Part I

Mathematical Foundations

1

Introduction and Motivation

Machine learning is about designing algorithms that automatically extract valuable information from data. The emphasis here is on "automatic", i.e., machine learning is concerned about general-purpose methodologies that can be applied to many datasets, while producing something that is meaningful. There are three concepts that are at the core of machine learning: data, a model, and learning.

Since machine learning is inherently data driven, *data* is at the core of machine learning. The goal of machine learning is to design general-purpose methodologies to extract valuable patterns from data, ideally without much domain-specific expertise. For example, given a large corpus of documents (e.g., books in many libraries), machine learning methods can be used to automatically find relevant topics that are shared across documents (Hoffman et al., 2010). To achieve this goal, we design *models* that are typically related to the process that generates data, similar to the dataset we are given. For example, in a regression setting, the model would describe a function that maps inputs to real-valued outputs. To paraphrase Mitchell (1997): A model is said to learn from data if its performance on a given task improves after the data is taken into account. The goal is to find good models that generalize well to yet unseen data, which we may care about in the future. *Learning* can be understood as a way to automatically find patterns and structure in data by optimizing the parameters of the model.

data

model

learning

While machine learning has seen many success stories, and software is readily available to design and train rich and flexible machine learning systems, we believe that the mathematical foundations of machine learning are important in order to understand fundamental principles upon which more complicated machine learning systems are built. Understanding these principles can facilitate creating new machine learning solutions, understanding and debugging existing approaches, and learning about the inherent assumptions and limitations of the methodologies we are working with.

1.1 Finding Words for Intuitions

A challenge we face regularly in machine learning is that concepts and words are slippery, and a particular component of the machine learning system can be abstracted to different mathematical concepts. For example, the word "algorithm" is used in at least two different senses in the context of machine learning. In the first sense, we use the phrase "machine learning algorithm" to mean a system that makes predictions based on input data. We refer to these algorithms as *predictors*. In the second sense, we use the exact same phrase "machine learning algorithm" to mean a system that adapts some internal parameters of the predictor so that it performs well on future unseen input data. Here we refer to this adaptation as *training* a system.

predictor

training

This book will not resolve the issue of ambiguity, but we want to highlight upfront that, depending on the context, the same expressions can mean different things. However, we attempt to make the context sufficiently clear to reduce the level of ambiguity.

The first part of this book introduces the mathematical concepts and foundations needed to talk about the three main components of a machine learning system: data, models, and learning. We will briefly outline these components here, and we will revisit them again in Chapter 8 once we have discussed the necessary mathematical concepts.

While not all data is numerical, it is often useful to consider data in a number format. In this book, we assume that *data* has already been appropriately converted into a numerical representation suitable for reading into a computer program. Therefore, we think of data as vectors. As another illustration of how subtle words are, there are (at least) three different ways to think about vectors: a vector as an array of numbers (a computer science view), a vector as an arrow with a direction and magnitude (a physics view), and a vector as an object that obeys addition and scaling (a mathematical view).

data as vectors

A *model* is typically used to describe a process for generating data, similar to the dataset at hand. Therefore, good models can also be thought of as simplified versions of the real (unknown) data-generating process, capturing aspects that are relevant for modeling the data and extracting hidden patterns from it. A good model can then be used to predict what would happen in the real world without performing real-world experiments.

model

We now come to the crux of the matter, the *learning* component of machine learning. Assume we are given a dataset and a suitable model. *Training* the model means to use the data available to optimize some parameters of the model with respect to a utility function that evaluates how well the model predicts the training data. Most training methods can be thought of as an approach analogous to climbing a hill to reach its peak. In this analogy, the peak of the hill corresponds to a maximum of some

learning

desired performance measure. However, in practice, we are interested in the model to perform well on unseen data. Performing well on data that we have already seen (training data) may only mean that we found a good way to memorize the data. However, this may not generalize well to unseen data, and, in practical applications, we often need to expose our machine learning system to situations that it has not encountered before.

Let us summarize the main concepts of machine learning that we cover in this book:

- We represent data as vectors.
- We choose an appropriate model, either using the probabilistic or optimization view.
- We learn from available data by using numerical optimization methods with the aim that the model performs well on data not used for training.

1.2 Two Ways to Read This Book

We can consider two strategies for understanding the mathematics for machine learning:

- **Bottom-up:** Building up the concepts from foundational to more advanced. This is often the preferred approach in more technical fields, such as mathematics. This strategy has the advantage that the reader at all times is able to rely on their previously learned concepts. Unfortunately, for a practitioner many of the foundational concepts are not particularly interesting by themselves, and the lack of motivation means that most foundational definitions are quickly forgotten.
- **Top-down:** Drilling down from practical needs to more basic requirements. This goal-driven approach has the advantage that the readers know at all times why they need to work on a particular concept, and there is a clear path of required knowledge. The downside of this strategy is that the knowledge is built on potentially shaky foundations, and the readers have to remember a set of words that they do not have any way of understanding.

We decided to write this book in a modular way to separate foundational (mathematical) concepts from applications so that this book can be read in both ways. The book is split into two parts, where Part I lays the mathematical foundations and Part II applies the concepts from Part I to a set of fundamental machine learning problems, which form four pillars of machine learning as illustrated in Figure 1.2: regression, dimensionality reduction, density estimation, and classification. Chapters in Part I mostly build upon the previous ones, but it is possible to skip a chapter and work backward if necessary. Chapters in Part II are only loosely coupled and can be read in any order. There are many pointers forward and backward

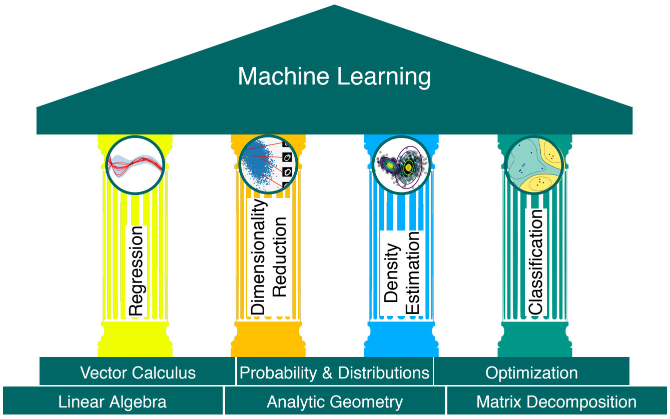


Figure 1.2 The foundations and four pillars of machine learning.

between the two parts of the book to link mathematical concepts with machine learning algorithms.

Of course there are more than two ways to read this book. Most readers learn using a combination of top-down and bottom-up approaches, sometimes building up basic mathematical skills before attempting more complex concepts, but also choosing topics based on applications of machine learning.

Part I Is about Mathematics

The four pillars of machine learning we cover in this book (see Figure 1.2) require a solid mathematical foundation, which is laid out in Part I.

We represent numerical data as vectors and represent a table of such data as a matrix. The study of vectors and matrices is called *linear algebra*, which we introduce in Chapter 2. The collection of vectors as a matrix is also described there.

linear algebra

Given two vectors representing two objects in the real world, we want to make statements about their similarity. The idea is that vectors that are similar should be predicted to have similar outputs by our machine learning algorithm (our predictor). To formalize the idea of similarity between vectors, we need to introduce operations that take two vectors as input and return a numerical value representing their similarity. The construction of similarity and distances is central to *analytic geometry* and is discussed in Chapter 3.

analytic geometry

In Chapter 4, we introduce some fundamental concepts about matrices and *matrix decomposition*. Some operations on matrices are extremely useful in machine learning, and they allow for an intuitive interpretation of the data and more efficient learning.

matrix decomposition

We often consider data to be noisy observations of some true underlying signal. We hope that by applying machine learning we can identify the signal from the noise. This requires us to have a language for quantifying what "noise" means. We often would also like to have predictors that allow us to express some sort of uncertainty, e.g., to quantify the confidence we have about the value of the prediction at a particular test data point. Quantification of uncertainty is the realm of *probability theory* and is covered in Chapter 6. probability theory

To train machine learning models, we typically find parameters that maximize some performance measure. Many optimization techniques require the concept of a gradient, which tells us the direction in which to search for a solution. Chapter 5 is about *vector calculus* and details the concept of gradients, which we subsequently use in Chapter 7, where we talk about *optimization* to find maxima/minima of functions. vector calculus optimization

Part II Is about Machine Learning

The second part of the book introduces *four pillars of machine learning* as shown in Figure 1.2. We illustrate how the mathematical concepts introduced in the first part of the book are the foundation for each pillar. Broadly speaking, chapters are ordered by difficulty (in ascending order).

In Chapter 8, we restate the three components of machine learning (data, models, and parameter estimation) in a mathematical fashion. In addition, we provide some guidelines for building experimental set-ups that guard against overly optimistic evaluations of machine learning systems. Recall that the goal is to build a predictor that performs well on unseen data.

In Chapter 9, we will have a close look at *linear regression*, where our objective is to find functions that map inputs $\boldsymbol{x} \in \mathbb{R}^D$ to corresponding observed function values $y \in \mathbb{R}$, which we can interpret as the labels of their respective inputs. We will discuss classical model fitting (parameter estimation) via maximum likelihood and maximum a posteriori estimation, as well as Bayesian linear regression, where we integrate the parameters out instead of optimizing them. linear regression

Chapter 10 focuses on *dimensionality reduction*, the second pillar in Figure 1.2, using principal component analysis. The key objective of dimensionality reduction is to find a compact, lower-dimensional representation of high-dimensional data $\boldsymbol{x} \in \mathbb{R}^D$, which is often easier to analyze than the original data. Unlike regression, dimensionality reduction is only concerned about modeling the data – there are no labels associated with a data point $\boldsymbol{x}$. dimensionality reduction

In Chapter 11, we will move to our third pillar: *density estimation*. The objective of density estimation is to find a probability distribution that describes a given dataset. We will focus on Gaussian mixture models for this purpose, and we will discuss an iterative scheme to find the parameters of this model. As in dimensionality reduction, there are no labels associated with the data points $\boldsymbol{x} \in \mathbb{R}^D$. However, we do not seek a low-dimensional representation of the data. Instead, we are interested in a density model that describes the data. density estimation

Chapter 12 concludes the book with an in-depth discussion of the fourth classification

pillar: *classification*. We will discuss classification in the context of support vector machines. Similar to regression (Chapter 9), we have inputs $\boldsymbol{x}$ and corresponding labels y. However, unlike regression, where the labels were real-valued, the labels in classification are integers, which requires special care.

1.3 Exercises and Feedback

We provide some exercises in Part I, which can be done mostly by pen and paper. For Part II, we provide programming tutorials (jupyter notebooks) to explore some properties of the machine learning algorithms we discuss in this book.

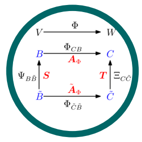
We appreciate that Cambridge University Press strongly supports our aim to democratize education and learning by making this book freely available for download at

`https://mml-book.com`

where tutorials, errata, and additional materials can be found. Mistakes can be reported and feedback provided using the preceding URL.

2

Linear Algebra

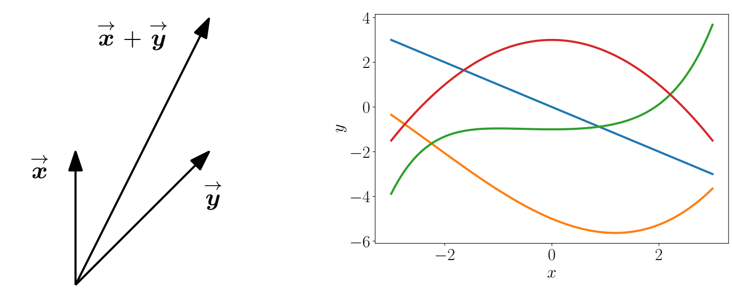


When formalizing intuitive concepts, a common approach is to construct a set of objects (symbols) and a set of rules to manipulate these objects. This is known as an *algebra*. Linear algebra is the study of vectors and certain rules to manipulate vectors. The vectors many of us know from school are called "geometric vectors", which are usually denoted by a small arrow above the letter, e.g., $\overrightarrow{x}$ and $\overrightarrow{y}$. In this book, we discuss more general concepts of vectors and use a bold letter to represent them, e.g., $\boldsymbol{x}$ and $\boldsymbol{y}$.

algebra

In general, vectors are special objects that can be added together and multiplied by scalars to produce another object of the same kind. From an abstract mathematical viewpoint, any object that satisfies these two properties can be considered a vector. Here are some examples of such vector objects:

1. Geometric vectors. This example of a vector may be familiar from high school mathematics and physics. Geometric vectors – see Figure 2.1(a) – are directed segments, which can be drawn (at least in two dimensions). Two geometric vectors $\vec{\boldsymbol{x}}, \vec{\boldsymbol{y}}$ can be added, such that $\vec{\boldsymbol{x}}+\vec{\boldsymbol{y}} = \vec{\boldsymbol{z}}$ is another geometric vector. Furthermore, multiplication by a scalar $\lambda\vec{\boldsymbol{x}}$, $\lambda \in \mathbb{R}$, is also a geometric vector. In fact, it is the original vector scaled by λ. Therefore, geometric vectors are instances of the vector concepts introduced previously. Interpreting vectors as geometric vectors enables us to use our intuitions about direction and magnitude to reason about mathematical operations.
2. Polynomials are also vectors; see Figure 2.1(b): Two polynomials can



(a) Geometric vectors.

(b) Polynomials.

Figure 2.1 Different types of vectors. Vectors can be surprising objects, including (a) geometric vectors and (b) polynomials.

be added together, which results in another polynomial; and they can be multiplied by a scalar $\lambda \in \mathbb{R}$, and the result is a polynomial as well. Therefore, polynomials are (rather unusual) instances of vectors. Note that polynomials are very different from geometric vectors. While geometric vectors are concrete "drawings", polynomials are abstract concepts. However, they are both vectors in the sense previously described.

3. Audio signals are vectors. Audio signals are represented as a series of numbers. We can add audio signals together, and their sum is a new audio signal. If we scale an audio signal, we also obtain an audio signal. Therefore, audio signals are a type of vector, too.
4. Elements of $\mathbb{R}^n$ (tuples of n real numbers) are vectors. $\mathbb{R}^n$ is more abstract than polynomials, and it is the concept we focus on in this book. For instance,

$$\boldsymbol{a} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \in \mathbb{R}^3 \tag{2.1}$$

is an example of a triplet of numbers. Adding two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ component-wise results in another vector: $\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{c} \in \mathbb{R}^n$. Moreover, multiplying $\boldsymbol{a} \in \mathbb{R}^n$ by $\lambda \in \mathbb{R}$ results in a scaled vector $\lambda\boldsymbol{a} \in \mathbb{R}^n$. Considering vectors as elements of $\mathbb{R}^n$ has an additional benefit that it loosely corresponds to arrays of real numbers on a computer. Many programming languages support array operations, which allow for convenient implementation of algorithms that involve vector operations.

Be careful to check whether array operations actually perform vector operations when implementing on a computer.

Linear algebra focuses on the similarities between these vector concepts. We can add them together and multiply them by scalars. We will largely focus on vectors in $\mathbb{R}^n$ since most algorithms in linear algebra are formulated in $\mathbb{R}^n$. We will see in Chapter 8 that we often consider data to be represented as vectors in $\mathbb{R}^n$. In this book, we will focus on finite-dimensional vector spaces, in which case there is a 1:1 correspondence between any kind of vector and $\mathbb{R}^n$. When it is convenient, we will use intuitions about geometric vectors and consider array-based algorithms.

Pavel Grinfeld's series on linear algebra: `http://tinyurl.com/nahclwm`

Gilbert Strang's course on linear algebra: `http://tinyurl.com/29p5q8j`

3Blue1Brown series on linear algebra: `https://tinyurl.com/h5g4kps`

One major idea in mathematics is the idea of "closure". This is the question: What is the set of all things that can result from my proposed operations? In the case of vectors: What is the set of vectors that can result by starting with a small set of vectors, and adding them to each other and scaling them? This results in a vector space (Section 2.4). The concept of a vector space and its properties underlie much of machine learning. The concepts introduced in this chapter are summarized in Figure 2.2.

This chapter is mostly based on the lecture notes and books by Drumm and Weil (2001), Strang (2003), Hogben (2013), Liesen and Mehrmann (2015), as well as Pavel Grinfeld's Linear Algebra series. Other excellent

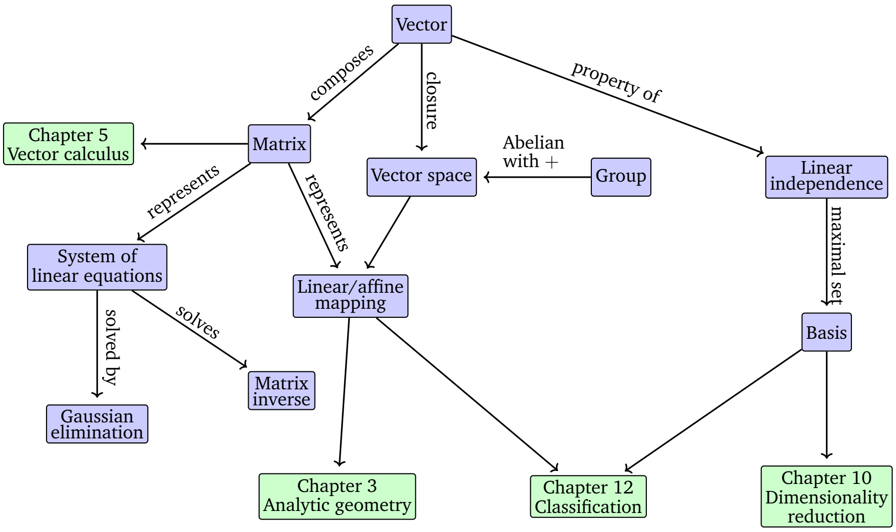


Figure 2.2 A mind map of the concepts introduced in this chapter, along with where they are used in other parts of the book.

resources are Gilbert Strang's Linear Algebra course at MIT and the Linear Algebra Series by 3Blue1Brown.

Linear algebra plays an important role in machine learning and general mathematics. The concepts introduced in this chapter are further expanded to include the idea of geometry in Chapter 3. In Chapter 5, we will discuss vector calculus, where a principled knowledge of matrix operations is essential. In Chapter 10, we will use projections (to be introduced in Section 3.8) for dimensionality reduction with principal component analysis (PCA). In Chapter 9, we will discuss linear regression, where linear algebra plays a central role for solving least-squares problems.

2.1 Systems of Linear Equations

Systems of linear equations play a central part of linear algebra. Many problems can be formulated as systems of linear equations, and linear algebra gives us the tools for solving them.

Example 2.1

A company produces products $N_1, \ldots, N_n$ for which resources $R_1, \ldots, R_m$ are required. To produce a unit of product N_j, a_{ij} units of resource R_i are needed, where $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

The objective is to find an optimal production plan, i.e., a plan of how many units x_j of product N_j should be produced if a total of b_i units of resource R_i are available and (ideally) no resources are left over.

If we produce $x_1, \ldots, x_n$ units of the corresponding products, we need a total of

$$a_{i1}x_1 + \cdots + a_{in}x_n \tag{2.2}$$

many units of resource R_i. An optimal production plan $(x_1, \ldots, x_n) \in \mathbb{R}^n$, therefore, has to satisfy the following system of equations:

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m \end{aligned}\,, \tag{2.3}$$

where $a_{ij} \in \mathbb{R}$ and $b_i \in \mathbb{R}$.

system of linear equations

solution

Equation (2.3) is the general form of a *system of linear equations*, and $x_1, \ldots, x_n$ are the *unknowns* of this system. Every n-tuple $(x_1, \ldots, x_n) \in \mathbb{R}^n$ that satisfies (2.3) is a *solution* of the linear equation system.

Example 2.2

The system of linear equations

$$\begin{array}{ccccccccc} x_1 & + & x_2 & + & x_3 & = & 3 & & (1) \\ x_1 & - & x_2 & + & 2x_3 & = & 2 & & (2) \\ 2x_1 & & & + & 3x_3 & = & 1 & & (3) \end{array} \tag{2.4}$$

has *no solution:* Adding the first two equations yields $2x_1 + 3x_3 = 5$, which contradicts the third equation (3).

Let us have a look at the system of linear equations

$$\begin{array}{ccccccccc} x_1 & + & x_2 & + & x_3 & = & 3 & & (1) \\ x_1 & - & x_2 & + & 2x_3 & = & 2 & & (2) \\ & & x_2 & + & x_3 & = & 2 & & (3) \end{array}\,. \tag{2.5}$$

From the first and third equation, it follows that $x_1 = 1$. From (1)+(2), we get $2x_1 + 3x_3 = 5$, i.e., $x_3 = 1$. From (3), we then get that $x_2 = 1$. Therefore, $(1, 1, 1)$ is the only possible and *unique solution* (verify that $(1, 1, 1)$ is a solution by plugging in).

As a third example, we consider

$$\begin{array}{ccccccccc} x_1 & + & x_2 & + & x_3 & = & 3 & & (1) \\ x_1 & - & x_2 & + & 2x_3 & = & 2 & & (2) \\ 2x_1 & & & + & 3x_3 & = & 5 & & (3) \end{array}\,. \tag{2.6}$$

Since (1)+(2)=(3), we can omit the third equation (redundancy). From (1) and (2), we get $2x_1 = 5 - 3x_3$ and $2x_2 = 1 + x_3$. We define $x_3 = a \in \mathbb{R}$ as a free variable, such that any triplet

$$\left(\frac{5}{2} - \frac{3}{2}a, \frac{1}{2} + \frac{1}{2}a, a\right), \quad a \in \mathbb{R} \tag{2.7}$$

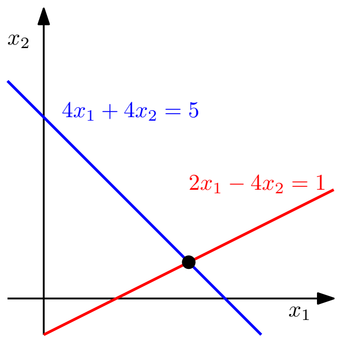


Figure 2.1 The solution space of a system of two linear equations with two variables can be geometrically interpreted as the intersection of two lines. Every linear equation represents a line.

is a solution of the system of linear equations, i.e., we obtain a solution set that contains *infinitely many* solutions.

In general, for a real-valued system of linear equations we obtain either no, exactly one, or infinitely many solutions. Linear regression (Chapter 9) solves a version of Example 2.1 when we cannot solve the system of linear equations.

Remark (Geometric Interpretation of Systems of Linear Equations). In a system of linear equations with two variables x_1, x_2, each linear equation defines a line on the x_1x_2-plane. Since a solution to a system of linear equations must satisfy all equations simultaneously, the solution set is the intersection of these lines. This intersection set can be a line (if the linear equations describe the same line), a point, or empty (when the lines are parallel). An illustration is given in Figure 2.1 for the system

$$\begin{aligned} 4x_1 + 4x_2 &= 5 \\ 2x_1 - 4x_2 &= 1 \end{aligned} \tag{2.8}$$

where the solution space is the point $(x_1, x_2) = (1, \frac{1}{4})$. Similarly, for three variables, each linear equation determines a plane in three-dimensional space. When we intersect these planes, i.e., satisfy all linear equations at the same time, we can obtain a solution set that is a plane, a line, a point or empty (when the planes have no common intersection). $\diamondsuit$

For a systematic approach to solving systems of linear equations, we will introduce a useful compact notation. We collect the coefficients a_{ij} into vectors and collect the vectors into matrices. In other words, we write the system from (2.3) in the following form:

$$\begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \end{bmatrix} x_1 + \begin{bmatrix} a_{12} \\ \vdots \\ a_{m2} \end{bmatrix} x_2 + \cdots + \begin{bmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{bmatrix} x_n = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \tag{2.9}$$

$$\iff \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}. \tag{2.10}$$

In the following, we will have a close look at these *matrices* and define computation rules. We will return to solving linear equations in Section 2.3.

