choice redistributes the opened door's probability onto the doors you neither picked nor saw opened. □

## Question 5: Logistic regression (20 pts)

**(a)** With $X = \left[\begin{smallmatrix} 1 & 2 \\ 1 & -1 \\ 1 & 0 \end{smallmatrix}\right]$, $\boldsymbol{y} = (1, 0, 1)$, $\boldsymbol{\theta}_0 = \mathbf{0}$, take one full-batch gradient-ascent step on the log-likelihood ($\alpha = 0.01$). **(b)** Why is cross-entropy preferred over squared error for this model?

*Solution.* **(a)** At $\boldsymbol{\theta}_0 = \mathbf{0}$ every score is 0 and $p_i = \sigma(0) = \frac{1}{2}$, so $\boldsymbol{y} - \boldsymbol{p} = (\frac{1}{2}, -\frac{1}{2}, \frac{1}{2})$ and the ascent gradient is $X^\top(\boldsymbol{y} - \boldsymbol{p}) = (\frac{1}{2}, \frac{3}{2})^\top$. One step: $\boldsymbol{\theta}_1 = \mathbf{0} + 0.01(0.5, 1.5) = (0.005, 0.015)$ (Exercise 8.2). **(b)** Cross-entropy is convex in $\boldsymbol{\theta}$ (Hessian $\frac{1}{m}X^\top \operatorname{diag}(p_i(1 - p_i))X \succeq 0$), so gradient descent reaches the global optimum; the squared error $\frac{1}{2m}\sum(p_i - y_i)^2$ on the sigmoid output is nonconvex and can trap a solver in local minima (Section 8.10). Cross-entropy is also the maximum-likelihood loss for Bernoulli labels. □

## Question 6: Support vector machines (35 pts)

**(a)** Define the margin and the support vectors. **(b)** For $\boldsymbol{x}_1 = (0, 0), y_1 = -1$ and $\boldsymbol{x}_2 = (2, 0), y_2 = +1$, solve the hard-margin SVM (find $\boldsymbol{\alpha}, \boldsymbol{w}, b$, the margin, and the boundary). **(c)** What changes in the dual for the soft margin?

*Solution.* **(a)** The margin is the width $2/\|\boldsymbol{w}\|$ of the empty corridor between the gutters $\boldsymbol{w}^\top\boldsymbol{x} + b = \pm 1$; the support vectors are the training points lying on the gutters, the only ones with $\alpha_i > 0$ (Section 10.4). **(b)** The constraint $\sum_i \alpha_i y_i = 0$ gives $\alpha_1 = \alpha_2 = \alpha$; both points are support vectors. Stationarity gives $\boldsymbol{w} = \alpha(2, 0) = (2\alpha, 0)$, and the unit-margin condition forces $\boldsymbol{w} = (1, 0)$, $b = -1$, so $\alpha = \frac{1}{2}$. The margin is $2/\|\boldsymbol{w}\| = 2$ and the boundary is $x_1 = 1$ (Exercise 10.1). **(c)** The soft margin adds slacks $\xi_i \geq 0$ penalized by $C$; the dual is unchanged except the multipliers gain the cap $0 \leq \alpha_i \leq C$ (Eq. (10.7)), which sorts points into non-support ($\alpha_i = 0$), margin ($0 < \alpha_i < C$), and violating ($\alpha_i = C$). □

## Question 7: Convexity and the KKT conditions (20 pts)

**(a)** Define a convex function. **(b)** State the KKT conditions for $\min f(\boldsymbol{w})$ s.t. $g_i(\boldsymbol{w}) \le 0$. **(c)** Prove that for convex differentiable $f$, $\nabla f(\boldsymbol{w}^\star) = \mathbf{0}$ implies $\boldsymbol{w}^\star$ is a global minimizer.

*Solution.* **(a)** $f$ is convex if $f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \le \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y})$ for all $\boldsymbol{x}, \boldsymbol{y}$ and $\lambda \in [0,1]$: every chord lies on or above the graph (Definition 8.8). **(b)** With $\mathcal{L} = f + \sum_i \alpha_i g_i$: stationarity $\nabla_{\boldsymbol{w}} \mathcal{L} = \mathbf{0}$; primal feasibility $g_i(\boldsymbol{w}^\star) \le 0$; dual feasibility $\alpha_i \ge 0$; complementary slackness $\alpha_i g_i(\boldsymbol{w}^\star) = 0$ (Table 9.1). For convex problems they are sufficient as well as necessary. **(c)** By the first-order convexity inequality, $f(\boldsymbol{y}) \ge f(\boldsymbol{w}^\star) + \nabla f(\boldsymbol{w}^\star)^\top (\boldsymbol{y} - \boldsymbol{w}^\star)$ for all $\boldsymbol{y}$. With $\nabla f(\boldsymbol{w}^\star) = \mathbf{0}$ the last term vanishes, so $f(\boldsymbol{y}) \ge f(\boldsymbol{w}^\star)$ everywhere: $\boldsymbol{w}^\star$ is global (Exercise 8.3). □