2.2 Matrices

Matrices play a central role in linear algebra. They can be used to compactly represent systems of linear equations, but they also represent linear functions (linear mappings) as we will see later in Section 2.7. Before we discuss some of these interesting topics, let us first define what a matrix is and what kind of operations we can do with matrices. We will see more properties of matrices in Chapter 4.

matrix

Definition 2.1 (Matrix)**.** With $m, n \in \mathbb{N}$ a real-valued (m, n) *matrix* $\boldsymbol{A}$ is an $m \cdot n$-tuple of elements a_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$, which is ordered according to a rectangular scheme consisting of m rows and n columns:

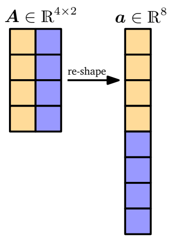
$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad a_{ij} \in \mathbb{R}. \tag{2.11}$$

row
column
row vector
column vector

By convention $(1, n)$-matrices are called *rows* and $(m, 1)$-matrices are called *columns*. These special matrices are also called *row/column vectors*.

$\mathbb{R}^{m \times n}$ is the set of all real-valued (m, n)-matrices. $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ can be equivalently represented as $\boldsymbol{a} \in \mathbb{R}^{mn}$ by stacking all n columns of the matrix into a long vector; see Figure 2.2.

Figure 2.2 By stacking its columns, a matrix $\boldsymbol{A}$ can be represented as a long vector $\boldsymbol{a}$.



2.2.1 Matrix Addition and Multiplication

The sum of two matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{B} \in \mathbb{R}^{m \times n}$ is defined as the element-wise sum, i.e.,

$$\boldsymbol{A} + \boldsymbol{B} := \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}. \tag{2.12}$$

Note the size of the matrices.

```
C =
np.einsum('il,
lj', A, B)
```

For matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{B} \in \mathbb{R}^{n \times k}$, the elements c_{ij} of the product $\boldsymbol{C} = \boldsymbol{AB} \in \mathbb{R}^{m \times k}$ are computed as

$$c_{ij} = \sum_{l=1}^{n} a_{il} b_{lj}, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, k. \tag{2.13}$$

This means, to compute element c_{ij} we multiply the elements of the ith row of $\boldsymbol{A}$ with the jth column of $\boldsymbol{B}$ and sum them up. Later in Section 3.2, we will call this the *dot product* of the corresponding row and column. In cases, where we need to be explicit that we are performing multiplication, we use the notation $\boldsymbol{A} \cdot \boldsymbol{B}$ to denote multiplication (explicitly showing "·").

There are n columns in $\boldsymbol{A}$ and n rows in $\boldsymbol{B}$ so that we can compute $a_{il}b_{lj}$ for $l = 1, \dots, n$. Commonly, the dot product between two vectors $\boldsymbol{a}, \boldsymbol{b}$ is denoted by $\boldsymbol{a}^\top \boldsymbol{b}$ or $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$.

Remark. Matrices can only be multiplied if their "neighboring" dimensions match. For instance, an $n \times k$-matrix $\boldsymbol{A}$ can be multiplied with a $k \times m$-matrix $\boldsymbol{B}$, but only from the left side:

$$\underbrace{\boldsymbol{A}}_{n\times k}\underbrace{\boldsymbol{B}}_{k\times m} = \underbrace{\boldsymbol{C}}_{n\times m} \tag{2.14}$$

The product $\boldsymbol{BA}$ is not defined if $m \neq n$ since the neighboring dimensions do not match. ◇

Remark. Matrix multiplication is *not* defined as an element-wise operation on matrix elements, i.e., $c_{ij} \neq a_{ij}b_{ij}$ (even if the size of $\boldsymbol{A}, \boldsymbol{B}$ was chosen appropriately). This kind of element-wise multiplication often appears in programming languages when we multiply (multi-dimensional) arrays with each other, and is called a *Hadamard product*. ◇

Hadamard product

Example 2.3

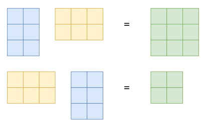
For $\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix} \in \mathbb{R}^{2\times 3}$, $\boldsymbol{B} = \begin{bmatrix} 0 & 2 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 2}$, we obtain

$$\boldsymbol{AB} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 2 & 5 \end{bmatrix} \in \mathbb{R}^{2\times 2}, \tag{2.15}$$

$$\boldsymbol{BA} = \begin{bmatrix} 0 & 2 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 4 & 2 \\ -2 & 0 & 2 \\ 3 & 2 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3}. \tag{2.16}$$

From this example, we can already see that matrix multiplication is not commutative, i.e., $\boldsymbol{AB} \neq \boldsymbol{BA}$; see also Figure 2.3 for an illustration.

Figure 2.3 Even if both matrix multiplications $\boldsymbol{AB}$ and $\boldsymbol{BA}$ are defined, the dimensions of the results can be different.



Definition 2.2 (Identity Matrix). In $\mathbb{R}^{n\times n}$, we define the *identity matrix*

identity matrix

$$\boldsymbol{I}_n := \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{n\times n} \tag{2.17}$$

as the $n \times n$-matrix containing 1 on the diagonal and 0 everywhere else.

Now that we defined matrix multiplication, matrix addition and the identity matrix, let us have a look at some properties of matrices:

associativity

- *Associativity:*

$$\forall \boldsymbol{A} \in \mathbb{R}^{m\times n}, \boldsymbol{B} \in \mathbb{R}^{n\times p}, \boldsymbol{C} \in \mathbb{R}^{p\times q} : (\boldsymbol{AB})\boldsymbol{C} = \boldsymbol{A}(\boldsymbol{BC}) \tag{2.18}$$

distributivity

- *Distributivity:*

$$\forall \boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times n}, \boldsymbol{C}, \boldsymbol{D} \in \mathbb{R}^{n\times p} : (\boldsymbol{A}+\boldsymbol{B})\boldsymbol{C} = \boldsymbol{AC} + \boldsymbol{BC} \tag{2.19a}$$

$$\boldsymbol{A}(\boldsymbol{C}+\boldsymbol{D}) = \boldsymbol{AC} + \boldsymbol{AD} \tag{2.19b}$$

- Multiplication with the identity matrix:

$$\forall \boldsymbol{A} \in \mathbb{R}^{m\times n} : \boldsymbol{I}_m\boldsymbol{A} = \boldsymbol{A}\boldsymbol{I}_n = \boldsymbol{A} \tag{2.20}$$

Note that $\boldsymbol{I}_m \neq \boldsymbol{I}_n$ for $m \neq n$.

2.2.2 Inverse and Transpose

A square matrix possesses the same number of columns and rows.

inverse

Definition 2.3 (Inverse). Consider a square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. Let matrix $\boldsymbol{B} \in \mathbb{R}^{n\times n}$ have the property that $\boldsymbol{AB} = \boldsymbol{I}_n = \boldsymbol{BA}$. $\boldsymbol{B}$ is called the *inverse* of $\boldsymbol{A}$ and denoted by $\boldsymbol{A}^{-1}$.

regular
invertible
nonsingular
singular
noninvertible

Unfortunately, not every matrix $\boldsymbol{A}$ possesses an inverse $\boldsymbol{A}^{-1}$. If this inverse does exist, $\boldsymbol{A}$ is called *regular/invertible/nonsingular*, otherwise *singular/noninvertible*. When the matrix inverse exists, it is unique. In Section 2.3, we will discuss a general way to compute the inverse of a matrix by solving a system of linear equations.

Remark (Existence of the Inverse of a 2×2-matrix). Consider a matrix

$$\boldsymbol{A} := \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \in \mathbb{R}^{2\times 2}\,. \tag{2.21}$$

If we multiply $\boldsymbol{A}$ with

$$\boldsymbol{A}' := \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \tag{2.22}$$

we obtain

$$\boldsymbol{AA}' = \begin{bmatrix} a_{11}a_{22} - a_{12}a_{21} & 0 \\ 0 & a_{11}a_{22} - a_{12}a_{21} \end{bmatrix} = (a_{11}a_{22} - a_{12}a_{21})\boldsymbol{I}\,. \tag{2.23}$$

Therefore,

$$\boldsymbol{A}^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \tag{2.24}$$

if and only if $a_{11}a_{22} - a_{12}a_{21} \neq 0$. In Section 4.1, we will see that $a_{11}a_{22} -$

$a_{12}a_{21}$ is the determinant of a 2×2-matrix. Furthermore, we can generally use the determinant to check whether a matrix is invertible. $\diamondsuit$

Example 2.4 (Inverse Matrix)
The matrices

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 1 \\ 4 & 4 & 5 \\ 6 & 7 & 7 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} -7 & -7 & 6 \\ 2 & 1 & -1 \\ 4 & 5 & -4 \end{bmatrix} \tag{2.25}$$

are inverse to each other since $\boldsymbol{AB} = \boldsymbol{I} = \boldsymbol{BA}$.

Definition 2.4 (Transpose)**.** For $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ the matrix $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ with $b_{ij} = a_{ji}$ is called the *transpose* of $\boldsymbol{A}$. We write $\boldsymbol{B} = \boldsymbol{A}^\top$.

transpose

In general, $\boldsymbol{A}^\top$ can be obtained by writing the columns of $\boldsymbol{A}$ as the rows of $\boldsymbol{A}^\top$. The following are important properties of inverses and transposes:

The main diagonal (sometimes called "principal diagonal", "primary diagonal", "leading diagonal", or "major diagonal") of a matrix $\boldsymbol{A}$ is the collection of entries A_{ij} where $i = j$.

The scalar case of (2.28) is $\frac{1}{2+4} = \frac{1}{6} \neq \frac{1}{2} + \frac{1}{4}$.

$$\boldsymbol{AA}^{-1} = \boldsymbol{I} = \boldsymbol{A}^{-1}\boldsymbol{A} \tag{2.26}$$

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1} \tag{2.27}$$

$$(\boldsymbol{A}+\boldsymbol{B})^{-1} \neq \boldsymbol{A}^{-1} + \boldsymbol{B}^{-1} \tag{2.28}$$

$$(\boldsymbol{A}^\top)^\top = \boldsymbol{A} \tag{2.29}$$

$$(\boldsymbol{A}+\boldsymbol{B})^\top = \boldsymbol{A}^\top + \boldsymbol{B}^\top \tag{2.30}$$

$$(\boldsymbol{AB})^\top = \boldsymbol{B}^\top\boldsymbol{A}^\top \tag{2.31}$$

Definition 2.5 (Symmetric Matrix)**.** A matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is *symmetric* if $\boldsymbol{A} = \boldsymbol{A}^\top$.

symmetric matrix

Note that only (n, n)-matrices can be symmetric. Generally, we call (n, n)-matrices also *square matrices* because they possess the same number of rows and columns. Moreover, if $\boldsymbol{A}$ is invertible, then so is $\boldsymbol{A}^\top$, and $(\boldsymbol{A}^{-1})^\top = (\boldsymbol{A}^\top)^{-1} =: \boldsymbol{A}^{-\top}$.

square matrix

Remark (Sum and Product of Symmetric Matrices)*.* The sum of symmetric matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$ is always symmetric. However, although their product is always defined, it is generally not symmetric:

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}. \tag{2.32}$$

$\diamondsuit$

2.2.3 Multiplication by a Scalar

Let us look at what happens to matrices when they are multiplied by a scalar $\lambda \in \mathbb{R}$. Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\lambda \in \mathbb{R}$. Then $\lambda\boldsymbol{A} = \boldsymbol{K}$, $K_{ij} = \lambda\, a_{ij}$. Practically, λ scales each element of $\boldsymbol{A}$. For $\lambda, \psi \in \mathbb{R}$, the following holds:

associativity

distributivity

- *Associativity:*
  $(\lambda\psi)\boldsymbol{C} = \lambda(\psi\boldsymbol{C}), \quad \boldsymbol{C} \in \mathbb{R}^{m\times n}$
- $\lambda(\boldsymbol{BC}) = (\lambda\boldsymbol{B})\boldsymbol{C} = \boldsymbol{B}(\lambda\boldsymbol{C}) = (\boldsymbol{BC})\lambda, \quad \boldsymbol{B} \in \mathbb{R}^{m\times n}, \boldsymbol{C} \in \mathbb{R}^{n\times k}$.
  Note that this allows us to move scalar values around.
- $(\lambda\boldsymbol{C})^\top = \boldsymbol{C}^\top\lambda^\top = \boldsymbol{C}^\top\lambda = \lambda\boldsymbol{C}^\top$ since $\lambda = \lambda^\top$ for all $\lambda \in \mathbb{R}$.
- *Distributivity:*
  $(\lambda + \psi)\boldsymbol{C} = \lambda\boldsymbol{C} + \psi\boldsymbol{C}, \quad \boldsymbol{C} \in \mathbb{R}^{m\times n}$
  $\lambda(\boldsymbol{B} + \boldsymbol{C}) = \lambda\boldsymbol{B} + \lambda\boldsymbol{C}, \quad \boldsymbol{B}, \boldsymbol{C} \in \mathbb{R}^{m\times n}$

Example 2.5 (Distributivity)
If we define

$$\boldsymbol{C} := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \tag{2.33}$$

then for any $\lambda, \psi \in \mathbb{R}$ we obtain

$$(\lambda + \psi)\boldsymbol{C} = \begin{bmatrix} (\lambda+\psi)1 & (\lambda+\psi)2 \\ (\lambda+\psi)3 & (\lambda+\psi)4 \end{bmatrix} = \begin{bmatrix} \lambda+\psi & 2\lambda+2\psi \\ 3\lambda+3\psi & 4\lambda+4\psi \end{bmatrix} \tag{2.34a}$$

$$= \begin{bmatrix} \lambda & 2\lambda \\ 3\lambda & 4\lambda \end{bmatrix} + \begin{bmatrix} \psi & 2\psi \\ 3\psi & 4\psi \end{bmatrix} = \lambda\boldsymbol{C} + \psi\boldsymbol{C}. \tag{2.34b}$$

2.2.4 Compact Representations of Systems of Linear Equations

If we consider the system of linear equations

$$\begin{aligned} 2x_1 + 3x_2 + 5x_3 &= 1 \\ 4x_1 - 2x_2 - 7x_3 &= 8 \\ 9x_1 + 5x_2 - 3x_3 &= 2 \end{aligned} \tag{2.35}$$

and use the rules for matrix multiplication, we can write this equation system in a more compact form as

$$\begin{bmatrix} 2 & 3 & 5 \\ 4 & -2 & -7 \\ 9 & 5 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ 2 \end{bmatrix}. \tag{2.36}$$

Note that x_1 scales the first column, x_2 the second one, and x_3 the third one.

Generally, a system of linear equations can be compactly represented in their matrix form as $\boldsymbol{Ax} = \boldsymbol{b}$; see (2.3), and the product $\boldsymbol{Ax}$ is a (linear) combination of the columns of $\boldsymbol{A}$. We will discuss linear combinations in more detail in Section 2.5.

2.3 Solving Systems of Linear Equations

In (2.3), we introduced the general form of an equation system, i.e.,

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m \,, \end{aligned} \tag{2.37}$$

where $a_{ij} \in \mathbb{R}$ and $b_i \in \mathbb{R}$ are known constants and x_j are unknowns, $i = 1, \dots, m$, $j = 1, \dots, n$. Thus far, we saw that matrices can be used as a compact way of formulating systems of linear equations so that we can write $\boldsymbol{Ax} = \boldsymbol{b}$, see (2.10). Moreover, we defined basic matrix operations, such as addition and multiplication of matrices. In the following, we will focus on solving systems of linear equations and provide an algorithm for finding the inverse of a matrix.

2.3.1 Particular and General Solution

Before discussing how to generally solve systems of linear equations, let us have a look at an example. Consider the system of equations

$$\begin{bmatrix} 1 & 0 & 8 & -4 \\ 0 & 1 & 2 & 12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 42 \\ 8 \end{bmatrix} . \tag{2.38}$$

The system has two equations and four unknowns. Therefore, in general we would expect infinitely many solutions. This system of equations is in a particularly easy form, where the first two columns consist of a 1 and a 0. Remember that we want to find scalars $x_1, \dots, x_4$, such that $\sum_{i=1}^{4} x_i \boldsymbol{c}_i = \boldsymbol{b}$, where we define $\boldsymbol{c}_i$ to be the ith column of the matrix and $\boldsymbol{b}$ the right-hand-side of (2.38). A solution to the problem in (2.38) can be found immediately by taking 42 times the first column and 8 times the second column so that

$$\boldsymbol{b} = \begin{bmatrix} 42 \\ 8 \end{bmatrix} = 42 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 8 \begin{bmatrix} 0 \\ 1 \end{bmatrix} . \tag{2.39}$$

Therefore, a solution is $[42, 8, 0, 0]^\top$. This solution is called a *particular solution* or *special solution*. However, this is not the only solution of this system of linear equations. To capture all the other solutions, we need to be creative in generating $\mathbf{0}$ in a non-trivial way using the columns of the matrix: Adding $\mathbf{0}$ to our special solution does not change the special solution. To do so, we express the third column using the first two columns (which are of this very simple form)

particular solution
special solution

$$\begin{bmatrix} 8 \\ 2 \end{bmatrix} = 8 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.40}$$

so that $\mathbf{0} = 8\boldsymbol{c}_1 + 2\boldsymbol{c}_2 - 1\boldsymbol{c}_3 + 0\boldsymbol{c}_4$ and $(x_1, x_2, x_3, x_4) = (8, 2, -1, 0)$. In fact, any scaling of this solution by $\lambda_1 \in \mathbb{R}$ produces the $\mathbf{0}$ vector, i.e.,

$$\begin{bmatrix} 1 & 0 & 8 & -4 \\ 0 & 1 & 2 & 12 \end{bmatrix} \left(\lambda_1 \begin{bmatrix} 8 \\ 2 \\ -1 \\ 0 \end{bmatrix} \right) = \lambda_1 (8\boldsymbol{c}_1 + 2\boldsymbol{c}_2 - \boldsymbol{c}_3) = \mathbf{0}\,. \tag{2.41}$$

Following the same line of reasoning, we express the fourth column of the matrix in (2.38) using the first two columns and generate another set of non-trivial versions of $\mathbf{0}$ as

$$\begin{bmatrix} 1 & 0 & 8 & -4 \\ 0 & 1 & 2 & 12 \end{bmatrix} \left(\lambda_2 \begin{bmatrix} -4 \\ 12 \\ 0 \\ -1 \end{bmatrix} \right) = \lambda_2 (-4\boldsymbol{c}_1 + 12\boldsymbol{c}_2 - \boldsymbol{c}_4) = \mathbf{0} \tag{2.42}$$

general solution

for any $\lambda_2 \in \mathbb{R}$. Putting everything together, we obtain all solutions of the equation system in (2.38), which is called the *general solution*, as the set

$$\left\{ \boldsymbol{x} \in \mathbb{R}^4 : \boldsymbol{x} = \begin{bmatrix} 42 \\ 8 \\ 0 \\ 0 \end{bmatrix} + \lambda_1 \begin{bmatrix} 8 \\ 2 \\ -1 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} -4 \\ 12 \\ 0 \\ -1 \end{bmatrix}, \lambda_1, \lambda_2 \in \mathbb{R} \right\}. \tag{2.43}$$

Remark. The general approach we followed consisted of the following three steps:

1. Find a particular solution to $\boldsymbol{Ax} = \boldsymbol{b}$.
2. Find all solutions to $\boldsymbol{Ax} = \mathbf{0}$.
3. Combine the solutions from steps 1. and 2. to the general solution.

Neither the general nor the particular solution is unique. $\diamondsuit$

The system of linear equations in the preceding example was easy to solve because the matrix in (2.38) has this particularly convenient form, which allowed us to find the particular and the general solution by inspection. However, general equation systems are not of this simple form. Fortunately, there exists a constructive algorithmic way of transforming any system of linear equations into this particularly simple form: Gaussian elimination. Key to Gaussian elimination are elementary transformations of systems of linear equations, which transform the equation system into a simple form. Then, we can apply the three steps to the simple form that we just discussed in the context of the example in (2.38).

2.3.2 Elementary Transformations

elementary transformations

Key to solving a system of linear equations are *elementary transformations* that keep the solution set the same, but that transform the equation system into a simpler form:

- Exchange of two equations (rows in the matrix representing the system of equations)
- Multiplication of an equation (row) with a constant $\lambda \in \mathbb{R}\backslash\{0\}$
- Addition of two equations (rows)

Example 2.6

For $a \in \mathbb{R}$, we seek all solutions of the following system of equations:

$$\begin{array}{rcrcrcrcrcr} -2x_1 & + & 4x_2 & - & 2x_3 & - & x_4 & + & 4x_5 & = & -3 \\ 4x_1 & - & 8x_2 & + & 3x_3 & - & 3x_4 & + & x_5 & = & 2 \\ x_1 & - & 2x_2 & + & x_3 & - & x_4 & + & x_5 & = & 0 \\ x_1 & - & 2x_2 & & & - & 3x_4 & + & 4x_5 & = & a \end{array}. \tag{2.44}$$

We start by converting this system of equations into the compact matrix notation $\boldsymbol{Ax} = \boldsymbol{b}$. We no longer mention the variables $\boldsymbol{x}$ explicitly and build the *augmented matrix* (in the form $[\boldsymbol{A} \,|\, \boldsymbol{b}]$)

augmented matrix

$$\left[\begin{array}{rrrrr|r} -2 & 4 & -2 & -1 & 4 & -3 \\ 4 & -8 & 3 & -3 & 1 & 2 \\ 1 & -2 & 1 & -1 & 1 & 0 \\ 1 & -2 & 0 & -3 & 4 & a \end{array}\right] \begin{array}{l} \text{Swap with } R_3 \\ \\ \text{Swap with } R_1 \\ \\ \end{array}$$

where we used the vertical line to separate the left-hand side from the right-hand side in (2.44). We use $\rightsquigarrow$ to indicate a transformation of the augmented matrix using elementary transformations.

The augmented matrix $[\boldsymbol{A} \,|\, \boldsymbol{b}]$ compactly represents the system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$.