## Practice Midterm: the linear-algebra core

Four shorter problems covering the matrix-methods half of the course, each worked in full.

### M1. Eigenvalues and diagonalization (15 pts)

Diagonalize $A = \left[\begin{smallmatrix} 5 & 4 \\ 4 & 5 \end{smallmatrix}\right]$ and use it to compute $A^2$.

*Solution.* $A$ is symmetric, so the spectral theorem applies. The characteristic polynomial is $(5-\lambda)^2 - 16 = 0$, i.e. $5 - \lambda = \pm 4$, giving $\lambda_1 = 9$, $\lambda_2 = 1$. For $\lambda_1 = 9$, $(A - 9I)\boldsymbol{v} = \left[\begin{smallmatrix} -4 & 4 \\ 4 & -4 \end{smallmatrix}\right]\boldsymbol{v} = \mathbf{0}$ gives $\boldsymbol{v}_1 = \frac{1}{\sqrt{2}}(1,1)$; for $\lambda_2 = 1$, $\boldsymbol{v}_2 = \frac{1}{\sqrt{2}}(1,-1)$, orthonormal as promised. With $Q = [\boldsymbol{v}_1\ \boldsymbol{v}_2]$ and $\Lambda = \operatorname{diag}(9,1)$, $A = Q\Lambda Q^\top$. Then $A^2 = Q\Lambda^2 Q^\top$ with $\Lambda^2 = \operatorname{diag}(81,1)$:

$$A^2 = \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 81 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 41 & 40 \\ 40 & 41 \end{bmatrix},$$

which the direct product $\left[\begin{smallmatrix} 5 & 4 \\ 4 & 5 \end{smallmatrix}\right]^2 = \left[\begin{smallmatrix} 25+16 & 20+20 \\ 20+20 & 16+25 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 41 & 40 \\ 40 & 41 \end{smallmatrix}\right]$ confirms. □

### M2. Singular value decomposition (15 pts)

Find the SVD of $A = \left[\begin{smallmatrix} 2 & 2 \\ -1 & 1 \end{smallmatrix}\right]$.

*Solution.* $A^\top A = \left[\begin{smallmatrix} 5 & 3 \\ 3 & 5 \end{smallmatrix}\right]$ has eigenvalues $5 \pm 3 = \{8, 2\}$, so the singular values are $\sigma_1 = \sqrt{8} = 2\sqrt{2}$, $\sigma_2 = \sqrt{2}$. The right singular vectors are the eigenvectors of $A^\top A$: $\boldsymbol{v}_1 = \frac{1}{\sqrt{2}}(1,1)$, $\boldsymbol{v}_2 = \frac{1}{\sqrt{2}}(1,-1)$. The left singular vectors are $\boldsymbol{u}_i = A\boldsymbol{v}_i / \sigma_i$:

$$\boldsymbol{u}_1 = \frac{1}{2\sqrt{2}} \cdot \tfrac{1}{\sqrt{2}}\begin{bmatrix} 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \boldsymbol{u}_2 = \frac{1}{\sqrt{2}} \cdot \tfrac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}.$$

Hence $A = U\Sigma V^\top$ with $U = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$, $\Sigma = \operatorname{diag}(2\sqrt{2}, \sqrt{2})$, $V = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$. One checks $U\Sigma V^\top = \left[\begin{smallmatrix} 2 & 2 \\ -1 & 1 \end{smallmatrix}\right]$. □

### M3. Least squares (10 pts)

Fit the line $y = \theta_1 + \theta_2 t$ to the points $(0,1), (1,2), (2,0)$ by least squares.