Swapping Rows 1 and 3 leads to

$$\left[\begin{array}{rrrrr|r} 1 & -2 & 1 & -1 & 1 & 0 \\ 4 & -8 & 3 & -3 & 1 & 2 \\ -2 & 4 & -2 & -1 & 4 & -3 \\ 1 & -2 & 0 & -3 & 4 & a \end{array}\right] \begin{array}{l} \\ -4R_1 \\ +2R_1 \\ -R_1 \end{array}$$

When we now apply the indicated transformations (e.g., subtract Row 1 four times from Row 2), we obtain

$$\left[\begin{array}{rrrrr|r} 1 & -2 & 1 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & -3 & 2 \\ 0 & 0 & 0 & -3 & 6 & -3 \\ 0 & 0 & -1 & -2 & 3 & a \end{array}\right] \begin{array}{l} \\ \\ \\ -R_2 - R_3 \end{array}$$

$$\rightsquigarrow \left[\begin{array}{rrrrr|r} 1 & -2 & 1 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & -3 & 2 \\ 0 & 0 & 0 & -3 & 6 & -3 \\ 0 & 0 & 0 & 0 & 0 & a+1 \end{array}\right] \begin{array}{l} \\ \cdot(-1) \\ \cdot(-\frac{1}{3}) \\ \\ \end{array}$$

$$\rightsquigarrow \left[\begin{array}{rrrrr|r} 1 & -2 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 3 & -2 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & a+1 \end{array}\right]$$

row-echelon form

This (augmented) matrix is in a convenient form, the *row-echelon form* (REF). Reverting this compact notation back into the explicit notation with the variables we seek, we obtain

$$\begin{array}{rcrcrcrcrcr} x_1 & - & 2x_2 & + & x_3 & - & x_4 & + & x_5 & = & 0 \\ & & & & x_3 & - & x_4 & + & 3x_5 & = & -2 \\ & & & & & & x_4 & - & 2x_5 & = & 1 \\ & & & & & & & & 0 & = & a+1 \end{array} . \tag{2.45}$$

particular solution

Only for $a = -1$ this system can be solved. A *particular solution* is

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} . \tag{2.46}$$

general solution

The *general solution*, which captures the set of all possible solutions, is

$$\left\{ \boldsymbol{x} \in \mathbb{R}^5 : \boldsymbol{x} = \begin{bmatrix} 2 \\ 0 \\ -1 \\ 1 \\ 0 \end{bmatrix} + \lambda_1 \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 2 \\ 0 \\ -1 \\ 2 \\ 1 \end{bmatrix} , \quad \lambda_1, \lambda_2 \in \mathbb{R} \right\} . \tag{2.47}$$

In the following, we will detail a constructive way to obtain a particular and general solution of a system of linear equations.

pivot

Remark (Pivots and Staircase Structure). The leading coefficient of a row (first nonzero number from the left) is called the *pivot* and is always strictly to the right of the pivot of the row above it. Therefore, any equation system in row-echelon form always has a "staircase" structure. ♢

row-echelon form

Definition 2.6 (Row-Echelon Form). A matrix is in *row-echelon form* if

- All rows that contain only zeros are at the bottom of the matrix; correspondingly, all rows that contain at least one nonzero element are on top of rows that contain only zeros.
- Looking at nonzero rows only, the first nonzero number from the left (also called the *pivot* or the *leading coefficient*) is always strictly to the right of the pivot of the row above it.

pivot
leading coefficient
In other texts, it is sometimes required that the pivot is 1.
basic variable
free variable

Remark (Basic and Free Variables). The variables corresponding to the pivots in the row-echelon form are called *basic variables* and the other variables are *free variables*. For example, in (2.45), x_1, x_3, x_4 are basic variables, whereas x_2, x_5 are free variables. ♢

Remark (Obtaining a Particular Solution). The row-echelon form makes our lives easier when we need to determine a particular solution. To do this, we express the right-hand side of the equation system using the pivot columns, such that $\boldsymbol{b} = \sum_{i=1}^{P} \lambda_i \boldsymbol{p}_i$, where $\boldsymbol{p}_i,\ i = 1, \ldots, P$, are the pivot columns. The λ_i are determined easiest if we start with the rightmost pivot column and work our way to the left.

In the previous example, we would try to find $\lambda_1, \lambda_2, \lambda_3$ so that

$$\lambda_1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \lambda_3 \begin{bmatrix} -1 \\ -1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \\ 1 \\ 0 \end{bmatrix} . \tag{2.48}$$

From here, we find relatively directly that $\lambda_3 = 1, \lambda_2 = -1, \lambda_1 = 2$. When we put everything together, we must not forget the non-pivot columns for which we set the coefficients implicitly to 0. Therefore, we get the particular solution $\boldsymbol{x} = [2, 0, -1, 1, 0]^\top$. ♢

Remark (Reduced Row Echelon Form). An equation system is in *reduced row-echelon form* (also: *row-reduced echelon form* or *row canonical form*) if

reduced row-echelon form

- It is in row-echelon form.
- Every pivot is 1.
- The pivot is the only nonzero entry in its column.

♢

The reduced row-echelon form will play an important role later in Section 2.3.3 because it allows us to determine the general solution of a system of linear equations in a straightforward way.

Gaussian elimination

Remark (Gaussian Elimination). *Gaussian elimination* is an algorithm that performs elementary transformations to bring a system of linear equations into reduced row-echelon form. ♢

Example 2.7 (Reduced Row Echelon Form)

Verify that the following matrix is in reduced row-echelon form (the pivots are in **bold**):

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{1} & 3 & 0 & 0 & 3 \\ 0 & 0 & \mathbf{1} & 0 & 9 \\ 0 & 0 & 0 & \mathbf{1} & -4 \end{bmatrix} . \tag{2.49}$$

The key idea for finding the solutions of $\boldsymbol{A}\boldsymbol{x} = \mathbf{0}$ is to look at the *non-pivot columns*, which we will need to express as a (linear) combination of the pivot columns. The reduced row echelon form makes this relatively straightforward, and we express the non-pivot columns in terms of sums and multiples of the pivot columns that are on their left: The second column is 3 times the first column (we can ignore the pivot columns on the right of the second column). Therefore, to obtain $\mathbf{0}$, we need to subtract the second column from three times the first column. Now, we look at the fifth column, which is our second non-pivot column. The fifth column can be expressed as 3 times the first pivot column, 9 times the second pivot column, and -4 times the third pivot column. We need to keep track of the indices of the pivot columns and translate this into 3 times the first column, 0 times the second column (which is a non-pivot column), 9 times the third column (which is our second pivot column), and -4 times the fourth column (which is the third pivot column). Then we need to subtract the fifth column to obtain **0**. In the end, we are still solving a homogeneous equation system.

To summarize, all solutions of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{x} \in \mathbb{R}^5$ are given by

$$\left\{ \boldsymbol{x} \in \mathbb{R}^5 : \boldsymbol{x} = \lambda_1 \begin{bmatrix} 3 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 3 \\ 0 \\ 9 \\ -4 \\ -1 \end{bmatrix}, \quad \lambda_1, \lambda_2 \in \mathbb{R} \right\}. \tag{2.50}$$

2.3.3 The Minus-1 Trick

In the following, we introduce a practical trick for reading out the solutions $\boldsymbol{x}$ of a homogeneous system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, where $\boldsymbol{A} \in \mathbb{R}^{k \times n}, \boldsymbol{x} \in \mathbb{R}^n$.

To start, we assume that $\boldsymbol{A}$ is in reduced row-echelon form without any rows that just contain zeros, i.e.,

$$\boldsymbol{A} = \begin{bmatrix} 0 & \cdots & 0 & \boldsymbol{1} & * & \cdots & * & 0 & * & \cdots & * & 0 & * & \cdots & * \\ \vdots & & \vdots & 0 & 0 & \cdots & 0 & \boldsymbol{1} & * & \cdots & * & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & 0 & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & 0 & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \boldsymbol{1} & * & \cdots & * \end{bmatrix}, \tag{2.51}$$

where $*$ can be an arbitrary real number, with the constraints that the first nonzero entry per row must be 1 and all other entries in the corresponding column must be 0. The columns $j_1, \ldots, j_k$ with the pivots (marked in **bold**) are the standard unit vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_k \in \mathbb{R}^k$. We extend this matrix to an $n \times n$-matrix $\tilde{\boldsymbol{A}}$ by adding $n - k$ rows of the form

$$\begin{bmatrix} 0 & \cdots & 0 & -1 & 0 & \cdots & 0 \end{bmatrix} \tag{2.52}$$

so that the diagonal of the augmented matrix $\tilde{\boldsymbol{A}}$ contains either 1 or -1. Then, the columns of $\tilde{\boldsymbol{A}}$ that contain the -1 as pivots are solutions of the homogeneous equation system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$. To be more precise, these columns form a basis (Section 2.6.1) of the solution space of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, which we will later call the *kernel* or *null space* (see Section 2.7.3).

kernel
null space

Example 2.8 (Minus-1 Trick)
Let us revisit the matrix in (2.49), which is already in reduced REF:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 3 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 9 \\ 0 & 0 & 0 & 1 & -4 \end{bmatrix}. \tag{2.53}$$

We now augment this matrix to a 5×5 matrix by adding rows of the form (2.52) at the places where the pivots on the diagonal are missing and obtain

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 1 & 3 & 0 & 0 & 3 \\ 0 & \boldsymbol{-1} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 9 \\ 0 & 0 & 0 & 1 & -4 \\ 0 & 0 & 0 & 0 & \boldsymbol{-1} \end{bmatrix}. \tag{2.54}$$

From this form, we can immediately read out the solutions of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ by taking the columns of $\tilde{\boldsymbol{A}}$, which contain -1 on the diagonal:

$$\left\{ \boldsymbol{x} \in \mathbb{R}^5 : \boldsymbol{x} = \lambda_1 \begin{bmatrix} 3 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 3 \\ 0 \\ 9 \\ -4 \\ -1 \end{bmatrix}, \quad \lambda_1, \lambda_2 \in \mathbb{R} \right\}, \tag{2.55}$$

which is identical to the solution in (2.50) that we obtained by "insight".

Calculating the Inverse

To compute the inverse $\boldsymbol{A}^{-1}$ of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, we need to find a matrix $\boldsymbol{X}$ that satisfies $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{I}_n$. Then, $\boldsymbol{X} = \boldsymbol{A}^{-1}$. We can write this down as a set of simultaneous linear equations $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{I}_n$, where we solve for $\boldsymbol{X} = [\boldsymbol{x}_1 | \cdots | \boldsymbol{x}_n]$. We use the augmented matrix notation for a compact representation of this set of systems of linear equations and obtain

$$[\boldsymbol{A}|\boldsymbol{I}_n] \quad \rightsquigarrow \cdots \rightsquigarrow \quad [\boldsymbol{I}_n|\boldsymbol{A}^{-1}]. \tag{2.56}$$

This means that if we bring the augmented equation system into reduced row-echelon form, we can read out the inverse on the right-hand side of the equation system. Hence, determining the inverse of a matrix is equivalent to solving systems of linear equations.

Example 2.9 (Calculating an Inverse Matrix by Gaussian Elimination)
To determine the inverse of

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{2.57}$$

we write down the augmented matrix

$$\left[\begin{array}{cccc|cccc} 1 & 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

and use Gaussian elimination to bring it into reduced row-echelon form

$$\left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & -1 & 2 & -2 & 2 \\ 0 & 1 & 0 & 0 & 1 & -1 & 2 & -2 \\ 0 & 0 & 1 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 & -1 & 2 \end{array}\right],$$

such that the desired inverse is given as its right-hand side:

$$\boldsymbol{A}^{-1} = \begin{bmatrix} -1 & 2 & -2 & 2 \\ 1 & -1 & 2 & -2 \\ 1 & -1 & 1 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}. \tag{2.58}$$

We can verify that (2.58) is indeed the inverse by performing the multiplication $\boldsymbol{A}\boldsymbol{A}^{-1}$ and observing that we recover $\boldsymbol{I}_4$.

2.3.4 Algorithms for Solving a System of Linear Equations

In the following, we briefly discuss approaches to solving a system of linear equations of the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. We make the assumption that a solution exists. Should there be no solution, we need to resort to approximate solutions, which we do not cover in this chapter. One way to solve the approximate problem is using the approach of linear regression, which we discuss in detail in Chapter 9.

In special cases, we may be able to determine the inverse $\boldsymbol{A}^{-1}$, such that the solution of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is given as $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$. However, this is only possible if $\boldsymbol{A}$ is a square matrix and invertible, which is often not the case. Otherwise, under mild assumptions (i.e., $\boldsymbol{A}$ needs to have linearly independent columns) we can use the transformation

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \iff \boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top\boldsymbol{b} \iff \boldsymbol{x} = (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{b} \tag{2.59}$$

and use the *Moore-Penrose pseudo-inverse* $(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top$ to determine the solution (2.59) that solves $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, which also corresponds to the minimum norm least-squares solution. A disadvantage of this approach is that it requires many computations for the matrix-matrix product and computing the inverse of $\boldsymbol{A}^\top\boldsymbol{A}$. Moreover, for reasons of numerical precision it is generally not recommended to compute the inverse or pseudo-inverse. In the following, we therefore briefly discuss alternative approaches to solving systems of linear equations.

Moore-Penrose pseudo-inverse

Gaussian elimination plays an important role when computing determinants (Section 4.1), checking whether a set of vectors is linearly independent (Section 2.5), computing the inverse of a matrix (Section 2.2.2), computing the rank of a matrix (Section 2.6.2), and determining a basis of a vector space (Section 2.6.1). Gaussian elimination is an intuitive and constructive way to solve a system of linear equations with thousands of variables. However, for systems with millions of variables, it is impractical as the required number of arithmetic operations scales cubically in the number of simultaneous equations.

In practice, systems of many linear equations are solved indirectly, by either stationary iterative methods, such as the Richardson method, the Jacobi method, the Gauß-Seidel method, and the successive over-relaxation method, or Krylov subspace methods, such as conjugate gradients, generalized minimal residual, or biconjugate gradients. We refer to the books by Stoer and Burlirsch (2002), Strang (2003), and Liesen and Mehrmann (2015) for further details.

Let $\boldsymbol{x}_*$ be a solution of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. The key idea of these iterative methods is to set up an iteration of the form

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{C}\boldsymbol{x}^{(k)} + \boldsymbol{d} \tag{2.60}$$

for suitable $\boldsymbol{C}$ and $\boldsymbol{d}$ that reduces the residual error $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}_*\|$ in every iteration and converges to $\boldsymbol{x}_*$. We will introduce norms $\|\cdot\|$, which allow us to compute similarities between vectors, in Section 3.1.

2.4 Vector Spaces

Thus far, we have looked at systems of linear equations and how to solve them (Section 2.3). We saw that systems of linear equations can be compactly represented using matrix-vector notation (2.10). In the following, we will have a closer look at vector spaces, i.e., a structured space in which vectors live.

In the beginning of this chapter, we informally characterized vectors as objects that can be added together and multiplied by a scalar, and they remain objects of the same type. Now, we are ready to formalize this, and we will start by introducing the concept of a group, which is a set of elements and an operation defined on these elements that keeps some structure of the set intact.

2.4.1 Groups

Groups play an important role in computer science. Besides providing a fundamental framework for operations on sets, they are heavily used in cryptography, coding theory, and graphics.

group

Definition 2.7 (Group)**.** Consider a set $\mathcal{G}$ and an operation $\otimes : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ defined on $\mathcal{G}$. Then $G := (\mathcal{G}, \otimes)$ is called a *group* if the following hold:

closure
associativity
neutral element
inverse element

1. *Closure* of $\mathcal{G}$ under $\otimes$: $\forall x, y \in \mathcal{G} : x \otimes y \in \mathcal{G}$
2. *Associativity:* $\forall x, y, z \in \mathcal{G} : (x \otimes y) \otimes z = x \otimes (y \otimes z)$
3. *Neutral element:* $\exists e \in \mathcal{G}\, \forall x \in \mathcal{G} : x \otimes e = x$ and $e \otimes x = x$
4. *Inverse element:* $\forall x \in \mathcal{G}\, \exists y \in \mathcal{G} : x \otimes y = e$ and $y \otimes x = e$, where e is the neutral element. We often write x^{-1} to denote the inverse element of x.

Remark. The inverse element is defined with respect to the operation $\otimes$ and does not necessarily mean $\frac{1}{x}$. $\diamondsuit$

Abelian group

If additionally $\forall x, y \in \mathcal{G} : x \otimes y = y \otimes x$, then $G = (\mathcal{G}, \otimes)$ is an *Abelian group* (commutative).

Example 2.10 (Groups)
Let us have a look at some examples of sets with associated operations and see whether they are groups:

$\mathbb{N}_0 := \mathbb{N} \cup \{0\}$

- $(\mathbb{Z}, +)$ is an Abelian group.
- $(\mathbb{N}_0, +)$ is not a group: Although $(\mathbb{N}_0, +)$ possesses a neutral element (0), the inverse elements are missing.
- $(\mathbb{Z}, \cdot)$ is not a group: Although $(\mathbb{Z}, \cdot)$ contains a neutral element (1), the inverse elements for any $z \in \mathbb{Z}, z \neq \pm 1$, are missing.
- $(\mathbb{R}, \cdot)$ is not a group since 0 does not possess an inverse element.
- $(\mathbb{R}\backslash\{0\}, \cdot)$ is Abelian.
- $(\mathbb{R}^n, +), (\mathbb{Z}^n, +), n \in \mathbb{N}$ are Abelian if $+$ is defined componentwise, i.e.,
$$(x_1, \cdots, x_n) + (y_1, \cdots, y_n) = (x_1 + y_1, \cdots, x_n + y_n). \tag{2.61}$$
Then, $(x_1, \cdots, x_n)^{-1} := (-x_1, \cdots, -x_n)$ is the inverse element and $e = (0, \cdots, 0)$ is the neutral element.
- $(\mathbb{R}^{m \times n}, +)$, the set of $m \times n$-matrices is Abelian (with componentwise addition as defined in (2.61)).
- Let us have a closer look at $(\mathbb{R}^{n \times n}, \cdot)$, i.e., the set of $n \times n$-matrices with matrix multiplication as defined in (2.13).
  - Closure and associativity follow directly from the definition of matrix multiplication.
  - Neutral element: The identity matrix $\boldsymbol{I}_n$ is the neutral element with respect to matrix multiplication "$\cdot$" in $(\mathbb{R}^{n \times n}, \cdot)$.

– Inverse element: If the inverse exists ($\boldsymbol{A}$ is regular), then $\boldsymbol{A}^{-1}$ is the inverse element of $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, and in exactly this case $(\mathbb{R}^{n\times n}, \cdot)$ is a group, called the *general linear group*.

Definition 2.8 (General Linear Group)**.** The set of regular (invertible) matrices $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is a group with respect to matrix multiplication as defined in (2.13) and is called *general linear group* $GL(n, \mathbb{R})$. However, since matrix multiplication is not commutative, the group is not Abelian.

general linear group

2.4.2 Vector Spaces

When we discussed groups, we looked at sets $\mathcal{G}$ and inner operations on $\mathcal{G}$, i.e., mappings $\mathcal{G} \times \mathcal{G} \to \mathcal{G}$ that only operate on elements in $\mathcal{G}$. In the following, we will consider sets that in addition to an inner operation $+$ also contain an outer operation $\cdot$, the multiplication of a vector $\boldsymbol{x} \in \mathcal{G}$ by a scalar $\lambda \in \mathbb{R}$. We can think of the inner operation as a form of addition, and the outer operation as a form of scaling. Note that the inner/outer operations have nothing to do with inner/outer products.

Definition 2.9 (Vector Space)**.** A real-valued *vector space* $V = (\mathcal{V}, +, \cdot)$ is a set $\mathcal{V}$ with two operations

vector space

$$+ : \mathcal{V} \times \mathcal{V} \to \mathcal{V} \tag{2.62}$$

$$\cdot : \mathbb{R} \times \mathcal{V} \to \mathcal{V} \tag{2.63}$$

where

1. $(\mathcal{V}, +)$ is an Abelian group
2. Distributivity:
   1. $\forall \lambda \in \mathbb{R}, \boldsymbol{x}, \boldsymbol{y} \in \mathcal{V} : \lambda \cdot (\boldsymbol{x} + \boldsymbol{y}) = \lambda \cdot \boldsymbol{x} + \lambda \cdot \boldsymbol{y}$
   2. $\forall \lambda, \psi \in \mathbb{R}, \boldsymbol{x} \in \mathcal{V} : (\lambda + \psi) \cdot \boldsymbol{x} = \lambda \cdot \boldsymbol{x} + \psi \cdot \boldsymbol{x}$
3. Associativity (outer operation): $\forall \lambda, \psi \in \mathbb{R}, \boldsymbol{x} \in \mathcal{V} : \lambda \cdot (\psi \cdot \boldsymbol{x}) = (\lambda\psi) \cdot \boldsymbol{x}$
4. Neutral element with respect to the outer operation: $\forall \boldsymbol{x} \in \mathcal{V} : 1 \cdot \boldsymbol{x} = \boldsymbol{x}$

The elements $\boldsymbol{x} \in V$ are called *vectors*. The neutral element of $(\mathcal{V}, +)$ is the zero vector $\boldsymbol{0} = [0, \ldots, 0]^\top$, and the inner operation $+$ is called *vector addition*. The elements $\lambda \in \mathbb{R}$ are called *scalars* and the outer operation $\cdot$ is a *multiplication by scalars*. Note that a scalar product is something different, and we will get to this in Section 3.2.

vector
vector addition
scalar
multiplication by scalars

Remark. A "vector multiplication" $\boldsymbol{ab}$, $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, is not defined. Theoretically, we could define an element-wise multiplication, such that $\boldsymbol{c} = \boldsymbol{ab}$ with $c_j = a_j b_j$. This "array multiplication" is common to many programming languages but makes mathematically limited sense using the standard rules for matrix multiplication: By treating vectors as $n \times 1$ matrices (which we usually do), we can use the matrix multiplication as defined in (2.13). However, then the dimensions of the vectors do not match. Only the following multiplications for vectors are defined: $\boldsymbol{a}\boldsymbol{b}^\top \in \mathbb{R}^{n\times n}$ (*outer product*), $\boldsymbol{a}^\top\boldsymbol{b} \in \mathbb{R}$ (inner/scalar/dot product). $\diamondsuit$

outer product

Example 2.11 (Vector Spaces)
Let us have a look at some important examples:

- $\mathcal{V} = \mathbb{R}^n, n \in \mathbb{N}$ is a vector space with operations defined as follows:
  - Addition: $\boldsymbol{x}+\boldsymbol{y} = (x_1,\ldots,x_n)+(y_1,\ldots,y_n) = (x_1+y_1,\ldots,x_n+y_n)$ for all $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n$
  - Multiplication by scalars: $\lambda\boldsymbol{x} = \lambda(x_1,\ldots,x_n) = (\lambda x_1,\ldots,\lambda x_n)$ for all $\lambda \in \mathbb{R}, \boldsymbol{x} \in \mathbb{R}^n$
- $\mathcal{V} = \mathbb{R}^{m\times n}, m,n \in \mathbb{N}$ is a vector space with
  - Addition: $\boldsymbol{A}+\boldsymbol{B} = \begin{bmatrix} a_{11}+b_{11} & \cdots & a_{1n}+b_{1n} \\ \vdots & & \vdots \\ a_{m1}+b_{m1} & \cdots & a_{mn}+b_{mn} \end{bmatrix}$ is defined elementwise for all $\boldsymbol{A},\boldsymbol{B} \in \mathcal{V}$
  - Multiplication by scalars: $\lambda\boldsymbol{A} = \begin{bmatrix} \lambda a_{11} & \cdots & \lambda a_{1n} \\ \vdots & & \vdots \\ \lambda a_{m1} & \cdots & \lambda a_{mn} \end{bmatrix}$ as defined in Section 2.2. Remember that $\mathbb{R}^{m\times n}$ is equivalent to $\mathbb{R}^{mn}$.
- $\mathcal{V} = \mathbb{C}$, with the standard definition of addition of complex numbers.

Remark. In the following, we will denote a vector space $(\mathcal{V},+,\cdot)$ by V when $+$ and $\cdot$ are the standard vector addition and scalar multiplication. Moreover, we will use the notation $\boldsymbol{x} \in V$ for vectors in $\mathcal{V}$ to simplify notation. $\diamondsuit$

Remark. The vector spaces $\mathbb{R}^n, \mathbb{R}^{n\times 1}, \mathbb{R}^{1\times n}$ are only different in the way we write vectors. In the following, we will not make a distinction between $\mathbb{R}^n$ and $\mathbb{R}^{n\times 1}$, which allows us to write n-tuples as *column vectors*

column vector

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}. \tag{2.64}$$

This simplifies the notation regarding vector space operations. However, we do distinguish between $\mathbb{R}^{n\times 1}$ and $\mathbb{R}^{1\times n}$ (the *row vectors*) to avoid confusion with matrix multiplication. By default, we write $\boldsymbol{x}$ to denote a column vector, and a row vector is denoted by $\boldsymbol{x}^\top$, the *transpose* of $\boldsymbol{x}$. $\diamondsuit$

row vector

transpose

2.4.3 Vector Subspaces

In the following, we will introduce vector subspaces. Intuitively, they are sets contained in the original vector space with the property that when we perform vector space operations on elements within this subspace, we will never leave it. In this sense, they are "closed". Vector subspaces are a key idea in machine learning. For example, Chapter 10 demonstrates how to use vector subspaces for dimensionality reduction.

Definition 2.10 (Vector Subspace). Let $V = (\mathcal{V}, +, \cdot)$ be a vector space and $\mathcal{U} \subseteq \mathcal{V}$, $\mathcal{U} \neq \emptyset$. Then $U = (\mathcal{U}, +, \cdot)$ is called *vector subspace* of V (or *linear subspace*) if U is a vector space with the vector space operations $+$ and $\cdot$ restricted to $\mathcal{U} \times \mathcal{U}$ and $\mathbb{R} \times \mathcal{U}$. We write $U \subseteq V$ to denote a subspace U of V.

vector subspace
linear subspace

If $\mathcal{U} \subseteq \mathcal{V}$ and V is a vector space, then U naturally inherits many properties directly from V because they hold for all $\boldsymbol{x} \in \mathcal{V}$, and in particular for all $\boldsymbol{x} \in \mathcal{U} \subseteq \mathcal{V}$. This includes the Abelian group properties, the distributivity, the associativity and the neutral element. To determine whether $(\mathcal{U}, +, \cdot)$ is a subspace of V we still do need to show

1. $\mathcal{U} \neq \emptyset$, in particular: $\boldsymbol{0} \in \mathcal{U}$
2. Closure of U:
   a. With respect to the outer operation: $\forall \lambda \in \mathbb{R} \, \forall \boldsymbol{x} \in \mathcal{U} : \lambda \boldsymbol{x} \in \mathcal{U}$.
   b. With respect to the inner operation: $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{U} : \boldsymbol{x} + \boldsymbol{y} \in \mathcal{U}$.

Example 2.12 (Vector Subspaces)
Let us have a look at some examples:

- For every vector space V, the trivial subspaces are V itself and $\{\boldsymbol{0}\}$.
- Only example D in Figure 2.1 is a subspace of $\mathbb{R}^2$ (with the usual inner/outer operations). In A and C, the closure property is violated; B does not contain $\boldsymbol{0}$.
- The solution set of a homogeneous system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ with n unknowns $\boldsymbol{x} = [x_1, \ldots, x_n]^\top$ is a subspace of $\mathbb{R}^n$.
- The solution of an inhomogeneous system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{b} \neq \boldsymbol{0}$ is not a subspace of $\mathbb{R}^n$.
- The intersection of arbitrarily many subspaces is a subspace itself.

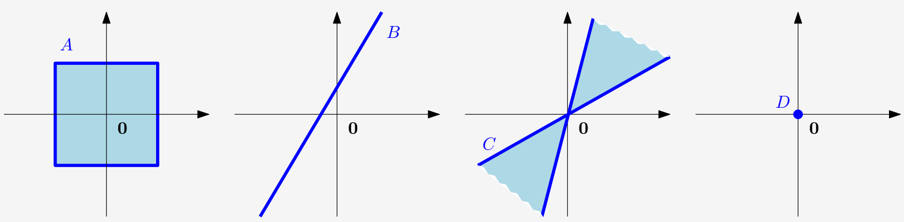


Figure 2.1 Not all subsets of $\mathbb{R}^2$ are subspaces. In A and C, the closure property is violated; B does not contain $\boldsymbol{0}$. Only D is a subspace.

Remark. Every subspace $U \subseteq (\mathbb{R}^n, +, \cdot)$ is the solution space of a homogeneous system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ for $\boldsymbol{x} \in \mathbb{R}^n$. $\diamondsuit$

2.5 Linear Independence

In the following, we will have a close look at what we can do with vectors (elements of the vector space). In particular, we can add vectors together and multiply them with scalars. The closure property guarantees that we end up with another vector in the same vector space. It is possible to find a set of vectors with which we can represent every vector in the vector space by adding them together and scaling them. This set of vectors is a *basis*, and we will discuss them in Section 2.6.1. Before we get there, we will need to introduce the concepts of linear combinations and linear independence.

Definition 2.11 (Linear Combination)**.** Consider a vector space V and a finite number of vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in V$. Then, every $\boldsymbol{v} \in V$ of the form

$$\boldsymbol{v} = \lambda_1 \boldsymbol{x}_1 + \cdots + \lambda_k \boldsymbol{x}_k = \sum_{i=1}^{k} \lambda_i \boldsymbol{x}_i \in V \tag{2.65}$$

linear combination

with $\lambda_1, \ldots, \lambda_k \in \mathbb{R}$ is a *linear combination* of the vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$.

The $\boldsymbol{0}$-vector can always be written as the linear combination of k vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ because $\boldsymbol{0} = \sum_{i=1}^{k} 0\boldsymbol{x}_i$ is always true. In the following, we are interested in non-trivial linear combinations of a set of vectors to represent $\boldsymbol{0}$, i.e., linear combinations of vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$, where not all coefficients λ_i in (2.65) are 0.

linearly dependent

linearly independent

Definition 2.12 (Linear (In)dependence)**.** Let us consider a vector space V with $k \in \mathbb{N}$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in V$. If there is a non-trivial linear combination, such that $\boldsymbol{0} = \sum_{i=1}^{k} \lambda_i \boldsymbol{x}_i$ with at least one $\lambda_i \neq 0$, the vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ are *linearly dependent*. If only the trivial solution exists, i.e., $\lambda_1 = \ldots = \lambda_k = 0$ the vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ are *linearly independent*.

Linear independence is one of the most important concepts in linear algebra. Intuitively, a set of linearly independent vectors consists of vectors that have no redundancy, i.e., if we remove any of those vectors from the set, we will lose something. Throughout the next sections, we will formalize this intuition more.

Example 2.13 (Linearly Dependent Vectors)

A geographic example may help to clarify the concept of linear independence. A person in Nairobi (Kenya) describing where Kigali (Rwanda) is might say ,"You can get to Kigali by first going $506\,\mathrm{km}$ Northwest to Kampala (Uganda) and then $374\,\mathrm{km}$ Southwest.". This is sufficient information to describe the location of Kigali because the geographic coordinate system may be considered a two-dimensional vector space (ignoring altitude and the Earth's curved surface). The person may add, "It is about $751\,\mathrm{km}$ West of here." Although this last statement is true, it is not necessary to find Kigali given the previous information (see Figure 2.2 for an illustration). In this example, the "$506\,\mathrm{km}$ Northwest" vector (blue) and the "$374\,\mathrm{km}$ Southwest" vector (purple) are linearly independent. This means the Southwest vector cannot be described in terms of the Northwest vector, and vice versa. However, the third "$751\,\mathrm{km}$ West" vector (black) is a linear combination of the other two vectors, and it makes the set of vectors linearly dependent. Equivalently, given "$751\,\mathrm{km}$ West" and "$374\,\mathrm{km}$ Southwest" can be linearly combined to obtain "$506\,\mathrm{km}$ Northwest".

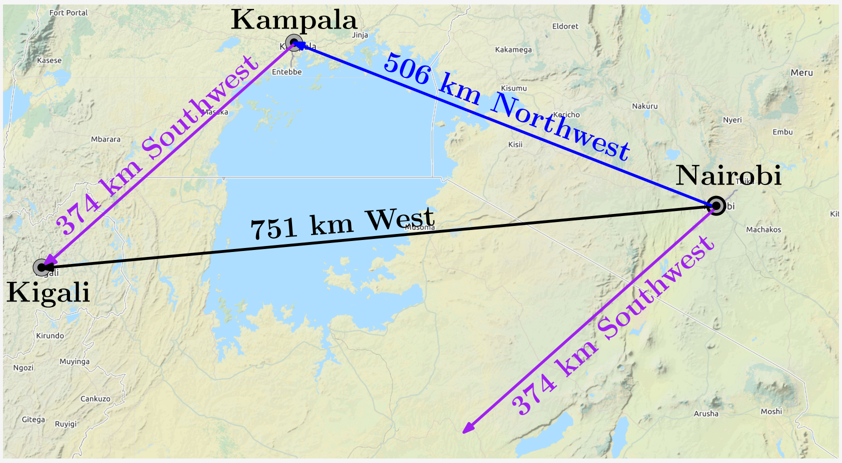


Figure 2.2 Geographic example (with crude approximations to cardinal directions) of linearly dependent vectors in a two-dimensional space (plane).

Remark. The following properties are useful to find out whether vectors are linearly independent:

- k vectors are either linearly dependent or linearly independent. There is no third option.
- If at least one of the vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ is $\boldsymbol{0}$ then they are linearly dependent. The same holds if two vectors are identical.
- The vectors $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_k : \boldsymbol{x}_i \neq \boldsymbol{0}, i = 1, \dots, k\}$, $k \geqslant 2$, are linearly dependent if and only if (at least) one of them is a linear combination of the others. In particular, if one vector is a multiple of another vector, i.e., $\boldsymbol{x}_i = \lambda \boldsymbol{x}_j$, $\lambda \in \mathbb{R}$ then the set $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_k : \boldsymbol{x}_i \neq \boldsymbol{0}, i = 1, \dots, k\}$ is linearly dependent.
- A practical way of checking whether vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k \in V$ are linearly independent is to use Gaussian elimination: Write all vectors as columns of a matrix $\boldsymbol{A}$ and perform Gaussian elimination until the matrix is in row echelon form (the reduced row-echelon form is unnecessary here):

- The pivot columns indicate the vectors, which are linearly independent of the vectors on the left. Note that there is an ordering of vectors when the matrix is built.
- The non-pivot columns can be expressed as linear combinations of the pivot columns on their left. For instance, the row-echelon form

$$\begin{bmatrix}1 & 3 & 0\\0 & 0 & 2\end{bmatrix} \tag{2.66}$$

tells us that the first and third columns are pivot columns. The second column is a non-pivot column because it is three times the first column.

All column vectors are linearly independent if and only if all columns are pivot columns. If there is at least one non-pivot column, the columns (and, therefore, the corresponding vectors) are linearly dependent.

$\diamondsuit$

Example 2.14

Consider $\mathbb{R}^4$ with

$$\boldsymbol{x}_1 = \begin{bmatrix}1\\2\\-3\\4\end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix}1\\1\\0\\2\end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix}-1\\-2\\1\\1\end{bmatrix}. \tag{2.67}$$

To check whether they are linearly dependent, we follow the general approach and solve

$$\lambda_1\boldsymbol{x}_1 + \lambda_2\boldsymbol{x}_2 + \lambda_3\boldsymbol{x}_3 = \lambda_1\begin{bmatrix}1\\2\\-3\\4\end{bmatrix} + \lambda_2\begin{bmatrix}1\\1\\0\\2\end{bmatrix} + \lambda_3\begin{bmatrix}-1\\-2\\1\\1\end{bmatrix} = \boldsymbol{0} \tag{2.68}$$

for $\lambda_1, \ldots, \lambda_3$. We write the vectors $\boldsymbol{x}_i$, $i = 1, 2, 3$, as the columns of a matrix and apply elementary row operations until we identify the pivot columns:

$$\begin{bmatrix}1 & 1 & -1\\2 & 1 & -2\\-3 & 0 & 1\\4 & 2 & 1\end{bmatrix} \rightsquigarrow \cdots \rightsquigarrow \begin{bmatrix}1 & 1 & -1\\0 & 1 & 0\\0 & 0 & 1\\0 & 0 & 0\end{bmatrix}. \tag{2.69}$$

Here, every column of the matrix is a pivot column. Therefore, there is no non-trivial solution, and we require $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0$ to solve the equation system. Hence, the vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ are linearly independent.

Remark. Consider a vector space V with k linearly independent vectors $\boldsymbol{b}_1, \dots, \boldsymbol{b}_k$ and m linear combinations

$$\begin{aligned}\boldsymbol{x}_1 &= \sum_{i=1}^{k} \lambda_{i1} \boldsymbol{b}_i \,, \\ &\vdots \\ \boldsymbol{x}_m &= \sum_{i=1}^{k} \lambda_{im} \boldsymbol{b}_i \,.\end{aligned} \tag{2.70}$$

Defining $\boldsymbol{B} = [\boldsymbol{b}_1, \dots, \boldsymbol{b}_k]$ as the matrix whose columns are the linearly independent vectors $\boldsymbol{b}_1, \dots, \boldsymbol{b}_k$, we can write

$$\boldsymbol{x}_j = \boldsymbol{B}\boldsymbol{\lambda}_j \,, \quad \boldsymbol{\lambda}_j = \begin{bmatrix} \lambda_{1j} \\ \vdots \\ \lambda_{kj} \end{bmatrix} , \quad j = 1, \dots, m \,, \tag{2.71}$$

in a more compact form.

We want to test whether $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$ are linearly independent. For this purpose, we follow the general approach of testing when $\sum_{j=1}^{m} \psi_j \boldsymbol{x}_j = \boldsymbol{0}$. With (2.71), we obtain

$$\sum_{j=1}^{m} \psi_j \boldsymbol{x}_j = \sum_{j=1}^{m} \psi_j \boldsymbol{B}\boldsymbol{\lambda}_j = \boldsymbol{B} \sum_{j=1}^{m} \psi_j \boldsymbol{\lambda}_j \,. \tag{2.72}$$

This means that $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_m\}$ are linearly independent if and only if the column vectors $\{\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_m\}$ are linearly independent.

$\diamondsuit$

Remark. In a vector space V, m linear combinations of k vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ are linearly dependent if $m > k$. $\diamondsuit$

Example 2.15

Consider a set of linearly independent vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \boldsymbol{b}_4 \in \mathbb{R}^n$ and

$$\begin{array}{rcrcrcrcr} \boldsymbol{x}_1 &=& \boldsymbol{b}_1 &-& 2\boldsymbol{b}_2 &+& \boldsymbol{b}_3 &-& \boldsymbol{b}_4 \\ \boldsymbol{x}_2 &=& -4\boldsymbol{b}_1 &-& 2\boldsymbol{b}_2 && &+& 4\boldsymbol{b}_4 \\ \boldsymbol{x}_3 &=& 2\boldsymbol{b}_1 &+& 3\boldsymbol{b}_2 &-& \boldsymbol{b}_3 &-& 3\boldsymbol{b}_4 \\ \boldsymbol{x}_4 &=& 17\boldsymbol{b}_1 &-& 10\boldsymbol{b}_2 &+& 11\boldsymbol{b}_3 &+& \boldsymbol{b}_4 \end{array} . \tag{2.73}$$

Are the vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_4 \in \mathbb{R}^n$ linearly independent? To answer this question, we investigate whether the column vectors

$$\left\{ \begin{bmatrix} 1 \\ -2 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -4 \\ -2 \\ 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \\ -1 \\ -3 \end{bmatrix}, \begin{bmatrix} 17 \\ -10 \\ 11 \\ 1 \end{bmatrix} \right\} \tag{2.74}$$

are linearly independent. The reduced row-echelon form of the corresponding linear equation system with coefficient matrix

$$\boldsymbol{A} = \begin{bmatrix} 1 & -4 & 2 & 17 \\ -2 & -2 & 3 & -10 \\ 1 & 0 & -1 & 11 \\ -1 & 4 & -3 & 1 \end{bmatrix} \tag{2.75}$$

is given as

$$\begin{bmatrix} 1 & 0 & 0 & -7 \\ 0 & 1 & 0 & -15 \\ 0 & 0 & 1 & -18 \\ 0 & 0 & 0 & 0 \end{bmatrix} . \tag{2.76}$$

We see that the corresponding linear equation system is non-trivially solvable: The last column is not a pivot column, and $\boldsymbol{x}_4 = -7\boldsymbol{x}_1 - 15\boldsymbol{x}_2 - 18\boldsymbol{x}_3$. Therefore, $\boldsymbol{x}_1, \dots, \boldsymbol{x}_4$ are linearly dependent as $\boldsymbol{x}_4$ can be expressed as a linear combination of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_3$.

2.6 Basis and Rank

In a vector space V, we are particularly interested in sets of vectors $\mathcal{A}$ that possess the property that any vector $\boldsymbol{v} \in V$ can be obtained by a linear combination of vectors in $\mathcal{A}$. These vectors are special vectors, and in the following, we will characterize them.

2.6.1 Generating Set and Basis

Definition 2.13 (Generating Set and Span)**.** Consider a vector space $V = (\mathcal{V}, +, \cdot)$ and set of vectors $\mathcal{A} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_k\} \subseteq \mathcal{V}$. If every vector $\boldsymbol{v} \in \mathcal{V}$ can be expressed as a linear combination of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$, $\mathcal{A}$ is called a *generating set* of V. The set of all linear combinations of vectors in $\mathcal{A}$ is called the *span* of $\mathcal{A}$. If $\mathcal{A}$ spans the vector space V, we write $V = \mathrm{span}[\mathcal{A}]$ or $V = \mathrm{span}[\boldsymbol{x}_1, \dots, \boldsymbol{x}_k]$.

generating set

span

Generating sets are sets of vectors that span vector (sub)spaces, i.e., every vector can be represented as a linear combination of the vectors in the generating set. Now, we will be more specific and characterize the smallest generating set that spans a vector (sub)space.

Definition 2.14 (Basis)**.** Consider a vector space $V = (\mathcal{V}, +, \cdot)$ and $\mathcal{A} \subseteq \mathcal{V}$. A generating set $\mathcal{A}$ of V is called *minimal* if there exists no smaller set $\tilde{\mathcal{A}} \subsetneq \mathcal{A} \subseteq \mathcal{V}$ that spans V. Every linearly independent generating set of V is minimal and is called a *basis* of V.

minimal

basis

Let $V = (\mathcal{V}, +, \cdot)$ be a vector space and $\mathcal{B} \subseteq \mathcal{V}, \mathcal{B} \neq \emptyset$. Then, the following statements are equivalent:

A basis is a minimal generating set and a maximal linearly independent set of vectors.

- $\mathcal{B}$ is a basis of V.
- $\mathcal{B}$ is a minimal generating set.
- $\mathcal{B}$ is a maximal linearly independent set of vectors in V, i.e., adding any other vector to this set will make it linearly dependent.
- Every vector $\boldsymbol{x} \in V$ is a linear combination of vectors from $\mathcal{B}$, and every linear combination is unique, i.e., with

$$\boldsymbol{x} = \sum_{i=1}^{k} \lambda_i \boldsymbol{b}_i = \sum_{i=1}^{k} \psi_i \boldsymbol{b}_i \tag{2.77}$$

  and $\lambda_i, \psi_i \in \mathbb{R}$, $\boldsymbol{b}_i \in \mathcal{B}$ it follows that $\lambda_i = \psi_i,\ i = 1, \ldots, k$.

Example 2.16

- In $\mathbb{R}^3$, the *canonical/standard basis* is

canonical basis

$$\mathcal{B} = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{2.78}$$

- Different bases in $\mathbb{R}^3$ are

$$\mathcal{B}_1 = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}, \mathcal{B}_2 = \left\{ \begin{bmatrix} 0.5 \\ 0.8 \\ 0.4 \end{bmatrix}, \begin{bmatrix} 1.8 \\ 0.3 \\ 0.3 \end{bmatrix}, \begin{bmatrix} -2.2 \\ -1.3 \\ 3.5 \end{bmatrix} \right\}. \tag{2.79}$$

- The set

$$\mathcal{A} = \left\{ \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ -1 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ -4 \end{bmatrix} \right\} \tag{2.80}$$

  is linearly independent, but not a generating set (and no basis) of $\mathbb{R}^4$: For instance, the vector $[1, 0, 0, 0]^\top$ cannot be obtained by a linear combination of elements in $\mathcal{A}$.

Remark. Every vector space V possesses a basis $\mathcal{B}$. The preceding examples show that there can be many bases of a vector space V, i.e., there is no unique basis. However, all bases possess the same number of elements, the *basis vectors*. $\diamondsuit$

basis vector

We only consider finite-dimensional vector spaces V. In this case, the *dimension* of V is the number of basis vectors of V, and we write $\dim(V)$. If $U \subseteq V$ is a subspace of V, then $\dim(U) \leqslant \dim(V)$ and $\dim(U) =$

dimension

$\dim(V)$ if and only if $U = V$. Intuitively, the dimension of a vector space can be thought of as the number of independent directions in this vector space.

The dimension of a vector space corresponds to the number of its basis vectors.

Remark. The dimension of a vector space is not necessarily the number of elements in a vector. For instance, the vector space $V = \text{span}[\begin{bmatrix} 0 \\ 1 \end{bmatrix}]$ is one-dimensional, although the basis vector possesses two elements. $\diamondsuit$

Remark. A basis of a subspace $U = \text{span}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m] \subseteq \mathbb{R}^n$ can be found by executing the following steps:

1. Write the spanning vectors as columns of a matrix $\boldsymbol{A}$
2. Determine the row-echelon form of $\boldsymbol{A}$.
3. The spanning vectors associated with the pivot columns are a basis of U.

$\diamondsuit$

Example 2.17 (Determining a Basis)
For a vector subspace $U \subseteq \mathbb{R}^5$, spanned by the vectors

$$\boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 2 \\ -1 \\ -1 \\ -1 \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} 2 \\ -1 \\ 1 \\ 2 \\ -2 \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} 3 \\ -4 \\ 3 \\ 5 \\ -3 \end{bmatrix}, \quad \boldsymbol{x}_4 = \begin{bmatrix} -1 \\ 8 \\ -5 \\ -6 \\ 1 \end{bmatrix} \in \mathbb{R}^5, \tag{2.81}$$

we are interested in finding out which vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_4$ are a basis for U. For this, we need to check whether $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_4$ are linearly independent. Therefore, we need to solve

$$\sum_{i=1}^{4} \lambda_i \boldsymbol{x}_i = \boldsymbol{0}, \tag{2.82}$$

which leads to a homogeneous system of equations with matrix

$$\begin{bmatrix} \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & -1 \\ 2 & -1 & -4 & 8 \\ -1 & 1 & 3 & -5 \\ -1 & 2 & 5 & -6 \\ -1 & -2 & -3 & 1 \end{bmatrix}. \tag{2.83}$$

With the basic transformation rules for systems of linear equations, we obtain the row-echelon form

$$\begin{bmatrix} 1 & 2 & 3 & -1 \\ 2 & -1 & -4 & 8 \\ -1 & 1 & 3 & -5 \\ -1 & 2 & 5 & -6 \\ -1 & -2 & -3 & 1 \end{bmatrix} \rightsquigarrow \cdots \rightsquigarrow \begin{bmatrix} 1 & 2 & 3 & -1 \\ 0 & 1 & 2 & -2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Since the pivot columns indicate which set of vectors is linearly independent, we see from the row-echelon form that $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4$ are linearly independent (because the system of linear equations $\lambda_1 \boldsymbol{x}_1 + \lambda_2 \boldsymbol{x}_2 + \lambda_4 \boldsymbol{x}_4 = \boldsymbol{0}$ can only be solved with $\lambda_1 = \lambda_2 = \lambda_4 = 0$). Therefore, $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4\}$ is a basis of U.

2.6.2 Rank

The number of linearly independent columns of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ equals the number of linearly independent rows and is called the *rank* of $\boldsymbol{A}$ and is denoted by $\mathrm{rk}(\boldsymbol{A})$. rank

Remark. The rank of a matrix has some important properties:

- $\mathrm{rk}(\boldsymbol{A}) = \mathrm{rk}(\boldsymbol{A}^\top)$, i.e., the column rank equals the row rank.
- The columns of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ span a subspace $U \subseteq \mathbb{R}^m$ with $\dim(U) = \mathrm{rk}(\boldsymbol{A})$. Later we will call this subspace the *image* or *range*. A basis of U can be found by applying Gaussian elimination to $\boldsymbol{A}$ to identify the pivot columns.
- The rows of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ span a subspace $W \subseteq \mathbb{R}^n$ with $\dim(W) = \mathrm{rk}(\boldsymbol{A})$. A basis of W can be found by applying Gaussian elimination to $\boldsymbol{A}^\top$.
- For all $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ it holds that $\boldsymbol{A}$ is regular (invertible) if and only if $\mathrm{rk}(\boldsymbol{A}) = n$.
- For all $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and all $\boldsymbol{b} \in \mathbb{R}^m$ it holds that the linear equation system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ can be solved if and only if $\mathrm{rk}(\boldsymbol{A}) = \mathrm{rk}(\boldsymbol{A}|\boldsymbol{b})$, where $\boldsymbol{A}|\boldsymbol{b}$ denotes the augmented system.
- For $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ the subspace of solutions for $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ possesses dimension $n - \mathrm{rk}(\boldsymbol{A})$. Later, we will call this subspace the *kernel* or the *null space*. kernel null space
- A matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ has *full rank* if its rank equals the largest possible rank for a matrix of the same dimensions. This means that the rank of a full-rank matrix is the lesser of the number of rows and columns, i.e., $\mathrm{rk}(\boldsymbol{A}) = \min(m, n)$. A matrix is said to be *rank deficient* if it does not have full rank. full rank rank deficient

$\diamondsuit$

Example 2.18 (Rank)

- $\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$.