*Solution.* With $A = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ and $\boldsymbol{b} = (1,2,0)^\top$, $A^\top A = \left[\begin{smallmatrix} 3 & 3 \\ 3 & 5 \end{smallmatrix}\right]$ ($\det = 6$) and $A^\top \boldsymbol{b} = (3,2)^\top$. Then

$$\boldsymbol{\theta}^\star = (A^\top A)^{-1} A^\top \boldsymbol{b} = \tfrac{1}{6}\begin{bmatrix} 5 & -3 \\ -3 & 3 \end{bmatrix}\begin{bmatrix} 3 \\ 2 \end{bmatrix} = \tfrac{1}{6}\begin{bmatrix} 9 \\ -3 \end{bmatrix} = \begin{bmatrix} 3/2 \\ -1/2 \end{bmatrix},$$

the line $y = \frac{3}{2} - \frac{1}{2}t$. The residual $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{\theta}^\star = (-\frac{1}{2}, 1, -\frac{1}{2})^\top$ satisfies $A^\top \boldsymbol{r} = \mathbf{0}$, confirming it is orthogonal to the column space. □

### M4. Positive definiteness (10 pts)

Is $A = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ positive definite? Give two arguments.

*Solution.* *Eigenvalues:* the characteristic polynomial $(2-\lambda)^2 - 1$ has roots $\lambda = 3, 1$, both positive, so $A \succ 0$ by Proposition 3.24. *Quadratic form:* completing the square, $\boldsymbol{x}^\top A \boldsymbol{x} = 2x_1^2 + 2x_1x_2 + 2x_2^2 = (x_1+x_2)^2 + x_1^2 + x_2^2$, a sum of squares that is strictly positive unless $x_1 = x_2 = 0$. Either way $A$ is positive definite; as a check its leading minors $2 > 0$ and $\det = 3 > 0$ are both positive (Sylvester's criterion). □

## Practice Midterm B: probability, calculus, and optimization

Four problems covering the analysis half of the course.

### B1. Expectation and variance (10 pts)

A discrete random variable takes values $0, 1, 2, 3$ with probabilities $0.1, 0.4, 0.3, 0.2$. Find $\mathbb{E}[X]$ and $\mathrm{Var}(X)$.

*Solution.* $\mathbb{E}[X] = \sum_x x\,p(x) = 0(0.1) + 1(0.4) + 2(0.3) + 3(0.2) = 0.4 + 0.6 + 0.6 = 1.6$. For the variance use $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$ with $\mathbb{E}[X^2] = 0 + 1(0.4) + 4(0.3) + 9(0.2) = 0.4 + 1.2 + 1.8 = 3.4$, so $\mathrm{Var}(X) = 3.4 - 1.6^2 = 3.4 - 2.56 = 0.84$ (standard deviation $\sqrt{0.84} \approx 0.92$). □

### B2. Gradient, Hessian, convexity (15 pts)

For $f(x, y) = 2x^2 + 3y^2 + 2xy - 4x$, find the gradient and Hessian, show $f$ is convex, and locate its minimum.

*Solution.* $\nabla f = (4x + 2y - 4,\ 6y + 2x)$ and $\nabla^2 f = \left[\begin{smallmatrix} 4 & 2 \\ 2 & 6 \end{smallmatrix}\right]$, constant. Its leading minors are $4 > 0$ and $\det = 24 - 4 = 20 > 0$ (eigenvalues $5 \pm \sqrt{5} \approx 2.76, 7.24$, both positive), so $\nabla^2 f \succ 0$ and $f$ is strictly convex. Setting $\nabla f = \mathbf{0}$: from $6y + 2x = 0$, $x = -3y$; substituting into $4x + 2y = 4$ gives $-12y + 2y = 4$, $y = -0.4$, $x = 1.2$. The unique global minimum is at $(1.2, -0.4)$. □

### B3. MLE and MAP (10 pts)

A coin is flipped 10 times, landing heads 7 times. Give the MLE of the bias $\mu$, and the posterior and MAP estimate under a Beta$(2, 2)$ prior.

*Solution.* The MLE maximizes $\mu^7(1-\mu)^3$; its log-derivative $7/\mu - 3/(1-\mu) = 0$ gives $\hat{\mu}_{\mathrm{MLE}} = 7/10 = 0.7$. With a Beta$(2, 2)$ prior the Beta–Bernoulli update adds the counts: posterior Beta$(2+7,\ 2+3) =$ Beta$(9, 5)$. Its mode (the MAP) is $(9-1)/(9+5-2) = 8/12 \approx 0.667$, and its mean is $9/14 \approx 0.643$. Both sit below the MLE, pulled toward the prior mean $\frac{1}{2}$ by the two pseudo-heads and two pseudo-tails. □

### B4. Gradient descent (15 pts)

Minimize $f(x) = \frac{1}{2}(x-3)^2$ by gradient descent from $x_0 = 0$ with step $\alpha = 0.5$. Write the iteration in closed form and give the first few iterates.