  $\boldsymbol{A}$ has two linearly independent rows/columns so that $\mathrm{rk}(\boldsymbol{A}) = 2$.

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

  We use Gaussian elimination to determine the rank:

$$\begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix} \rightsquigarrow \cdots \rightsquigarrow \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.84}$$

  Here, we see that the number of linearly independent rows and columns is 2, such that $\mathrm{rk}(\boldsymbol{A}) = 2$.

2.7 Linear Mappings

In the following, we will study mappings on vector spaces that preserve their structure, which will allow us to define the concept of a coordinate. In the beginning of the chapter, we said that vectors are objects that can be added together and multiplied by a scalar, and the resulting object is still a vector. We wish to preserve this property when applying the mapping: Consider two real vector spaces V, W. A mapping $\Phi : V \to W$ preserves the structure of the vector space if

$$\Phi(\boldsymbol{x} + \boldsymbol{y}) = \Phi(\boldsymbol{x}) + \Phi(\boldsymbol{y}) \tag{2.85}$$

$$\Phi(\lambda \boldsymbol{x}) = \lambda \Phi(\boldsymbol{x}) \tag{2.86}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in V$ and $\lambda \in \mathbb{R}$. We can summarize this in the following definition:

linear mapping

vector space homomorphism

linear transformation

Definition 2.15 (Linear Mapping)**.** For vector spaces V, W, a mapping $\Phi : V \to W$ is called a *linear mapping* (or *vector space homomorphism/ linear transformation*) if

$$\forall \boldsymbol{x}, \boldsymbol{y} \in V \, \forall \lambda, \psi \in \mathbb{R} : \Phi(\lambda \boldsymbol{x} + \psi \boldsymbol{y}) = \lambda \Phi(\boldsymbol{x}) + \psi \Phi(\boldsymbol{y}) \,. \tag{2.87}$$

It turns out that we can represent linear mappings as matrices (Section 2.7.1). Recall that we can also collect a set of vectors as columns of a matrix. When working with matrices, we have to keep in mind what the matrix represents: a linear mapping or a collection of vectors. We will see more about linear mappings in Chapter 4. Before we continue, we will briefly introduce special mappings.

Definition 2.16 (Injective, Surjective, Bijective)**.** Consider a mapping $\Phi : \mathcal{V} \to \mathcal{W}$, where $\mathcal{V}, \mathcal{W}$ can be arbitrary sets. Then Φ is called

injective

surjective

bijective

- *Injective* if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{V} : \Phi(\boldsymbol{x}) = \Phi(\boldsymbol{y}) \implies \boldsymbol{x} = \boldsymbol{y}$.
- *Surjective* if $\Phi(\mathcal{V}) = \mathcal{W}$.
- *Bijective* if it is injective and surjective.

If Φ is surjective, then every element in $\mathcal{W}$ can be "reached" from $\mathcal{V}$ using Φ. A bijective Φ can be "undone", i.e., there exists a mapping $\Psi : \mathcal{W} \to \mathcal{V}$ so that $\Psi \circ \Phi(\boldsymbol{x}) = \boldsymbol{x}$. This mapping Ψ is then called the inverse of Φ and normally denoted by Φ^{-1}.

With these definitions, we introduce the following special cases of linear mappings between vector spaces V and W:

isomorphism
endomorphism
automorphism

- *Isomorphism:* $\Phi : V \to W$ linear and bijective
- *Endomorphism:* $\Phi : V \to V$ linear
- *Automorphism:* $\Phi : V \to V$ linear and bijective
- We define $\mathrm{id}_V : V \to V$, $\boldsymbol{x} \mapsto \boldsymbol{x}$ as the *identity mapping* or *identity automorphism* in V.

identity mapping
identity automorphism

Example 2.19 (Homomorphism)
The mapping $\Phi : \mathbb{R}^2 \to \mathbb{C},\ \Phi(\boldsymbol{x}) = x_1 + ix_2$, is a homomorphism:

$$\begin{aligned}
\Phi\left(\begin{bmatrix}x_1\\x_2\end{bmatrix} + \begin{bmatrix}y_1\\y_2\end{bmatrix}\right) &= (x_1+y_1) + i(x_2+y_2) = x_1 + ix_2 + y_1 + iy_2 \\
&= \Phi\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right) + \Phi\left(\begin{bmatrix}y_1\\y_2\end{bmatrix}\right) \\
\Phi\left(\lambda\begin{bmatrix}x_1\\x_2\end{bmatrix}\right) &= \lambda x_1 + \lambda i x_2 = \lambda(x_1 + ix_2) = \lambda\Phi\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right).
\end{aligned} \tag{2.88}$$

This also justifies why complex numbers can be represented as tuples in $\mathbb{R}^2$: There is a bijective linear mapping that converts the elementwise addition of tuples in $\mathbb{R}^2$ into the set of complex numbers with the corresponding addition. Note that we only showed linearity, but not the bijection.

Theorem 2.17 (Theorem 3.59 in Axler (2015))**.** *Finite-dimensional vector spaces V and W are isomorphic if and only if* $\dim(V) = \dim(W)$.

Theorem 2.17 states that there exists a linear, bijective mapping between two vector spaces of the same dimension. Intuitively, this means that vector spaces of the same dimension are kind of the same thing, as they can be transformed into each other without incurring any loss.

Theorem 2.17 also gives us the justification to treat $\mathbb{R}^{m\times n}$ (the vector space of $m \times n$-matrices) and $\mathbb{R}^{mn}$ (the vector space of vectors of length mn) the same, as their dimensions are mn, and there exists a linear, bijective mapping that transforms one into the other.

Remark. Consider vector spaces V, W, X. Then:

- For linear mappings $\Phi : V \to W$ and $\Psi : W \to X$, the mapping $\Psi \circ \Phi : V \to X$ is also linear.
- If $\Phi : V \to W$ is an isomorphism, then $\Phi^{-1} : W \to V$ is an isomorphism, too.

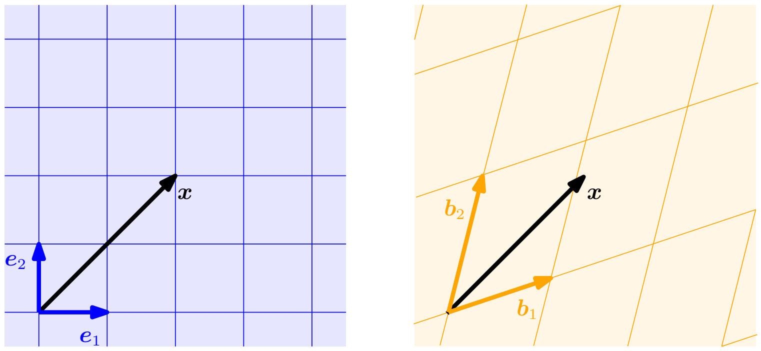


Figure 2.1 Two different coordinate systems defined by two sets of basis vectors. A vector $\boldsymbol{x}$ has different coordinate representations depending on which coordinate system is chosen.

- If $\Phi : V \to W,\ \Psi : V \to W$ are linear, then $\Phi + \Psi$ and $\lambda\Phi,\ \lambda \in \mathbb{R}$, are linear, too.

$\diamondsuit$

2.7.1 Matrix Representation of Linear Mappings

Any n-dimensional vector space is isomorphic to $\mathbb{R}^n$ (Theorem 2.17). We consider a basis $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ of an n-dimensional vector space V. In the following, the order of the basis vectors will be important. Therefore, we write

$$B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \tag{2.89}$$

ordered basis

and call this n-tuple an *ordered basis* of V.

Remark (Notation). We are at the point where notation gets a bit tricky. Therefore, we summarize some parts here. $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ is an ordered basis, $\mathcal{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ is an (unordered) basis, and $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n]$ is a matrix whose columns are the vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$. $\diamondsuit$

Definition 2.18 (Coordinates)**.** Consider a vector space V and an ordered basis $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ of V. For any $\boldsymbol{x} \in V$ we obtain a unique representation (linear combination)

$$\boldsymbol{x} = \alpha_1 \boldsymbol{b}_1 + \ldots + \alpha_n \boldsymbol{b}_n \tag{2.90}$$

coordinate

of $\boldsymbol{x}$ with respect to B. Then $\alpha_1, \ldots, \alpha_n$ are the *coordinates* of $\boldsymbol{x}$ with respect to B, and the vector

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \in \mathbb{R}^n \tag{2.91}$$

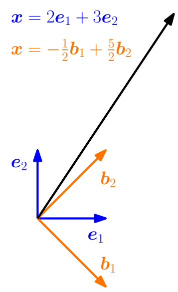
coordinate vector
coordinate representation

is the *coordinate vector/coordinate representation* of $\boldsymbol{x}$ with respect to the ordered basis B.

A basis effectively defines a coordinate system. We are familiar with the Cartesian coordinate system in two dimensions, which is spanned by the canonical basis vectors $\boldsymbol{e}_1, \boldsymbol{e}_2$. In this coordinate system, a vector $\boldsymbol{x} \in \mathbb{R}^2$ has a representation that tells us how to linearly combine $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ to obtain $\boldsymbol{x}$. However, any basis of $\mathbb{R}^2$ defines a valid coordinate system, and the same vector $\boldsymbol{x}$ from before may have a different coordinate representation in the $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ basis. In Figure 2.1, the coordinates of $\boldsymbol{x}$ with respect to the standard basis $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ is $[2, 2]^\top$. However, with respect to the basis $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ the same vector $\boldsymbol{x}$ is represented as $[1.09, 0.72]^\top$, i.e., $\boldsymbol{x} = 1.09\boldsymbol{b}_1 + 0.72\boldsymbol{b}_2$. In the following sections, we will discover how to obtain this representation.

Example 2.20
Let us have a look at a geometric vector $\boldsymbol{x} \in \mathbb{R}^2$ with coordinates $[2, 3]^\top$ with respect to the standard basis $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ of $\mathbb{R}^2$. This means, we can write $\boldsymbol{x} = 2\boldsymbol{e}_1 + 3\boldsymbol{e}_2$. However, we do not have to choose the standard basis to represent this vector. If we use the basis vectors $\boldsymbol{b}_1 = [1, -1]^\top, \boldsymbol{b}_2 = [1, 1]^\top$ we will obtain the coordinates $\frac{1}{2}[-1, 5]^\top$ to represent the same vector with respect to $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ (see Figure 2.2).

Figure 2.2 Different coordinate representations of a vector $\boldsymbol{x}$, depending on the choice of basis.



Remark. For an n-dimensional vector space V and an ordered basis B of V, the mapping $\Phi : \mathbb{R}^n \to V$, $\Phi(\boldsymbol{e}_i) = \boldsymbol{b}_i$, $i = 1, \ldots, n$, is linear (and because of Theorem 2.17 an isomorphism), where $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n)$ is the standard basis of $\mathbb{R}^n$.

$\diamondsuit$

Now we are ready to make an explicit connection between matrices and linear mappings between finite-dimensional vector spaces.

Definition 2.19 (Transformation Matrix)**.** Consider vector spaces V, W with corresponding (ordered) bases $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ and $C = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_m)$. Moreover, we consider a linear mapping $\Phi : V \to W$. For $j \in \{1, \ldots, n\}$,

$$\Phi(\boldsymbol{b}_j) = \alpha_{1j}\boldsymbol{c}_1 + \cdots + \alpha_{mj}\boldsymbol{c}_m = \sum_{i=1}^{m} \alpha_{ij}\boldsymbol{c}_i \tag{2.92}$$

is the unique representation of $\Phi(\boldsymbol{b}_j)$ with respect to C. Then, we call the $m \times n$-matrix $\boldsymbol{A}_\Phi$, whose elements are given by

$$A_\Phi(i, j) = \alpha_{ij} \,, \tag{2.93}$$

the *transformation matrix* of Φ (with respect to the ordered bases B of V and C of W).

transformation matrix

The coordinates of $\Phi(\boldsymbol{b}_j)$ with respect to the ordered basis C of W are the j-th column of $\boldsymbol{A}_\Phi$. Consider (finite-dimensional) vector spaces V, W with ordered bases B, C and a linear mapping $\Phi : V \to W$ with transformation matrix $\boldsymbol{A}_\Phi$. If $\hat{\boldsymbol{x}}$ is the coordinate vector of $\boldsymbol{x} \in V$ with respect to B and $\hat{\boldsymbol{y}}$ the coordinate vector of $\boldsymbol{y} = \Phi(\boldsymbol{x}) \in W$ with respect to C, then

$$\hat{\boldsymbol{y}} = \boldsymbol{A}_\Phi \hat{\boldsymbol{x}} \,. \tag{2.94}$$

This means that the transformation matrix can be used to map coordinates with respect to an ordered basis in V to coordinates with respect to an ordered basis in W.

Example 2.21 (Transformation Matrix)

Consider a homomorphism $\Phi : V \to W$ and ordered bases $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_3)$ of V and $C = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_4)$ of W. With

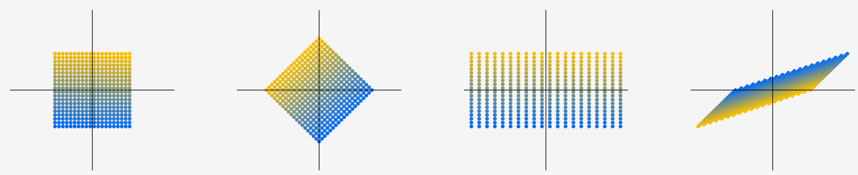
$$\begin{aligned} \Phi(\boldsymbol{b}_1) &= \boldsymbol{c}_1 - \boldsymbol{c}_2 + 3\boldsymbol{c}_3 - \boldsymbol{c}_4 \\ \Phi(\boldsymbol{b}_2) &= 2\boldsymbol{c}_1 + \boldsymbol{c}_2 + 7\boldsymbol{c}_3 + 2\boldsymbol{c}_4 \\ \Phi(\boldsymbol{b}_3) &= 3\boldsymbol{c}_2 + \boldsymbol{c}_3 + 4\boldsymbol{c}_4 \end{aligned} \tag{2.95}$$

the transformation matrix $\boldsymbol{A}_\Phi$ with respect to B and C satisfies $\Phi(\boldsymbol{b}_k) = \sum_{i=1}^4 \alpha_{ik}\boldsymbol{c}_i$ for $k = 1, \ldots, 3$ and is given as

$$\boldsymbol{A}_\Phi = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3] = \begin{bmatrix} 1 & 2 & 0 \\ -1 & 1 & 3 \\ 3 & 7 & 1 \\ -1 & 2 & 4 \end{bmatrix}, \tag{2.96}$$

where the $\boldsymbol{\alpha}_j$, $j = 1, 2, 3$, are the coordinate vectors of $\Phi(\boldsymbol{b}_j)$ with respect to C.

Example 2.22 (Linear Transformations of Vectors)



(a) Original data. (b) Rotation by $45°$. (c) Stretch along the horizontal axis. (d) General linear mapping.

Figure 2.3 Three examples of linear transformations of the vectors shown as dots in (a); (b) Rotation by $45°$; (c) Stretching of the horizontal coordinates by 2; (d) Combination of reflection, rotation and stretching.

We consider three linear transformations of a set of vectors in $\mathbb{R}^2$ with the transformation matrices

$$\boldsymbol{A}_1 = \begin{bmatrix} \cos(\frac{\pi}{4}) & -\sin(\frac{\pi}{4}) \\ \sin(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) \end{bmatrix}, \; \boldsymbol{A}_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \; \boldsymbol{A}_3 = \frac{1}{2}\begin{bmatrix} 3 & -1 \\ 1 & -1 \end{bmatrix}. \tag{2.97}$$

Figure 2.3 gives three examples of linear transformations of a set of vectors. Figure 2.3(a) shows 400 vectors in $\mathbb{R}^2$, each of which is represented by a dot at the corresponding (x_1, x_2)-coordinates. The vectors are arranged in a square. When we use matrix $\boldsymbol{A}_1$ in (2.97) to linearly transform each of these vectors, we obtain the rotated square in Figure 2.3(b). If we apply the linear mapping represented by $\boldsymbol{A}_2$, we obtain the rectangle in Figure 2.3(c) where each x_1-coordinate is stretched by 2. Figure 2.3(d) shows the original square from Figure 2.3(a) when linearly transformed using $\boldsymbol{A}_3$, which is a combination of a reflection, a rotation, and a stretch.

2.7.2 Basis Change

In the following, we will have a closer look at how transformation matrices of a linear mapping $\Phi : V \to W$ change if we change the bases in V and W. Consider two ordered bases

$$B = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_n), \quad \tilde{B} = (\tilde{\boldsymbol{b}}_1, \dots, \tilde{\boldsymbol{b}}_n) \tag{2.98}$$

of V and two ordered bases

$$C = (\boldsymbol{c}_1, \dots, \boldsymbol{c}_m), \quad \tilde{C} = (\tilde{\boldsymbol{c}}_1, \dots, \tilde{\boldsymbol{c}}_m) \tag{2.99}$$

of W. Moreover, $\boldsymbol{A}_\Phi \in \mathbb{R}^{m\times n}$ is the transformation matrix of the linear mapping $\Phi : V \to W$ with respect to the bases B and C, and $\tilde{\boldsymbol{A}}_\Phi \in \mathbb{R}^{m\times n}$ is the corresponding transformation mapping with respect to $\tilde{B}$ and $\tilde{C}$. In the following, we will investigate how $\boldsymbol{A}$ and $\tilde{\boldsymbol{A}}$ are related, i.e., how/whether we can transform $\boldsymbol{A}_\Phi$ into $\tilde{\boldsymbol{A}}_\Phi$ if we choose to perform a basis change from B, C to $\tilde{B}, \tilde{C}$.

Remark. We effectively get different coordinate representations of the identity mapping id_V. In the context of Figure 2.2, this would mean to map coordinates with respect to $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ onto coordinates with respect to $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ without changing the vector $\boldsymbol{x}$. By changing the basis and correspondingly the representation of vectors, the transformation matrix with respect to this new basis can have a particularly simple form that allows for straightforward computation. ♢

Example 2.23 (Basis Change)
Consider a transformation matrix

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{2.100}$$

with respect to the canonical basis in $\mathbb{R}^2$. If we define a new basis

$$B = (\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}) \tag{2.101}$$

we obtain a diagonal transformation matrix

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.102}$$

with respect to B, which is easier to work with than $\boldsymbol{A}$.

In the following, we will look at mappings that transform coordinate vectors with respect to one basis into coordinate vectors with respect to a different basis. We will state our main result first and then provide an explanation.

Theorem 2.20 (Basis Change)**.** *For a linear mapping $\Phi : V \to W$, ordered bases*

$$B = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_n), \quad \tilde{B} = (\tilde{\boldsymbol{b}}_1, \dots, \tilde{\boldsymbol{b}}_n) \tag{2.103}$$

of V and

$$C = (\boldsymbol{c}_1, \dots, \boldsymbol{c}_m), \quad \tilde{C} = (\tilde{\boldsymbol{c}}_1, \dots, \tilde{\boldsymbol{c}}_m) \tag{2.104}$$

of W, and a transformation matrix $\boldsymbol{A}_\Phi$ of Φ with respect to B and C, the corresponding transformation matrix $\tilde{\boldsymbol{A}}_\Phi$ with respect to the bases $\tilde{B}$ and $\tilde{C}$ is given as

$$\tilde{\boldsymbol{A}}_\Phi = \boldsymbol{T}^{-1} \boldsymbol{A}_\Phi \boldsymbol{S} \,. \tag{2.105}$$

Here, $\boldsymbol{S} \in \mathbb{R}^{n \times n}$ is the transformation matrix of id_V that maps coordinates with respect to $\tilde{B}$ onto coordinates with respect to B, and $\boldsymbol{T} \in \mathbb{R}^{m \times m}$ is the transformation matrix of id_W that maps coordinates with respect to $\tilde{C}$ onto coordinates with respect to C.

Proof Following Drumm and Weil (2001), we can write the vectors of the new basis $\tilde{B}$ of V as a linear combination of the basis vectors of B, such that

$$\tilde{\boldsymbol{b}}_j = s_{1j} \boldsymbol{b}_1 + \dots + s_{nj} \boldsymbol{b}_n = \sum_{i=1}^{n} s_{ij} \boldsymbol{b}_i \,, \quad j = 1, \dots, n \,. \tag{2.106}$$

Similarly, we write the new basis vectors $\tilde{C}$ of W as a linear combination of the basis vectors of C, which yields

$$\tilde{\boldsymbol{c}}_k = t_{1k} \boldsymbol{c}_1 + \dots + t_{mk} \boldsymbol{c}_m = \sum_{l=1}^{m} t_{lk} \boldsymbol{c}_l \,, \quad k = 1, \dots, m \,. \tag{2.107}$$

We define $\boldsymbol{S} = ((s_{ij})) \in \mathbb{R}^{n \times n}$ as the transformation matrix that maps coordinates with respect to $\tilde{B}$ onto coordinates with respect to B and $\boldsymbol{T} = ((t_{lk})) \in \mathbb{R}^{m \times m}$ as the transformation matrix that maps coordinates with respect to $\tilde{C}$ onto coordinates with respect to C. In particular, the jth column of $\boldsymbol{S}$ is the coordinate representation of $\tilde{\boldsymbol{b}}_j$ with respect to B and the kth column of $\boldsymbol{T}$ is the coordinate representation of $\tilde{\boldsymbol{c}}_k$ with respect to C. Note that both $\boldsymbol{S}$ and $\boldsymbol{T}$ are regular.

We are going to look at $\Phi(\tilde{\boldsymbol{b}}_j)$ from two perspectives. First, applying the mapping Φ, we get that for all $j = 1, \dots, n$

$$\Phi(\tilde{\boldsymbol{b}}_j) = \sum_{k=1}^{m} \underbrace{\tilde{a}_{kj}\tilde{\boldsymbol{c}}_k}_{\in W} \overset{(2.107)}{=} \sum_{k=1}^{m} \tilde{a}_{kj} \sum_{l=1}^{m} t_{lk}\boldsymbol{c}_l = \sum_{l=1}^{m} \left(\sum_{k=1}^{m} t_{lk}\tilde{a}_{kj} \right) \boldsymbol{c}_l \,, \tag{2.108}$$

where we first expressed the new basis vectors $\tilde{\boldsymbol{c}}_k \in W$ as linear combinations of the basis vectors $\boldsymbol{c}_l \in W$ and then swapped the order of summation.

Alternatively, when we express the $\tilde{\boldsymbol{b}}_j \in V$ as linear combinations of $\boldsymbol{b}_j \in V$, we arrive at

$$\Phi(\tilde{\boldsymbol{b}}_j) \overset{(2.106)}{=} \Phi\left(\sum_{i=1}^{n} s_{ij}\boldsymbol{b}_i \right) = \sum_{i=1}^{n} s_{ij}\Phi(\boldsymbol{b}_i) = \sum_{i=1}^{n} s_{ij} \sum_{l=1}^{m} a_{li}\boldsymbol{c}_l \tag{2.109a}$$

$$= \sum_{l=1}^{m} \left(\sum_{i=1}^{n} a_{li}s_{ij} \right) \boldsymbol{c}_l \,, \quad j = 1, \dots, n \,, \tag{2.109b}$$

where we exploited the linearity of Φ. Comparing (2.108) and (2.109b), it follows for all $j = 1, \dots, n$ and $l = 1, \dots, m$ that

$$\sum_{k=1}^{m} t_{lk}\tilde{a}_{kj} = \sum_{i=1}^{n} a_{li}s_{ij} \tag{2.110}$$

and, therefore,

$$\boldsymbol{T}\tilde{\boldsymbol{A}}_\Phi = \boldsymbol{A}_\Phi\boldsymbol{S} \in \mathbb{R}^{m\times n} \,, \tag{2.111}$$

such that

$$\tilde{\boldsymbol{A}}_\Phi = \boldsymbol{T}^{-1}\boldsymbol{A}_\Phi\boldsymbol{S} \,, \tag{2.112}$$

which proves Theorem 2.20. □

Theorem 2.20 tells us that with a basis change in V (B is replaced with $\tilde{B}$) and W (C is replaced with $\tilde{C}$), the transformation matrix $\boldsymbol{A}_\Phi$ of a linear mapping $\Phi : V \to W$ is replaced by an equivalent matrix $\tilde{\boldsymbol{A}}_\Phi$ with

$$\tilde{\boldsymbol{A}}_\Phi = \boldsymbol{T}^{-1}\boldsymbol{A}_\Phi\boldsymbol{S}. \tag{2.113}$$

Figure 2.2 illustrates this relation: Consider a homomorphism $\Phi : V \to W$ and ordered bases $B, \tilde{B}$ of V and $C, \tilde{C}$ of W. The mapping Φ_{CB} is an instantiation of Φ and maps basis vectors of B onto linear combinations of basis vectors of C. Assume that we know the transformation matrix $\boldsymbol{A}_\Phi$ of Φ_{CB} with respect to the ordered bases B, C. When we perform a basis change from B to $\tilde{B}$ in V and from C to $\tilde{C}$ in W, we can determine the

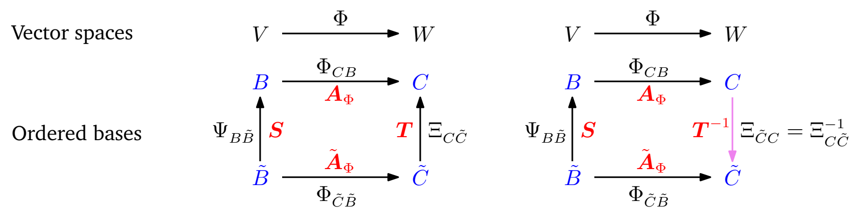


Figure 2.2 For a homomorphism $\Phi : V \to W$ and ordered bases $B, \tilde{B}$ of V and $C, \tilde{C}$ of W (marked in blue), we can express the mapping $\Phi_{\tilde{C}\tilde{B}}$ with respect to the bases $\tilde{B}, \tilde{C}$ equivalently as a composition of the homomorphisms $\Phi_{\tilde{C}\tilde{B}} = \Xi_{\tilde{C}C} \circ \Phi_{CB} \circ \Psi_{B\tilde{B}}$ with respect to the bases in the subscripts. The corresponding transformation matrices are in red.

corresponding transformation matrix $\tilde{\boldsymbol{A}}_\Phi$ as follows: First, we find the matrix representation of the linear mapping $\Psi_{B\tilde{B}} : V \to V$ that maps coordinates with respect to the new basis $\tilde{B}$ onto the (unique) coordinates with respect to the "old" basis B (in V). Then, we use the transformation matrix $\boldsymbol{A}_\Phi$ of $\Phi_{CB} : V \to W$ to map these coordinates onto the coordinates with respect to C in W. Finally, we use a linear mapping $\Xi_{\tilde{C}C} : W \to W$ to map the coordinates with respect to C onto coordinates with respect to $\tilde{C}$. Therefore, we can express the linear mapping $\Phi_{\tilde{C}\tilde{B}}$ as a composition of linear mappings that involve the "old" basis:

$$\Phi_{\tilde{C}\tilde{B}} = \Xi_{\tilde{C}C} \circ \Phi_{CB} \circ \Psi_{B\tilde{B}} = \Xi^{-1}_{C\tilde{C}} \circ \Phi_{CB} \circ \Psi_{B\tilde{B}} \,. \tag{2.114}$$

Concretely, we use $\Psi_{B\tilde{B}} = \mathrm{id}_V$ and $\Xi_{C\tilde{C}} = \mathrm{id}_W$, i.e., the identity mappings that map vectors onto themselves, but with respect to a different basis.

equivalent

Definition 2.21 (Equivalence)**.** Two matrices $\boldsymbol{A}, \tilde{\boldsymbol{A}} \in \mathbb{R}^{m\times n}$ are *equivalent* if there exist regular matrices $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{T} \in \mathbb{R}^{m\times m}$, such that $\tilde{\boldsymbol{A}} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{S}$.

similar

Definition 2.22 (Similarity)**.** Two matrices $\boldsymbol{A}, \tilde{\boldsymbol{A}} \in \mathbb{R}^{n\times n}$ are *similar* if there exists a regular matrix $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ with $\tilde{\boldsymbol{A}} = \boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{S}$

Remark. Similar matrices are always equivalent. However, equivalent matrices are not necessarily similar. $\diamondsuit$

Remark. Consider vector spaces V, W, X. From the remark that follows Theorem 2.17, we already know that for linear mappings $\Phi : V \to W$ and $\Psi : W \to X$ the mapping $\Psi \circ \Phi : V \to X$ is also linear. With transformation matrices $\boldsymbol{A}_\Phi$ and $\boldsymbol{A}_\Psi$ of the corresponding mappings, the overall transformation matrix is $\boldsymbol{A}_{\Psi\circ\Phi} = \boldsymbol{A}_\Psi\boldsymbol{A}_\Phi$. $\diamondsuit$

In light of this remark, we can look at basis changes from the perspective of composing linear mappings:

- $\boldsymbol{A}_\Phi$ is the transformation matrix of a linear mapping $\Phi_{CB} : V \to W$ with respect to the bases B, C.
- $\tilde{\boldsymbol{A}}_\Phi$ is the transformation matrix of the linear mapping $\Phi_{\tilde{C}\tilde{B}} : V \to W$ with respect to the bases $\tilde{B}, \tilde{C}$.
- $\boldsymbol{S}$ is the transformation matrix of a linear mapping $\Psi_{B\tilde{B}} : V \to V$ (automorphism) that represents $\tilde{B}$ in terms of B. Normally, $\Psi = \mathrm{id}_V$ is the identity mapping in V.

- $\boldsymbol{T}$ is the transformation matrix of a linear mapping $\Xi_{C\tilde{C}} : W \to W$ (automorphism) that represents $\tilde{C}$ in terms of C. Normally, $\Xi = \mathrm{id}_W$ is the identity mapping in W.

If we (informally) write down the transformations just in terms of bases, then $\boldsymbol{A}_\Phi : B \to C$, $\tilde{\boldsymbol{A}}_\Phi : \tilde{B} \to \tilde{C}$, $\boldsymbol{S} : \tilde{B} \to B$, $\boldsymbol{T} : \tilde{C} \to C$ and $\boldsymbol{T}^{-1} : C \to \tilde{C}$, and

$$\tilde{B} \to \tilde{C} = \tilde{B} \to B \to C \to \tilde{C} \tag{2.115}$$

$$\tilde{\boldsymbol{A}}_\Phi = \boldsymbol{T}^{-1} \boldsymbol{A}_\Phi \boldsymbol{S}\,. \tag{2.116}$$

Note that the execution order in (2.116) is from right to left because vectors are multiplied at the right-hand side so that $\boldsymbol{x} \mapsto \boldsymbol{S}\boldsymbol{x} \mapsto \boldsymbol{A}_\Phi(\boldsymbol{S}\boldsymbol{x}) \mapsto \boldsymbol{T}^{-1}\big(\boldsymbol{A}_\Phi(\boldsymbol{S}\boldsymbol{x})\big) = \tilde{\boldsymbol{A}}_\Phi \boldsymbol{x}$.

Example 2.24 (Basis Change)

Consider a linear mapping $\Phi : \mathbb{R}^3 \to \mathbb{R}^4$ whose transformation matrix is

$$\boldsymbol{A}_\Phi = \begin{bmatrix} 1 & 2 & 0 \\ -1 & 1 & 3 \\ 3 & 7 & 1 \\ -1 & 2 & 4 \end{bmatrix} \tag{2.117}$$

with respect to the standard bases

$$B = \left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right), \quad C = \left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right). \tag{2.118}$$

We seek the transformation matrix $\tilde{\boldsymbol{A}}_\Phi$ of Φ with respect to the new bases

$$\tilde{B} = \left(\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right) \in \mathbb{R}^3, \quad \tilde{C} = \left(\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right). \tag{2.119}$$

Then,

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{T} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.120}$$

where the ith column of $\boldsymbol{S}$ is the coordinate representation of $\tilde{\boldsymbol{b}}_i$ in terms of the basis vectors of B. Since B is the standard basis, the coordinate representation is straightforward to find. For a general basis B, we would need to solve a linear equation system to find the λ_i such that

$\sum_{i=1}^{3} \lambda_i \boldsymbol{b}_i = \tilde{\boldsymbol{b}}_j$, $j = 1, \ldots, 3$. Similarly, the jth column of $\boldsymbol{T}$ is the coordinate representation of $\tilde{\boldsymbol{c}}_j$ in terms of the basis vectors of C.

Therefore, we obtain

$$\tilde{\boldsymbol{A}}_\Phi = \boldsymbol{T}^{-1}\boldsymbol{A}_\Phi \boldsymbol{S} = \frac{1}{2}\begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2 \end{bmatrix}\begin{bmatrix} 3 & 2 & 1 \\ 0 & 4 & 2 \\ 10 & 8 & 4 \\ 1 & 6 & 3 \end{bmatrix} \tag{2.121a}$$

$$= \begin{bmatrix} -4 & -4 & -2 \\ 6 & 0 & 0 \\ 4 & 8 & 4 \\ 1 & 6 & 3 \end{bmatrix}. \tag{2.121b}$$

In Chapter 4, we will be able to exploit the concept of a basis change to find a basis with respect to which the transformation matrix of an endomorphism has a particularly simple (diagonal) form. In Chapter 10, we will look at a data compression problem and find a convenient basis onto which we can project the data while minimizing the compression loss.

2.7.3 Image and Kernel

The image and kernel of a linear mapping are vector subspaces with certain important properties. In the following, we will characterize them more carefully.

kernel
null space

Definition 2.23 (Image and Kernel)**.**
For $\Phi : V \to W$, we define the *kernel/null space*

$$\ker(\Phi) := \Phi^{-1}(\mathbf{0}_W) = \{\boldsymbol{v} \in V : \Phi(\boldsymbol{v}) = \mathbf{0}_W\} \tag{2.122}$$

image
range

and the *image/range*

$$\mathrm{Im}(\Phi) := \Phi(V) = \{\boldsymbol{w} \in W | \exists \boldsymbol{v} \in V : \Phi(\boldsymbol{v}) = \boldsymbol{w}\}. \tag{2.123}$$

domain
codomain

We also call V and W also the *domain* and *codomain* of Φ, respectively.

Intuitively, the kernel is the set of vectors $\boldsymbol{v} \in V$ that Φ maps onto the neutral element $\mathbf{0}_W \in W$. The image is the set of vectors $\boldsymbol{w} \in W$ that can be "reached" by Φ from any vector in V. An illustration is given in Figure 2.2.

Remark. Consider a linear mapping $\Phi : V \to W$, where V, W are vector spaces.

- It always holds that $\Phi(\mathbf{0}_V) = \mathbf{0}_W$ and, therefore, $\mathbf{0}_V \in \ker(\Phi)$. In particular, the null space is never empty.
- $\mathrm{Im}(\Phi) \subseteq W$ is a subspace of W, and $\ker(\Phi) \subseteq V$ is a subspace of V.

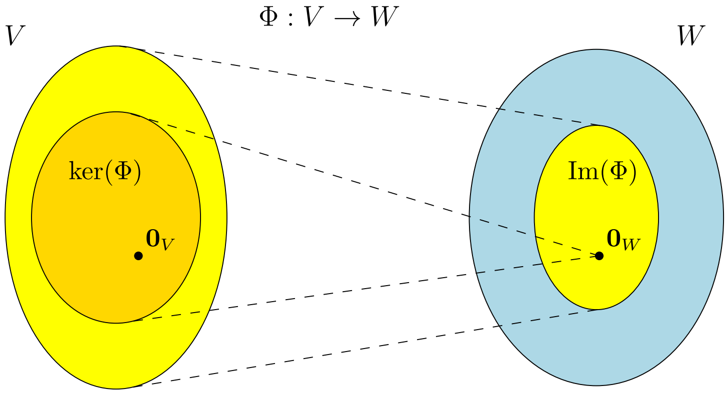


Figure 2.2 Kernel and image of a linear mapping $\Phi : V \to W$.

- Φ is injective (one-to-one) if and only if $\ker(\Phi) = \{\mathbf{0}\}$.

$\diamondsuit$

Remark (Null Space and Column Space). Let us consider $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and a linear mapping $\Phi : \mathbb{R}^n \to \mathbb{R}^m$, $\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}$.

- For $\boldsymbol{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_n]$, where $\boldsymbol{a}_i$ are the columns of $\boldsymbol{A}$, we obtain

$$\mathrm{Im}(\Phi) = \{\boldsymbol{A}\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^n\} = \left\{\sum_{i=1}^{n} x_i \boldsymbol{a}_i : x_1, \dots, x_n \in \mathbb{R}\right\} \tag{2.124a}$$

$$= \mathrm{span}[\boldsymbol{a}_1, \dots, \boldsymbol{a}_n] \subseteq \mathbb{R}^m , \tag{2.124b}$$

  i.e., the image is the span of the columns of $\boldsymbol{A}$, also called the *column space*. Therefore, the column space (image) is a subspace of $\mathbb{R}^m$, where m is the "height" of the matrix.

column space

- $\mathrm{rk}(\boldsymbol{A}) = \dim(\mathrm{Im}(\Phi))$.
- The kernel/null space $\ker(\Phi)$ is the general solution to the homogeneous system of linear equations $\boldsymbol{A}\boldsymbol{x} = \mathbf{0}$ and captures all possible linear combinations of the elements in $\mathbb{R}^n$ that produce $\mathbf{0} \in \mathbb{R}^m$.
- The kernel is a subspace of $\mathbb{R}^n$, where n is the "width" of the matrix.
- The kernel focuses on the relationship among the columns, and we can use it to determine whether/how we can express a column as a linear combination of other columns.

$\diamondsuit$

Example 2.25 (Image and Kernel of a Linear Mapping)

The mapping

$$\Phi : \mathbb{R}^4 \to \mathbb{R}^2, \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \mapsto \begin{bmatrix} 1 & 2 & -1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 + 2x_2 - x_3 \\ x_1 + x_4 \end{bmatrix} \tag{2.125a}$$

$$= x_1 \begin{bmatrix}1\\1\end{bmatrix} + x_2 \begin{bmatrix}2\\0\end{bmatrix} + x_3 \begin{bmatrix}-1\\0\end{bmatrix} + x_4 \begin{bmatrix}0\\1\end{bmatrix} \tag{2.125b}$$

is linear. To determine $\mathrm{Im}(\Phi)$, we can take the span of the columns of the transformation matrix and obtain

$$\mathrm{Im}(\Phi) = \mathrm{span}[\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}-1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}]. \tag{2.126}$$

To compute the kernel (null space) of Φ, we need to solve $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, i.e., we need to solve a homogeneous equation system. To do this, we use Gaussian elimination to transform $\boldsymbol{A}$ into reduced row-echelon form:

$$\begin{bmatrix}1 & 2 & -1 & 0\\1 & 0 & 0 & 1\end{bmatrix} \rightsquigarrow \cdots \rightsquigarrow \begin{bmatrix}1 & 0 & 0 & 1\\0 & 1 & -\frac{1}{2} & -\frac{1}{2}\end{bmatrix}. \tag{2.127}$$

This matrix is in reduced row-echelon form, and we can use the Minus-1 Trick to compute a basis of the kernel (see Section 2.3.3). Alternatively, we can express the non-pivot columns (columns 3 and 4) as linear combinations of the pivot columns (columns 1 and 2). The third column $\boldsymbol{a}_3$ is equivalent to $-\frac{1}{2}$ times the second column $\boldsymbol{a}_2$. Therefore, $\boldsymbol{0} = \boldsymbol{a}_3 + \frac{1}{2}\boldsymbol{a}_2$. In the same way, we see that $\boldsymbol{a}_4 = \boldsymbol{a}_1 - \frac{1}{2}\boldsymbol{a}_2$ and, therefore, $\boldsymbol{0} = \boldsymbol{a}_1 - \frac{1}{2}\boldsymbol{a}_2 - \boldsymbol{a}_4$. Overall, this gives us the kernel (null space) as

$$\ker(\Phi) = \mathrm{span}[\begin{bmatrix}0\\\frac{1}{2}\\1\\0\end{bmatrix}, \begin{bmatrix}-1\\\frac{1}{2}\\0\\1\end{bmatrix}]. \tag{2.128}$$

rank-nullity theorem

Theorem 2.24 (Rank-Nullity Theorem)**.** *For vector spaces V, W and a linear mapping $\Phi : V \to W$ it holds that*

$$\dim(\ker(\Phi)) + \dim(\mathrm{Im}(\Phi)) = \dim(V). \tag{2.129}$$

fundamental theorem of linear mappings

The rank-nullity theorem is also referred to as the *fundamental theorem of linear mappings* (Axler, 2015, theorem 3.22). The following are direct consequences of Theorem 2.24:

- If $\dim(\mathrm{Im}(\Phi)) < \dim(V)$, then $\ker(\Phi)$ is non-trivial, i.e., the kernel contains more than $\boldsymbol{0}_V$ and $\dim(\ker(\Phi)) \geqslant 1$.
- If $\boldsymbol{A}_\Phi$ is the transformation matrix of Φ with respect to an ordered basis and $\dim(\mathrm{Im}(\Phi)) < \dim(V)$, then the system of linear equations $\boldsymbol{A}_\Phi\boldsymbol{x} = \boldsymbol{0}$ has infinitely many solutions.
- If $\dim(V) = \dim(W)$, then the following three-way equivalence holds:
  - Φ is injective
  - Φ is surjective
  - Φ is bijective

  since $\mathrm{Im}(\Phi) \subseteq W$.

2.8 Affine Spaces

In the following, we will have a closer look at spaces that are offset from the origin, i.e., spaces that are no longer vector subspaces. Moreover, we will briefly discuss properties of mappings between these affine spaces, which resemble linear mappings.

Remark. In the machine learning literature, the distinction between linear and affine is sometimes not clear so that we can find references to affine spaces/mappings as linear spaces/mappings. ♢

2.8.1 Affine Subspaces

Definition 2.25 (Affine Subspace)**.** Let V be a vector space, $\boldsymbol{x}_0 \in V$ and $U \subseteq V$ a subspace. Then the subset

$$L = \boldsymbol{x}_0 + U := \{\boldsymbol{x}_0 + \boldsymbol{u} : \boldsymbol{u} \in U\} \tag{2.130a}$$
$$= \{\boldsymbol{v} \in V | \exists \boldsymbol{u} \in U : \boldsymbol{v} = \boldsymbol{x}_0 + \boldsymbol{u}\} \subseteq V \tag{2.130b}$$

is called *affine subspace* or *linear manifold* of V. U is called *direction* or *direction space*, and $\boldsymbol{x}_0$ is called *support point*. In Chapter 12, we refer to such a subspace as a *hyperplane*.

affine subspace
linear manifold
direction
direction space
support point
hyperplane

Note that the definition of an affine subspace excludes $\boldsymbol{0}$ if $\boldsymbol{x}_0 \notin U$. Therefore, an affine subspace is not a (linear) subspace (vector subspace) of V for $\boldsymbol{x}_0 \notin U$.

Examples of affine subspaces are points, lines, and planes in $\mathbb{R}^3$, which do not (necessarily) go through the origin.

Remark. Consider two affine subspaces $L = \boldsymbol{x}_0 + U$ and $\tilde{L} = \tilde{\boldsymbol{x}}_0 + \tilde{U}$ of a vector space V. Then, $L \subseteq \tilde{L}$ if and only if $U \subseteq \tilde{U}$ and $\boldsymbol{x}_0 - \tilde{\boldsymbol{x}}_0 \in \tilde{U}$.

Affine subspaces are often described by *parameters*: Consider a k-dimensional affine space $L = \boldsymbol{x}_0 + U$ of V. If $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_k)$ is an ordered basis of U, then every element $\boldsymbol{x} \in L$ can be uniquely described as

$$\boldsymbol{x} = \boldsymbol{x}_0 + \lambda_1 \boldsymbol{b}_1 + \ldots + \lambda_k \boldsymbol{b}_k \,, \tag{2.131}$$

where $\lambda_1, \dots, \lambda_k \in \mathbb{R}$. This representation is called *parametric equation* of L with directional vectors $\boldsymbol{b}_1, \dots, \boldsymbol{b}_k$ and *parameters* $\lambda_1, \dots, \lambda_k$. ♢

parametric equation
parameters

Example 2.26 (Affine Subspaces)

- One-dimensional affine subspaces are called *lines* and can be written as $\boldsymbol{y} = \boldsymbol{x}_0 + \lambda \boldsymbol{b}_1$, where $\lambda \in \mathbb{R}$ and $U = \operatorname{span}[\boldsymbol{b}_1] \subseteq \mathbb{R}^n$ is a one-dimensional subspace of $\mathbb{R}^n$. This means that a line is defined by a support point $\boldsymbol{x}_0$ and a vector $\boldsymbol{b}_1$ that defines the direction. See Figure 2.2 for an illustration.

line

plane

- Two-dimensional affine subspaces of $\mathbb{R}^n$ are called *planes*. The parametric equation for planes is $\boldsymbol{y} = \boldsymbol{x}_0 + \lambda_1 \boldsymbol{b}_1 + \lambda_2 \boldsymbol{b}_2$, where $\lambda_1, \lambda_2 \in \mathbb{R}$ and $U = \mathrm{span}[\boldsymbol{b}_1, \boldsymbol{b}_2] \subseteq \mathbb{R}^n$. This means that a plane is defined by a support point $\boldsymbol{x}_0$ and two linearly independent vectors $\boldsymbol{b}_1, \boldsymbol{b}_2$ that span the direction space.

hyperplane

- In $\mathbb{R}^n$, the $(n-1)$-dimensional affine subspaces are called *hyperplanes*, and the corresponding parametric equation is $\boldsymbol{y} = \boldsymbol{x}_0 + \sum_{i=1}^{n-1} \lambda_i \boldsymbol{b}_i$, where $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{n-1}$ form a basis of an $(n-1)$-dimensional subspace U of $\mathbb{R}^n$. This means that a hyperplane is defined by a support point $\boldsymbol{x}_0$ and $(n-1)$ linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{n-1}$ that span the direction space. In $\mathbb{R}^2$, a line is also a hyperplane. In $\mathbb{R}^3$, a plane is also a hyperplane.

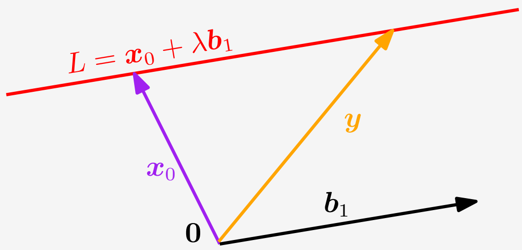


Figure 2.2 Lines are affine subspaces. Vectors $\boldsymbol{y}$ on a line $\boldsymbol{x}_0 + \lambda \boldsymbol{b}_1$ lie in an affine subspace L with support point $\boldsymbol{x}_0$ and direction $\boldsymbol{b}_1$.