*Solution.* Here $f'(x) = x - 3$, so the update is $x_{t+1} = x_t - 0.5(x_t - 3) = 0.5x_t + 1.5$. Subtracting the fixed point 3: $x_{t+1} - 3 = 0.5(x_t - 3)$, so $x_t - 3 = (0.5)^t(x_0 - 3) = -3(0.5)^t$, i.e. $x_t = 3 - 3(0.5)^t$. The iterates are

$$0 \to 1.5 \to 2.25 \to 2.625 \to 2.8125 \to 2.906 \to \cdots \to 3,$$

converging geometrically to $x^\star = 3$ with ratio $\frac{1}{2} = |1 - \alpha f''|$. Any $\alpha \in (0, 2)$ converges here (since $f'' = 1$); $\alpha = 1$ would reach 3 in a single step, and $\alpha \geq 2$ would diverge. □

## Worked Problem Set C: short answers across the course

Five quick problems, each a self-contained drill on one tool.

### C1. Inverse and determinant (6 pts)

Find $\det A$ and $A^{-1}$ for $A = \left[\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix}\right]$.

*Solution.* $\det A = 1 \cdot 4 - 2 \cdot 3 = -2 \neq 0$, so $A$ is invertible. By the $2 \times 2$ formula (Eq. (1.10)), $A^{-1} = \frac{1}{-2}\left[\begin{smallmatrix} 4 & -2 \\ -3 & 1 \end{smallmatrix}\right] = \left[\begin{smallmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{smallmatrix}\right]$. Check $AA^{-1} = \left[\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix}\right]\left[\begin{smallmatrix} -2 & 1 \\ 1.5 & -0.5 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. □

### C2. Pseudoinverse of a singular matrix (8 pts)

Find $A^\dagger$ for the rank-one $A = \left[\begin{smallmatrix} 1 & 2 \\ 2 & 4 \end{smallmatrix}\right]$.

*Solution.* $A$ is singular ($\det = 0$), so $A^{-1}$ does not exist, but $A^\dagger$ always does. Write $A = \boldsymbol{u}\boldsymbol{u}^\top$ with $\boldsymbol{u} = (1, 2)$, since $\boldsymbol{u}\boldsymbol{u}^\top = \left[\begin{smallmatrix} 1 & 2 \\ 2 & 4 \end{smallmatrix}\right]$. In SVD terms $A = \sigma_1 \hat{\boldsymbol{u}}\hat{\boldsymbol{u}}^\top$ with $\hat{\boldsymbol{u}} = \boldsymbol{u}/\sqrt{5}$ and $\sigma_1 = \|\boldsymbol{u}\|^2 = 5$, so $A^\dagger = \sigma_1^{-1}\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}^\top = \frac{1}{5} \cdot \frac{1}{5}\boldsymbol{u}\boldsymbol{u}^\top = \frac{1}{25}\left[\begin{smallmatrix} 1 & 2 \\ 2 & 4 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0.04 & 0.08 \\ 0.08 & 0.16 \end{smallmatrix}\right]$. This $A^\dagger \boldsymbol{b}$ is the minimum-norm least-squares solution (Chapter 4). □

### C3. Variance of a sum (6 pts)

Two fair six-sided dice are rolled. Find the mean and variance of their sum.

*Solution.* For one die, $\mathbb{E}[X] = \frac{1+\cdots+6}{6} = 3.5$ and $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \frac{91}{6} - 3.5^2 = \frac{35}{12}$. The dice are independent, so means and *variances* add: $\mathbb{E}[\text{sum}] = 7$ and $\mathrm{Var}(\text{sum}) = 2 \cdot \frac{35}{12} = \frac{35}{6} \approx 5.83$. (Independence is essential for the variance step; without it a covariance term would appear.) □

### C4. Gradient, Hessian, convexity (6 pts)

For $f(x, y) = x^2 + xy + y^2$, give $\nabla f$ and $\nabla^2 f$, evaluate the gradient at $(1, 2)$, and decide whether $f$ is convex.

*Solution.* $\nabla f = (2x + y,\ x + 2y)$, so $\nabla f(1, 2) = (4, 5)$. The Hessian $\nabla^2 f = \left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ is constant with eigenvalues 3 and 1, both positive, so $\nabla^2 f \succ 0$ and $f$ is strictly convex; its unique minimum is at $\nabla f = \mathbf{0}$, i.e. $(0, 0)$. □

### C5. Eigenvalues, trace, determinant (6 pts)

Find the eigenvalues of $A = \left[\begin{smallmatrix} 4 & 1 \\ 2 & 3 \end{smallmatrix}\right]$ and verify the trace and determinant identities.