Remark (Inhomogeneous systems of linear equations and affine subspaces). For $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{x} \in \mathbb{R}^m$, the solution of the system of linear equations $\boldsymbol{A\lambda} = \boldsymbol{x}$ is either the empty set or an affine subspace of $\mathbb{R}^n$ of dimension $n - \mathrm{rk}(\boldsymbol{A})$. In particular, the solution of the linear equation $\lambda_1 \boldsymbol{b}_1 + \ldots + \lambda_n \boldsymbol{b}_n = \boldsymbol{x}$, where $(\lambda_1, \ldots, \lambda_n) \neq (0, \ldots, 0)$, is a hyperplane in $\mathbb{R}^n$.

In $\mathbb{R}^n$, every k-dimensional affine subspace is the solution of an inhomogeneous system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{R}^{m \times n}, \boldsymbol{b} \in \mathbb{R}^m$ and $\mathrm{rk}(\boldsymbol{A}) = n - k$. Recall that for homogeneous equation systems $\boldsymbol{Ax} = \boldsymbol{0}$ the solution was a vector subspace, which we can also think of as a special affine space with support point $\boldsymbol{x}_0 = \boldsymbol{0}$. ♢

2.8.2 *Affine Mappings*

Similar to linear mappings between vector spaces, which we discussed in Section 2.7, we can define affine mappings between two affine spaces. Linear and affine mappings are closely related. Therefore, many properties that we already know from linear mappings, e.g., that the composition of linear mappings is a linear mapping, also hold for affine mappings.

Definition 2.26 (Affine Mapping). For two vector spaces V, W, a linear mapping $\Phi : V \to W$, and $\boldsymbol{a} \in W$, the mapping

$$\phi : V \to W \tag{2.132}$$

$$\boldsymbol{x} \mapsto \boldsymbol{a} + \Phi(\boldsymbol{x}) \tag{2.133}$$

is an *affine mapping* from V to W. The vector $\boldsymbol{a}$ is called the *translation vector* of ϕ.

affine mapping

translation vector

- Every affine mapping $\phi : V \to W$ is also the composition of a linear mapping $\Phi : V \to W$ and a translation $\tau : W \to W$ in W, such that $\phi = \tau \circ \Phi$. The mappings Φ and τ are uniquely determined.
- The composition $\phi' \circ \phi$ of affine mappings $\phi : V \to W$, $\phi' : W \to X$ is affine.
- Affine mappings keep the geometric structure invariant. They also preserve the dimension and parallelism.

2.9 Further Reading

There are many resources for learning linear algebra, including the textbooks by Strang (2003), Golan (2007), Axler (2015), and Liesen and Mehrmann (2015). There are also several online resources that we mentioned in the introduction to this chapter. We only covered Gaussian elimination here, but there are many other approaches for solving systems of linear equations, and we refer to numerical linear algebra textbooks by Stoer and Burlirsch (2002), Golub and Van Loan (2012), and Horn and Johnson (2013) for an in-depth discussion.

In this book, we distinguish between the topics of linear algebra (e.g., vectors, matrices, linear independence, basis) and topics related to the geometry of a vector space. In Chapter 3, we will introduce the inner product, which induces a norm. These concepts allow us to define angles, lengths and distances, which we will use for orthogonal projections. Projections turn out to be key in many machine learning algorithms, such as linear regression and principal component analysis, both of which we will cover in Chapters 9 and 10, respectively.

Exercises

2.1 We consider $(\mathbb{R}\backslash\{-1\}, \star)$, where

$$a \star b := ab + a + b, \qquad a, b \in \mathbb{R}\backslash\{-1\} \tag{2.134}$$

a. Show that $(\mathbb{R}\backslash\{-1\}, \star)$ is an Abelian group.
b. Solve

$$3 \star x \star x = 15$$

in the Abelian group $(\mathbb{R}\backslash\{-1\}, \star)$, where $\star$ is defined in (2.134).

2.2 Let n be in $\mathbb{N}\backslash\{0\}$. Let k, x be in $\mathbb{Z}$. We define the congruence class $\bar{k}$ of the integer k as the set

$$\begin{aligned}\overline{k} &= \{x \in \mathbb{Z} \mid x - k = 0 \pmod{n}\} \\ &= \{x \in \mathbb{Z} \mid \exists a \in \mathbb{Z} \colon (x - k = n \cdot a)\}\,.\end{aligned}$$

We now define $\mathbb{Z}/n\mathbb{Z}$ (sometimes written $\mathbb{Z}_n$) as the set of all congruence classes modulo n. Euclidean division implies that this set is a finite set containing n elements:

$$\mathbb{Z}_n = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\}$$

For all $\overline{a}, \overline{b} \in \mathbb{Z}_n$, we define

$$\overline{a} \oplus \overline{b} := \overline{a+b}$$

a. Show that $(\mathbb{Z}_n, \oplus)$ is a group. Is it Abelian?
b. We now define another operation $\otimes$ for all $\overline{a}$ and $\overline{b}$ in $\mathbb{Z}_n$ as

$$\overline{a} \otimes \overline{b} = \overline{a \times b}\,, \tag{2.135}$$

where $a \times b$ represents the usual multiplication in $\mathbb{Z}$.
Let $n = 5$. Draw the times table of the elements of $\mathbb{Z}_5\backslash\{\overline{0}\}$ under $\otimes$, i.e., calculate the products $\overline{a} \otimes \overline{b}$ for all $\overline{a}$ and $\overline{b}$ in $\mathbb{Z}_5\backslash\{\overline{0}\}$.
Hence, show that $\mathbb{Z}_5\backslash\{\overline{0}\}$ is closed under $\otimes$ and possesses a neutral element for $\otimes$. Display the inverse of all elements in $\mathbb{Z}_5\backslash\{\overline{0}\}$ under $\otimes$. Conclude that $(\mathbb{Z}_5\backslash\{\overline{0}\}, \otimes)$ is an Abelian group.
c. Show that $(\mathbb{Z}_8\backslash\{\overline{0}\}, \otimes)$ is not a group.
d. We recall that the Bézout theorem states that two integers a and b are relatively prime (i.e., $gcd(a, b) = 1$) if and only if there exist two integers u and v such that $au + bv = 1$. Show that $(\mathbb{Z}_n\backslash\{\overline{0}\}, \otimes)$ is a group if and only if $n \in \mathbb{N}\backslash\{0\}$ is prime.

2.3 Consider the set $\mathcal{G}$ of 3×3 matrices defined as follows:

$$\mathcal{G} = \left\{ \begin{bmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3} \,\middle|\, x, y, z \in \mathbb{R} \right\}$$

We define $\cdot$ as the standard matrix multiplication.
Is $(\mathcal{G}, \cdot)$ a group? If yes, is it Abelian? Justify your answer.

2.4 Compute the following matrix products, if possible:

a.

$$\begin{bmatrix}1 & 2\\4 & 5\\7 & 8\end{bmatrix}\begin{bmatrix}1 & 1 & 0\\0 & 1 & 1\\1 & 0 & 1\end{bmatrix}$$

b.

$$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}\begin{bmatrix}1 & 1 & 0\\0 & 1 & 1\\1 & 0 & 1\end{bmatrix}$$

c.

$$\begin{bmatrix}1 & 1 & 0\\0 & 1 & 1\\1 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$$

d.

$$\begin{bmatrix}1 & 2 & 1 & 2\\4 & 1 & -1 & -4\end{bmatrix}\begin{bmatrix}0 & 3\\1 & -1\\2 & 1\\5 & 2\end{bmatrix}$$

e.

$$\begin{bmatrix}0 & 3\\1 & -1\\2 & 1\\5 & 2\end{bmatrix}\begin{bmatrix}1 & 2 & 1 & 2\\4 & 1 & -1 & -4\end{bmatrix}$$

2.5 Find the set $\mathcal{S}$ of all solutions in $\boldsymbol{x}$ of the following inhomogeneous linear systems $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A}$ and $\boldsymbol{b}$ are defined as follows:

a.

$$\boldsymbol{A} = \begin{bmatrix}1 & 1 & -1 & -1\\2 & 5 & -7 & -5\\2 & -1 & 1 & 3\\5 & 2 & -4 & 2\end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix}1\\-2\\4\\6\end{bmatrix}$$

b.

$$\boldsymbol{A} = \begin{bmatrix}1 & -1 & 0 & 0 & 1\\1 & 1 & 0 & -3 & 0\\2 & -1 & 0 & 1 & -1\\-1 & 2 & 0 & -2 & -1\end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix}3\\6\\5\\-1\end{bmatrix}$$

2.6 Using Gaussian elimination, find all solutions of the inhomogeneous equation system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ with

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

2.7 Find all solutions in $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3$ of the equation system $\boldsymbol{A}\boldsymbol{x} = 12\boldsymbol{x}$, where

$$\boldsymbol{A} = \begin{bmatrix} 6 & 4 & 3 \\ 6 & 0 & 9 \\ 0 & 8 & 0 \end{bmatrix}$$

and $\sum_{i=1}^{3} x_i = 1$.

2.8 Determine the inverses of the following matrices if possible:

a.

$$\boldsymbol{A} = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{bmatrix}$$

b.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

2.9 Which of the following sets are subspaces of $\mathbb{R}^3$?

a. $A = \{(\lambda, \lambda + \mu^3, \lambda - \mu^3) \mid \lambda, \mu \in \mathbb{R}\}$

b. $B = \{(\lambda^2, -\lambda^2, 0) \mid \lambda \in \mathbb{R}\}$

c. Let γ be in $\mathbb{R}$.
$C = \{(\xi_1, \xi_2, \xi_3) \in \mathbb{R}^3 \mid \xi_1 - 2\xi_2 + 3\xi_3 = \gamma\}$

d. $D = \{(\xi_1, \xi_2, \xi_3) \in \mathbb{R}^3 \mid \xi_2 \in \mathbb{Z}\}$

2.10 Are the following sets of vectors linearly independent?

a.

$$\boldsymbol{x}_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

b.

$$\boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

2.11 Write

$$\boldsymbol{y} = \begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}$$

as linear combination of

$$\boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \boldsymbol{x}_3 = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

2.12 Consider two subspaces of $\mathbb{R}^4$:

$$U_1 = \text{span}[\begin{bmatrix}1\\1\\-3\\1\end{bmatrix}, \begin{bmatrix}2\\-1\\0\\-1\end{bmatrix}, \begin{bmatrix}-1\\1\\-1\\1\end{bmatrix}], \quad U_2 = \text{span}[\begin{bmatrix}-1\\-2\\2\\1\end{bmatrix}, \begin{bmatrix}2\\-2\\0\\0\end{bmatrix}, \begin{bmatrix}-3\\6\\-2\\-1\end{bmatrix}].$$

Determine a basis of $U_1 \cap U_2$.

2.13 Consider two subspaces U_1 and U_2, where U_1 is the solution space of the homogeneous equation system $\boldsymbol{A}_1\boldsymbol{x} = \boldsymbol{0}$ and U_2 is the solution space of the homogeneous equation system $\boldsymbol{A}_2\boldsymbol{x} = \boldsymbol{0}$ with

$$\boldsymbol{A}_1 = \begin{bmatrix}1 & 0 & 1\\1 & -2 & -1\\2 & 1 & 3\\1 & 0 & 1\end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix}3 & -3 & 0\\1 & 2 & 3\\7 & -5 & 2\\3 & -1 & 2\end{bmatrix}.$$

a. Determine the dimension of U_1, U_2.
b. Determine bases of U_1 and U_2.
c. Determine a basis of $U_1 \cap U_2$.

2.14 Consider two subspaces U_1 and U_2, where U_1 is spanned by the columns of $\boldsymbol{A}_1$ and U_2 is spanned by the columns of $\boldsymbol{A}_2$ with

$$\boldsymbol{A}_1 = \begin{bmatrix}1 & 0 & 1\\1 & -2 & -1\\2 & 1 & 3\\1 & 0 & 1\end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix}3 & -3 & 0\\1 & 2 & 3\\7 & -5 & 2\\3 & -1 & 2\end{bmatrix}.$$

a. Determine the dimension of U_1, U_2
b. Determine bases of U_1 and U_2
c. Determine a basis of $U_1 \cap U_2$

2.15 Let $F = \{(x, y, z) \in \mathbb{R}^3 \mid x+y-z = 0\}$ and $G = \{(a-b, a+b, a-3b) \mid a, b \in \mathbb{R}\}$.

a. Show that F and G are subspaces of $\mathbb{R}^3$.
b. Calculate $F \cap G$ without resorting to any basis vector.
c. Find one basis for F and one for G, calculate $F \cap G$ using the basis vectors previously found and check your result with the previous question.

2.16 Are the following mappings linear?

a. Let $a, b \in \mathbb{R}$.

$$\begin{aligned}\Phi : L^1([a, b]) &\to \mathbb{R}\\ f \mapsto \Phi(f) &= \int_a^b f(x)dx\,,\end{aligned}$$

where $L^1([a, b])$ denotes the set of integrable functions on $[a, b]$.

b.

$$\begin{aligned}\Phi : C^1 &\to C^0\\ f \mapsto \Phi(f) &= f'\,,\end{aligned}$$

where for $k \geqslant 1$, C^k denotes the set of k times continuously differentiable functions, and C^0 denotes the set of continuous functions.

c.

$$\begin{aligned}\Phi &: \mathbb{R} \to \mathbb{R}\\ x &\mapsto \Phi(x) = \cos(x)\end{aligned}$$

d.

$$\begin{aligned}\Phi &: \mathbb{R}^3 \to \mathbb{R}^2\\ \boldsymbol{x} &\mapsto \begin{bmatrix}1 & 2 & 3\\ 1 & 4 & 3\end{bmatrix}\boldsymbol{x}\end{aligned}$$

e. Let θ be in $[0, 2\pi[$ and

$$\begin{aligned}\Phi &: \mathbb{R}^2 \to \mathbb{R}^2\\ \boldsymbol{x} &\mapsto \begin{bmatrix}\cos(\theta) & \sin(\theta)\\ -\sin(\theta) & \cos(\theta)\end{bmatrix}\boldsymbol{x}\end{aligned}$$

2.17 Consider the linear mapping

$$\begin{aligned}\Phi &: \mathbb{R}^3 \to \mathbb{R}^4\\ \Phi\left(\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}\right) &= \begin{bmatrix}3x_1 + 2x_2 + x_3\\ x_1 + x_2 + x_3\\ x_1 - 3x_2\\ 2x_1 + 3x_2 + x_3\end{bmatrix}\end{aligned}$$

- Find the transformation matrix $\boldsymbol{A}_\Phi$.
- Determine $\mathrm{rk}(\boldsymbol{A}_\Phi)$.
- Compute the kernel and image of Φ. What are $\dim(\ker(\Phi))$ and $\dim(\mathrm{Im}(\Phi))$?

2.18 Let E be a vector space. Let f and g be two automorphisms on E such that $f \circ g = \mathrm{id}_E$ (i.e., $f \circ g$ is the identity mapping id_E). Show that $\ker(f) = \ker(g \circ f)$, $\mathrm{Im}(g) = \mathrm{Im}(g \circ f)$ and that $\ker(f) \cap \mathrm{Im}(g) = \{\boldsymbol{0}_E\}$.

2.19 Consider an endomorphism $\Phi : \mathbb{R}^3 \to \mathbb{R}^3$ whose transformation matrix (with respect to the standard basis in $\mathbb{R}^3$) is

$$\boldsymbol{A}_\Phi = \begin{bmatrix}1 & 1 & 0\\ 1 & -1 & 0\\ 1 & 1 & 1\end{bmatrix}.$$

a. Determine $\ker(\Phi)$ and $\mathrm{Im}(\Phi)$.

b. Determine the transformation matrix $\tilde{\boldsymbol{A}}_\Phi$ with respect to the basis

$$B = \left(\begin{bmatrix}1\\1\\1\end{bmatrix}, \begin{bmatrix}1\\2\\1\end{bmatrix}, \begin{bmatrix}1\\0\\0\end{bmatrix}\right),$$

i.e., perform a basis change toward the new basis B.

2.20 Let us consider $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}'_1, \boldsymbol{b}'_2$, 4 vectors of $\mathbb{R}^2$ expressed in the standard basis of $\mathbb{R}^2$ as

$$\boldsymbol{b}_1 = \begin{bmatrix}2\\1\end{bmatrix}, \quad \boldsymbol{b}_2 = \begin{bmatrix}-1\\-1\end{bmatrix}, \quad \boldsymbol{b}'_1 = \begin{bmatrix}2\\-2\end{bmatrix}, \quad \boldsymbol{b}'_2 = \begin{bmatrix}1\\1\end{bmatrix}$$

and let us define two ordered bases $B = (\boldsymbol{b}_1, \boldsymbol{b}_2)$ and $B' = (\boldsymbol{b}'_1, \boldsymbol{b}'_2)$ of $\mathbb{R}^2$.

a. Show that B and B' are two bases of $\mathbb{R}^2$ and draw those basis vectors.
b. Compute the matrix $\boldsymbol{P}_1$ that performs a basis change from B' to B.
c. We consider $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3$, three vectors of $\mathbb{R}^3$ defined in the standard basis of $\mathbb{R}^3$ as

$$\boldsymbol{c}_1 = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}, \quad \boldsymbol{c}_2 = \begin{bmatrix} 0 \\ -1 \\ 2 \end{bmatrix}, \quad \boldsymbol{c}_3 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

and we define $C = (\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3)$.

(i) Show that C is a basis of $\mathbb{R}^3$, e.g., by using determinants (see Section 4.1).
(ii) Let us call $C' = (\boldsymbol{c}'_1, \boldsymbol{c}'_2, \boldsymbol{c}'_3)$ the standard basis of $\mathbb{R}^3$. Determine the matrix $\boldsymbol{P}_2$ that performs the basis change from C to C'.

d. We consider a homomorphism $\Phi : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$, such that

$$\begin{aligned} \Phi(\boldsymbol{b}_1 + \boldsymbol{b}_2) &= \boldsymbol{c}_2 + \boldsymbol{c}_3 \\ \Phi(\boldsymbol{b}_1 - \boldsymbol{b}_2) &= 2\boldsymbol{c}_1 - \boldsymbol{c}_2 + 3\boldsymbol{c}_3 \end{aligned}$$

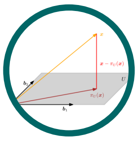
where $B = (\boldsymbol{b}_1, \boldsymbol{b}_2)$ and $C = (\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3)$ are ordered bases of $\mathbb{R}^2$ and $\mathbb{R}^3$, respectively.
Determine the transformation matrix $\boldsymbol{A}_\Phi$ of Φ with respect to the ordered bases B and C.

e. Determine $\boldsymbol{A}'$, the transformation matrix of Φ with respect to the bases $\boldsymbol{B}'$ and $\boldsymbol{C}'$.
f. Let us consider the vector $\boldsymbol{x} \in \mathbb{R}^2$ whose coordinates in B' are $[2, 3]^\top$. In other words, $\boldsymbol{x} = 2\boldsymbol{b}'_1 + 3\boldsymbol{b}'_2$.

(i) Calculate the coordinates of $\boldsymbol{x}$ in B.
(ii) Based on that, compute the coordinates of $\Phi(\boldsymbol{x})$ expressed in C.
(iii) Then, write $\Phi(\boldsymbol{x})$ in terms of $\boldsymbol{c}'_1, \boldsymbol{c}'_2, \boldsymbol{c}'_3$.
(iv) Use the representation of $\boldsymbol{x}$ in B' and the matrix $\boldsymbol{A}'$ to find this result directly.

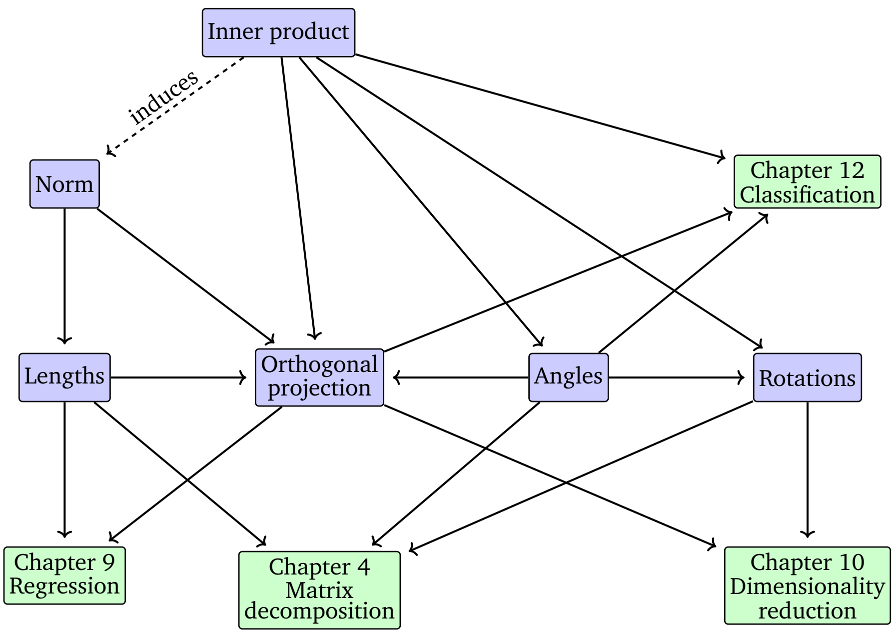
3

Analytic Geometry



In Chapter 2, we studied vectors, vector spaces, and linear mappings at a general but abstract level. In this chapter, we will add some geometric interpretation and intuition to all of these concepts. In particular, we will look at geometric vectors and compute their lengths and distances or angles between two vectors. To be able to do this, we equip the vector space with an inner product that induces the geometry of the vector space. Inner products and their corresponding norms and metrics capture the intuitive notions of similarity and distances, which we use to develop the support vector machine in Chapter 12. We will then use the concepts of lengths and angles between vectors to discuss orthogonal projections, which will play a central role when we discuss principal component analysis in Chapter 10 and regression via maximum likelihood estimation in Chapter 9. Figure 3.1 gives an overview of how concepts in this chapter are related and how they are connected to other chapters of the book.

Figure 3.1 A mind map of the concepts introduced in this chapter, along with when they are used in other parts of the book.



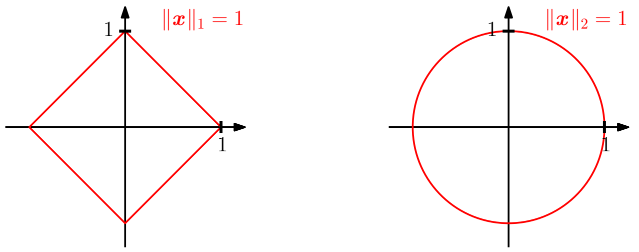


Figure 3.1 For different norms, the red lines indicate the set of vectors with norm 1. Left: Manhattan norm; Right: Euclidean distance.

3.1 Norms

When we think of geometric vectors, i.e., directed line segments that start at the origin, then intuitively the length of a vector is the distance of the "end" of this directed line segment from the origin. In the following, we will discuss the notion of the length of vectors using the concept of a norm.

Definition 3.1 (Norm)**.** A *norm* on a vector space V is a function

norm

$$\|\cdot\| : V \to \mathbb{R}\,, \tag{3.1}$$

$$\boldsymbol{x} \mapsto \|\boldsymbol{x}\|\,, \tag{3.2}$$

which assigns each vector $\boldsymbol{x}$ its *length* $\|\boldsymbol{x}\| \in \mathbb{R}$, such that for all $\lambda \in \mathbb{R}$ and $\boldsymbol{x}, \boldsymbol{y} \in V$ the following hold:

length

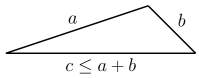
absolutely homogeneous

- *Absolutely homogeneous:* $\|\lambda\boldsymbol{x}\| = |\lambda|\|\boldsymbol{x}\|$
- *Triangle inequality:* $\|\boldsymbol{x} + \boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$
- *Positive definite:* $\|\boldsymbol{x}\| \geqslant 0$ and $\|\boldsymbol{x}\| = 0 \iff \boldsymbol{x} = \boldsymbol{0}$

triangle inequality

positive definite

Figure 3.2 Triangle inequality.