*Solution.* The characteristic polynomial is $(4 - \lambda)(3 - \lambda) - 2 = \lambda^2 - 7\lambda + 10 = (\lambda - 2)(\lambda - 5)$, so $\lambda = 2, 5$. Then $\lambda_1 + \lambda_2 = 7 = \operatorname{tr} A$ and $\lambda_1 \lambda_2 = 10 = \det A$, confirming Eq. (3.6). Both eigenvalues are nonzero, so $A$ is invertible. □

## Worked Problem Set D: more short answers across the course

These six problems each restate the question in full and work the algebra end to end, spanning projection, Gaussian conditioning, concentration, maximum likelihood, the softmax, and linear-programming duality.

### D1. Projection onto a line (6 pts)

Project $\boldsymbol{b} = (1, 2, 3)$ onto the line spanned by $\boldsymbol{a} = (1, 1, 1)$. Give the projection and the residual, and verify the residual is orthogonal to $\boldsymbol{a}$.

*Solution.* The projection of $\boldsymbol{b}$ onto $\mathrm{span}\{\boldsymbol{a}\}$ is $\hat{\boldsymbol{b}} = \dfrac{\boldsymbol{a}^\top \boldsymbol{b}}{\boldsymbol{a}^\top \boldsymbol{a}}\,\boldsymbol{a}$. Here $\boldsymbol{a}^\top \boldsymbol{b} = 1 + 2 + 3 = 6$ and $\boldsymbol{a}^\top \boldsymbol{a} = 1 + 1 + 1 = 3$, so the coefficient is $6/3 = 2$ and $\hat{\boldsymbol{b}} = 2(1, 1, 1) = (2, 2, 2)$. The residual is $\boldsymbol{r} = \boldsymbol{b} - \hat{\boldsymbol{b}} = (-1, 0, 1)$, and $\boldsymbol{r}^\top \boldsymbol{a} = -1 + 0 + 1 = 0$,

confirming orthogonality. The projection is the closest point of the line to $\boldsymbol{b}$, and the residual is the (perpendicular) error of that one-parameter least-squares fit. □

## D2. Gaussian conditioning (8 pts)

Let $(X, Y)$ be jointly Gaussian with zero means and covariance $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 4 & 2 \\ 2 & 3 \end{smallmatrix}\right]$. Find the conditional distribution of $Y$ given $X = x$.

*Solution.* For a joint Gaussian, $Y \mid X = x$ is Gaussian with mean $\mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(x - \mu_X)$ and variance $\Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$. With zero means and the entries above,

$$\mathbb{E}[Y \mid X = x] = \frac{\Sigma_{YX}}{\Sigma_{XX}}\, x = \frac{2}{4}\, x = 0.5\, x, \qquad \operatorname{Var}(Y \mid X) = \Sigma_{YY} - \frac{\Sigma_{YX}^2}{\Sigma_{XX}} = 3 - \frac{2^2}{4} = 2.$$

So $Y \mid X = x \sim \mathcal{N}(0.5x,\ 2)$. The conditional mean is linear in $x$ (this is exactly linear regression), and conditioning on $X$ shrinks the variance of $Y$ from 3 down to 2, the amount of uncertainty $X$ removes. □

## D3. A Hoeffding bound (6 pts)

A coin with $\mathbb{P}(\text{heads}) = p$ is flipped $n = 100$ times; let $\bar{X}$ be the sample fraction of heads. Bound $\mathbb{P}(|\bar{X} - p| \geq 0.1)$.

*Solution.* Hoeffding's inequality for an average of $n$ independent $[0, 1]$ variables gives $\mathbb{P}(|\bar{X} - \mathbb{E}[\bar{X}]| \geq \varepsilon) \leq 2\exp(-2n\varepsilon^2)$. With $n = 100$ and $\varepsilon = 0.1$,

$$2\exp\!\left(-2(100)(0.1)^2\right) = 2\exp(-2) \approx 2(0.135) = 0.271.$$

At most about a 27% chance the sample fraction is off by 0.1 or more, and the bound holds for *every* $p$, a distribution-free guarantee. Halving the tolerance to 0.05 would demand four times as many flips for the same bound, the hallmark $1/\varepsilon^2$ scaling. □

## D4. Maximum likelihood for a Poisson rate (6 pts)

Event counts in four equal intervals are $2, 3, 5, 2$, modeled as iid Poisson($\lambda$). Find the maximum likelihood estimate of $\lambda$.

*Solution.* The log-likelihood is $\ell(\lambda) = \sum_i \left(x_i \ln \lambda - \lambda - \ln x_i!\right)$. Differentiating and setting to zero,

$$\ell'(\lambda) = \frac{\sum_i x_i}{\lambda} - n = 0 \implies \hat{\lambda} = \frac{1}{n}\sum_i x_i = \frac{2+3+5+2}{4} = \frac{12}{4} = 3.$$

The Poisson rate's MLE is the sample mean of the counts, the same "MLE equals the empirical average" pattern seen for the Bernoulli bias and the Gaussian mean. □

## D5. Softmax and cross-entropy (8 pts)

A three-class classifier produces logits $\boldsymbol{z} = (2, 1, 0)$. Compute the softmax probabilities and the cross-entropy loss when the true class is the first.

*Solution.* The softmax is $p_k = e^{z_k} \big/ \sum_j e^{z_j}$. The exponentials are $e^2 = 7.389$, $e^1 = 2.718$, $e^0 = 1$, summing to 11.107, so

$$\boldsymbol{p} = \left(\tfrac{7.389}{11.107},\ \tfrac{2.718}{11.107},\ \tfrac{1}{11.107}\right) = (0.665,\ 0.245,\ 0.090).$$

The probabilities are positive and sum to 1. The cross-entropy loss looks only at the true class, here the first: $L = -\ln p_1 = -\ln(0.665) \approx 0.408$. As the model puts more mass on the correct class, $p_1 \to 1$ and the loss falls toward 0; this is the multiclass generalization of logistic regression's log loss. □

## D6. Linear programming and duality (10 pts)

For the linear program max $2x + 3y$ subject to $x + y \le 4$, $x + 2y \le 6$, $x, y \ge 0$: (a) solve it by enumerating vertices; (b) write the dual and verify strong duality.

*Solution.* (a) The feasible corners are $(0,0)$, $(4,0)$, $(0,3)$, and the intersection of $x+y=4$ with $x+2y=6$, which solves to $(x,y) = (2,2)$. The objective $2x+3y$ takes values $0,\ 8,\ 9,\ 10$ at these vertices, so the maximum is $10$ at $(2,2)$, where both inequality constraints are active. (b) With $\boldsymbol{c} = (2,3)$, $\boldsymbol{b} = (4,6)$, and $A = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, the dual is $\min 4u + 6v$ s.t. $u + v \ge 2$, $u + 2v \ge 3$, $u, v \ge 0$. Because both primal constraints are active, both dual variables are positive, so by complementary slackness both dual constraints are tight: $u + v = 2$ and $u + 2v = 3$, giving $v = 1$ and $u = 1$. The dual objective is $4(1) + 6(1) = 10$, equal to the primal optimum, confirming strong duality. The dual values $u = v = 1$ are the shadow prices of the two resource limits. $\square$

*For more practice, every chapter's exercises are drawn from the course's problem sets and examinations; work them first, then attempt these papers closed-book under time.*

# Bibliography


[1] David Austin. We recommend a singular value decomposition. American Mathematical Society, Feature Column, 2009. https://www.ams.org/publicoutreach/feature-column/fcarc-svd.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. https://web.stanford.edu/~boyd/cvxbook/.

[4] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[5] Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning*. Cambridge University Press, Cambridge, UK, 2020. Freely available at https://mml-book.github.io.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[7] Alhussein Fawzi, Matej Balog, Aja Huang, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610:47–53, 2022.

[8] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 4 edition, 2013.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016. https://www.deeplearningbook.org.

[10] Charles R. Harris et al. Array programming with NumPy. *Nature*, 585:357–362, 2020.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2 edition, 2009.

[12] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, UK, 2 edition, 2012.

[13] Aapo Hyvärinen and Erkki Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5):411–430, 2000.

[14] I. T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2 edition, 2002.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

[16] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 2 edition, 2006.

[17] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

[18] John C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, 1998.

[19] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[20] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, Cambridge, UK, 2014.

[21] Gilbert Strang. *Introduction to Linear Algebra*. Wellesley–Cambridge Press, Wellesley, MA, 5 edition, 2016.

[22] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.

[23] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2 edition, 2018.

[24] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1997.