In geometric terms, the triangle inequality states that for any triangle, the sum of the lengths of any two sides must be greater than or equal to the length of the remaining side; see Figure 3.2 for an illustration. Definition 3.1 is in terms of a general vector space V (Section 2.4), but in this book we will only consider a finite-dimensional vector space $\mathbb{R}^n$. Recall that for a vector $\boldsymbol{x} \in \mathbb{R}^n$ we denote the elements of the vector using a subscript, that is, x_i is the i^{th} element of the vector $\boldsymbol{x}$.

Example 3.1 (Manhattan Norm)

The *Manhattan norm* on $\mathbb{R}^n$ is defined for $\boldsymbol{x} \in \mathbb{R}^n$ as

Manhattan norm

$$\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i|\,, \tag{3.3}$$

where $|\cdot|$ is the absolute value. The left panel of Figure 3.1 shows all vectors $\boldsymbol{x} \in \mathbb{R}^2$ with $\|\boldsymbol{x}\|_1 = 1$. The Manhattan norm is also called ℓ_1 *norm*.

ℓ_1 norm

Euclidean norm

Example 3.2 (Euclidean Norm)
The *Euclidean norm* of $\boldsymbol{x} \in \mathbb{R}^n$ is defined as

$$\|\boldsymbol{x}\|_2 := \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}} \tag{3.4}$$

Euclidean distance

ℓ_2 norm

and computes the *Euclidean distance* of $\boldsymbol{x}$ from the origin. The right panel of Figure 3.1 shows all vectors $\boldsymbol{x} \in \mathbb{R}^2$ with $\|\boldsymbol{x}\|_2 = 1$. The Euclidean norm is also called *ℓ_2 norm*.

Remark. Throughout this book, we will use the Euclidean norm (3.4) by default if not stated otherwise. ♢

3.2 Inner Products

Inner products allow for the introduction of intuitive geometrical concepts, such as the length of a vector and the angle or distance between two vectors. A major purpose of inner products is to determine whether vectors are orthogonal to each other.

3.2.1 Dot Product

scalar product

dot product

We may already be familiar with a particular type of inner product, the *scalar product/dot product* in $\mathbb{R}^n$, which is given by

$$\boldsymbol{x}^\top \boldsymbol{y} = \sum_{i=1}^{n} x_i y_i \,. \tag{3.5}$$

We will refer to this particular inner product as the dot product in this book. However, inner products are more general concepts with specific properties, which we will now introduce.

3.2.2 General Inner Products

bilinear mapping

Recall the linear mapping from Section 2.7, where we can rearrange the mapping with respect to addition and multiplication with a scalar. A *bilinear mapping* Ω is a mapping with two arguments, and it is linear in each argument, i.e., when we look at a vector space V then it holds that for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in V,\ \lambda, \psi \in \mathbb{R}$ that

$$\Omega(\lambda \boldsymbol{x} + \psi \boldsymbol{y}, \boldsymbol{z}) = \lambda \Omega(\boldsymbol{x}, \boldsymbol{z}) + \psi \Omega(\boldsymbol{y}, \boldsymbol{z}) \tag{3.6}$$

$$\Omega(\boldsymbol{x}, \lambda \boldsymbol{y} + \psi \boldsymbol{z}) = \lambda \Omega(\boldsymbol{x}, \boldsymbol{y}) + \psi \Omega(\boldsymbol{x}, \boldsymbol{z}) \,. \tag{3.7}$$

Here, (3.6) asserts that Ω is linear in the first argument, and (3.7) asserts that Ω is linear in the second argument (see also (2.87)).

Definition 3.2. Let V be a vector space and $\Omega : V \times V \to \mathbb{R}$ be a bilinear mapping that takes two vectors and maps them onto a real number. Then

- Ω is called *symmetric* if $\Omega(\boldsymbol{x}, \boldsymbol{y}) = \Omega(\boldsymbol{y}, \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{y} \in V$, i.e., the order of the arguments does not matter.
- Ω is called *positive definite* if

symmetric

positive definite

$$\forall \boldsymbol{x} \in V \backslash \{\boldsymbol{0}\} : \Omega(\boldsymbol{x}, \boldsymbol{x}) > 0 \,, \quad \Omega(\boldsymbol{0}, \boldsymbol{0}) = 0 \,. \tag{3.8}$$

Definition 3.3. Let V be a vector space and $\Omega : V \times V \to \mathbb{R}$ be a bilinear mapping that takes two vectors and maps them onto a real number. Then

- A positive definite, symmetric bilinear mapping $\Omega : V \times V \to \mathbb{R}$ is called an *inner product* on V. We typically write $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ instead of $\Omega(\boldsymbol{x}, \boldsymbol{y})$.
- The pair $(V, \langle \cdot, \cdot \rangle)$ is called an *inner product space* or (real) *vector space with inner product*. If we use the dot product defined in (3.5), we call $(V, \langle \cdot, \cdot \rangle)$ a *Euclidean vector space*.

inner product

inner product space

vector space with inner product

Euclidean vector space

We will refer to these spaces as inner product spaces in this book.

Example 3.3 (Inner Product That Is Not the Dot Product)
Consider $V = \mathbb{R}^2$. If we define

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle := x_1 y_1 - (x_1 y_2 + x_2 y_1) + 2 x_2 y_2 \tag{3.9}$$

then $\langle \cdot, \cdot \rangle$ is an inner product but different from the dot product. The proof will be an exercise.

3.2.3 Symmetric, Positive Definite Matrices

Symmetric, positive definite matrices play an important role in machine learning, and they are defined via the inner product. In Section 4.3, we will return to symmetric, positive definite matrices in the context of matrix decompositions. The idea of symmetric positive semidefinite matrices is key in the definition of kernels (Section 12.4).

Consider an n-dimensional vector space V with an inner product $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ (see Definition 3.3) and an ordered basis $B = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n)$ of V. Recall from Section 2.6.1 that any vectors $\boldsymbol{x}, \boldsymbol{y} \in V$ can be written as linear combinations of the basis vectors so that $\boldsymbol{x} = \sum_{i=1}^{n} \psi_i \boldsymbol{b}_i \in V$ and $\boldsymbol{y} = \sum_{j=1}^{n} \lambda_j \boldsymbol{b}_j \in V$ for suitable $\psi_i, \lambda_j \in \mathbb{R}$. Due to the bilinearity of the inner product, it holds for all $\boldsymbol{x}, \boldsymbol{y} \in V$ that

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \left\langle \sum_{i=1}^{n} \psi_i \boldsymbol{b}_i, \sum_{j=1}^{n} \lambda_j \boldsymbol{b}_j \right\rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} \psi_i \langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle \lambda_j = \hat{\boldsymbol{x}}^\top \boldsymbol{A} \hat{\boldsymbol{y}} \,, \tag{3.10}$$

where $A_{ij} := \langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle$ and $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}$ are the coordinates of $\boldsymbol{x}$ and $\boldsymbol{y}$ with respect to the basis B. This implies that the inner product $\langle \cdot, \cdot \rangle$ is uniquely determined through $\boldsymbol{A}$. The symmetry of the inner product also means that $\boldsymbol{A}$

is symmetric. Furthermore, the positive definiteness of the inner product implies that

$$\forall \boldsymbol{x} \in V \backslash\{\mathbf{0}\}: \boldsymbol{x}^{\top} \boldsymbol{A} \boldsymbol{x}>0 . \tag{3.11}$$

symmetric, positive definite
positive definite
symmetric, positive semidefinite

Definition 3.4 (Symmetric, Positive Definite Matrix)**.** A symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ that satisfies (3.11) is called *symmetric, positive definite*, or just *positive definite*. If only $\geqslant$ holds in (3.11), then $\boldsymbol{A}$ is called *symmetric, positive semidefinite*.

Example 3.4 (Symmetric, Positive Definite Matrices)
Consider the matrices

$$\boldsymbol{A}_1=\begin{bmatrix}9 & 6 \\ 6 & 5\end{bmatrix}, \quad \boldsymbol{A}_2=\begin{bmatrix}9 & 6 \\ 6 & 3\end{bmatrix} . \tag{3.12}$$

$\boldsymbol{A}_1$ is positive definite because it is symmetric and

$$\boldsymbol{x}^{\top} \boldsymbol{A}_1 \boldsymbol{x}=\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}9 & 6 \\ 6 & 5\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix} \tag{3.13a}$$

$$=9 x_1^2+12 x_1 x_2+5 x_2^2=(3 x_1+2 x_2)^2+x_2^2>0 \tag{3.13b}$$

for all $\boldsymbol{x} \in V \backslash\{\mathbf{0}\}$. In contrast, $\boldsymbol{A}_2$ is symmetric but not positive definite because $\boldsymbol{x}^{\top} \boldsymbol{A}_2 \boldsymbol{x}=9 x_1^2+12 x_1 x_2+3 x_2^2=(3 x_1+2 x_2)^2-x_2^2$ can be less than 0, e.g., for $\boldsymbol{x}=[2,-3]^{\top}$.

If $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric, positive definite, then

$$\langle\boldsymbol{x}, \boldsymbol{y}\rangle=\hat{\boldsymbol{x}}^{\top} \boldsymbol{A} \hat{\boldsymbol{y}} \tag{3.14}$$

defines an inner product with respect to an ordered basis B, where $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ are the coordinate representations of $\boldsymbol{x}, \boldsymbol{y} \in V$ with respect to B.

Theorem 3.5. *For a real-valued, finite-dimensional vector space V and an ordered basis B of V, it holds that $\langle\cdot, \cdot\rangle: V \times V \rightarrow \mathbb{R}$ is an inner product if and only if there exists a symmetric, positive definite matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ with*

$$\langle\boldsymbol{x}, \boldsymbol{y}\rangle=\hat{\boldsymbol{x}}^{\top} \boldsymbol{A} \hat{\boldsymbol{y}} . \tag{3.15}$$

The following properties hold if $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric and positive definite:

- The null space (kernel) of $\boldsymbol{A}$ consists only of $\mathbf{0}$ because $\boldsymbol{x}^{\top} \boldsymbol{A} \boldsymbol{x}>0$ for all $\boldsymbol{x} \neq \mathbf{0}$. This implies that $\boldsymbol{A} \boldsymbol{x} \neq \mathbf{0}$ if $\boldsymbol{x} \neq \mathbf{0}$.
- The diagonal elements a_{ii} of $\boldsymbol{A}$ are positive because $a_{ii}=\boldsymbol{e}_i^{\top} \boldsymbol{A} \boldsymbol{e}_i>0$, where $\boldsymbol{e}_i$ is the ith vector of the standard basis in $\mathbb{R}^n$.

3.3 Lengths and Distances

In Section 3.1, we already discussed norms that we can use to compute the length of a vector. Inner products and norms are closely related in the sense that any inner product induces a norm

Inner products induce norms.

$$\|\boldsymbol{x}\| := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle} \tag{3.16}$$

in a natural way, such that we can compute lengths of vectors using the inner product. However, not every norm is induced by an inner product. The Manhattan norm (3.3) is an example of a norm without a corresponding inner product. In the following, we will focus on norms that are induced by inner products and introduce geometric concepts, such as lengths, distances, and angles.

Remark (Cauchy-Schwarz Inequality). For an inner product vector space $(V, \langle\cdot,\cdot\rangle)$ the induced norm $\|\cdot\|$ satisfies the *Cauchy-Schwarz inequality*

Cauchy-Schwarz inequality

$$|\langle \boldsymbol{x}, \boldsymbol{y} \rangle| \leqslant \|\boldsymbol{x}\|\|\boldsymbol{y}\|\,. \tag{3.17}$$

$\diamondsuit$

Example 3.5 (Lengths of Vectors Using Inner Products)
In geometry, we are often interested in lengths of vectors. We can now use an inner product to compute them using (3.16). Let us take $\boldsymbol{x} = [1, 1]^\top \in \mathbb{R}^2$. If we use the dot product as the inner product, with (3.16) we obtain

$$\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}} = \sqrt{1^2 + 1^2} = \sqrt{2} \tag{3.18}$$

as the length of $\boldsymbol{x}$. Let us now choose a different inner product:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \begin{bmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix} \boldsymbol{y} = x_1 y_1 - \frac{1}{2}(x_1 y_2 + x_2 y_1) + x_2 y_2\,. \tag{3.19}$$

If we compute the norm of a vector, then this inner product returns smaller values than the dot product if x_1 and x_2 have the same sign (and $x_1 x_2 > 0$); otherwise, it returns greater values than the dot product. With this inner product, we obtain

$$\langle \boldsymbol{x}, \boldsymbol{x} \rangle = x_1^2 - x_1 x_2 + x_2^2 = 1 - 1 + 1 = 1 \implies \|\boldsymbol{x}\| = \sqrt{1} = 1\,, \tag{3.20}$$

such that $\boldsymbol{x}$ is "shorter" with this inner product than with the dot product.

Definition 3.6 (Distance and Metric)**.** Consider an inner product space $(V, \langle\cdot,\cdot\rangle)$. Then

$$d(\boldsymbol{x}, \boldsymbol{y}) := \|\boldsymbol{x} - \boldsymbol{y}\| = \sqrt{\langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle} \tag{3.21}$$

is called the *distance* between $\boldsymbol{x}$ and $\boldsymbol{y}$ for $\boldsymbol{x}, \boldsymbol{y} \in V$. If we use the dot product as the inner product, then the distance is called *Euclidean distance*.

distance

Euclidean distance

The mapping

$$d : V \times V \to \mathbb{R} \tag{3.22}$$

$$(\boldsymbol{x}, \boldsymbol{y}) \mapsto d(\boldsymbol{x}, \boldsymbol{y}) \tag{3.23}$$

metric

is called a *metric*.

Remark. Similar to the length of a vector, the distance between vectors does not require an inner product: a norm is sufficient. If we have a norm induced by an inner product, the distance may vary depending on the choice of the inner product. ◇

A metric d satisfies the following:

positive definite

symmetric

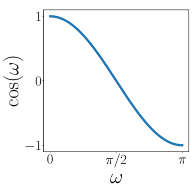
triangle inequality

1. d is *positive definite*, i.e., $d(\boldsymbol{x}, \boldsymbol{y}) \geqslant 0$ for all $\boldsymbol{x}, \boldsymbol{y} \in V$ and $d(\boldsymbol{x}, \boldsymbol{y}) = 0 \iff \boldsymbol{x} = \boldsymbol{y}$.
2. d is *symmetric*, i.e., $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{y} \in V$.
3. *Triangle inequality:* $d(\boldsymbol{x}, \boldsymbol{z}) \leqslant d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z})$ for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in V$.

Remark. At first glance, the lists of properties of inner products and metrics look very similar. However, by comparing Definition 3.3 with Definition 3.6 we observe that $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ and $d(\boldsymbol{x}, \boldsymbol{y})$ behave in opposite directions. Very similar $\boldsymbol{x}$ and $\boldsymbol{y}$ will result in a large value for the inner product and a small value for the metric. ◇

3.4 Angles and Orthogonality

Figure 3.2 When restricted to $[0, \pi]$ then $f(\omega) = \cos(\omega)$ returns a unique number in the interval $[-1, 1]$.



In addition to enabling the definition of lengths of vectors, as well as the distance between two vectors, inner products also capture the geometry of a vector space by defining the angle ω between two vectors. We use the Cauchy-Schwarz inequality (3.17) to define angles ω in inner product spaces between two vectors $\boldsymbol{x}, \boldsymbol{y}$, and this notion coincides with our intuition in $\mathbb{R}^2$ and $\mathbb{R}^3$. Assume that $\boldsymbol{x} \neq \boldsymbol{0}, \boldsymbol{y} \neq \boldsymbol{0}$. Then

$$-1 \leqslant \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{x}\| \, \|\boldsymbol{y}\|} \leqslant 1 \,. \tag{3.24}$$

Therefore, there exists a unique $\omega \in [0, \pi]$, illustrated in Figure 3.2, with

$$\cos \omega = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\|\boldsymbol{x}\| \, \|\boldsymbol{y}\|} \,. \tag{3.25}$$

angle

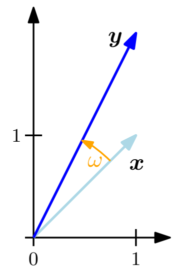
The number ω is the *angle* between the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. Intuitively, the angle between two vectors tells us how similar their orientations are. For example, using the dot product, the angle between $\boldsymbol{x}$ and $\boldsymbol{y} = 4\boldsymbol{x}$, i.e., $\boldsymbol{y}$ is a scaled version of $\boldsymbol{x}$, is 0: Their orientation is the same.

Example 3.6 (Angle between Vectors)
Let us compute the angle between $\boldsymbol{x} = [1, 1]^\top \in \mathbb{R}^2$ and $\boldsymbol{y} = [1, 2]^\top \in \mathbb{R}^2$; see Figure 3.3, where we use the dot product as the inner product. Then we get

$$\cos\omega = \frac{\langle \boldsymbol{x}, \boldsymbol{y}\rangle}{\sqrt{\langle \boldsymbol{x}, \boldsymbol{x}\rangle \langle \boldsymbol{y}, \boldsymbol{y}\rangle}} = \frac{\boldsymbol{x}^\top \boldsymbol{y}}{\sqrt{\boldsymbol{x}^\top \boldsymbol{x} \boldsymbol{y}^\top \boldsymbol{y}}} = \frac{3}{\sqrt{10}}\,, \tag{3.26}$$

and the angle between the two vectors is $\arccos(\frac{3}{\sqrt{10}}) \approx 0.32\,\mathrm{rad}$, which corresponds to about 18°.

Figure 3.3 The angle ω between two vectors $\boldsymbol{x}, \boldsymbol{y}$ is computed using the inner product.



A key feature of the inner product is that it also allows us to characterize vectors that are orthogonal.

orthogonal

orthonormal

Definition 3.7 (Orthogonality)**.** Two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are *orthogonal* if and only if $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = 0$, and we write $\boldsymbol{x} \perp \boldsymbol{y}$. If additionally $\|\boldsymbol{x}\| = 1 = \|\boldsymbol{y}\|$, i.e., the vectors are unit vectors, then $\boldsymbol{x}$ and $\boldsymbol{y}$ are *orthonormal*.

An implication of this definition is that the $\boldsymbol{0}$-vector is orthogonal to every vector in the vector space.

Remark. Orthogonality is the generalization of the concept of perpendicularity to bilinear forms that do not have to be the dot product. In our context, geometrically, we can think of orthogonal vectors as having a right angle with respect to a specific inner product. ♢

Example 3.7 (Orthogonal Vectors)

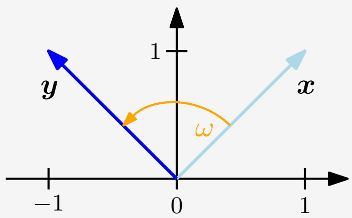


Figure 3.1 The angle ω between two vectors $\boldsymbol{x}, \boldsymbol{y}$ can change depending on the inner product.

Consider two vectors $\boldsymbol{x} = [1, 1]^\top, \boldsymbol{y} = [-1, 1]^\top \in \mathbb{R}^2$; see Figure 3.1. We are interested in determining the angle ω between them using two different inner products. Using the dot product as the inner product yields an angle ω between $\boldsymbol{x}$ and $\boldsymbol{y}$ of 90°, such that $\boldsymbol{x} \perp \boldsymbol{y}$. However, if we choose the inner product

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \boldsymbol{x}^\top \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{y}\,, \tag{3.27}$$

we get that the angle ω between $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by

$$\cos\omega = \frac{\langle \boldsymbol{x}, \boldsymbol{y}\rangle}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} = -\frac{1}{3} \implies \omega \approx 1.91\,\mathrm{rad} \approx 109.5^\circ\,, \tag{3.28}$$

and $\boldsymbol{x}$ and $\boldsymbol{y}$ are not orthogonal. Therefore, vectors that are orthogonal with respect to one inner product do not have to be orthogonal with respect to a different inner product.

orthogonal matrix

Definition 3.8 (Orthogonal Matrix)**.** A square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is an *orthogonal matrix* if and only if its columns are orthonormal so that

$$\boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{I} = \boldsymbol{A}^\top\boldsymbol{A}\,, \tag{3.29}$$

which implies that

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^\top\,, \tag{3.30}$$

i.e., the inverse is obtained by simply transposing the matrix.

It is convention to call these matrices "orthogonal" but a more precise description would be "orthonormal". Transformations with orthogonal matrices preserve distances and angles.

Transformations by orthogonal matrices are special because the length of a vector $\boldsymbol{x}$ is not changed when transforming it using an orthogonal matrix $\boldsymbol{A}$. For the dot product, we obtain

$$\|\boldsymbol{A}\boldsymbol{x}\|^2 = (\boldsymbol{A}\boldsymbol{x})^\top(\boldsymbol{A}\boldsymbol{x}) = \boldsymbol{x}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^\top\boldsymbol{I}\boldsymbol{x} = \boldsymbol{x}^\top\boldsymbol{x} = \|\boldsymbol{x}\|^2\,. \tag{3.31}$$

Moreover, the angle between any two vectors $\boldsymbol{x}, \boldsymbol{y}$, as measured by their inner product, is also unchanged when transforming both of them using an orthogonal matrix $\boldsymbol{A}$. Assuming the dot product as the inner product, the angle of the images $\boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{A}\boldsymbol{y}$ is given as

$$\cos\omega = \frac{(\boldsymbol{A}\boldsymbol{x})^\top(\boldsymbol{A}\boldsymbol{y})}{\|\boldsymbol{A}\boldsymbol{x}\|\,\|\boldsymbol{A}\boldsymbol{y}\|} = \frac{\boldsymbol{x}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{y}}{\sqrt{\boldsymbol{x}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x}\boldsymbol{y}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{y}}} = \frac{\boldsymbol{x}^\top\boldsymbol{y}}{\|\boldsymbol{x}\|\,\|\boldsymbol{y}\|}\,, \tag{3.32}$$

which gives exactly the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$. This means that orthogonal matrices $\boldsymbol{A}$ with $\boldsymbol{A}^\top = \boldsymbol{A}^{-1}$ preserve both angles and distances. It turns out that orthogonal matrices define transformations that are rotations (with the possibility of flips). In Section 3.9, we will discuss more details about rotations.

3.5 Orthonormal Basis

In Section 2.6.1, we characterized properties of basis vectors and found that in an n-dimensional vector space, we need n basis vectors, i.e., n vectors that are linearly independent. In Sections 3.3 and 3.4, we used inner products to compute the length of vectors and the angle between vectors. In the following, we will discuss the special case where the basis vectors are orthogonal to each other and where the length of each basis vector is 1. We will call this basis then an orthonormal basis.

Let us introduce this more formally.

Definition 3.9 (Orthonormal Basis). Consider an n-dimensional vector space V and a basis $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ of V. If

$$\langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle = 0 \quad \text{for } i \neq j \tag{3.33}$$

$$\langle \boldsymbol{b}_i, \boldsymbol{b}_i \rangle = 1 \tag{3.34}$$

for all $i, j = 1, \ldots, n$ then the basis is called an *orthonormal basis* (*ONB*). If only (3.33) is satisfied, then the basis is called an *orthogonal basis*. Note that (3.34) implies that every basis vector has length/norm 1.

orthonormal basis
ONB
orthogonal basis

Recall from Section 2.6.1 that we can use Gaussian elimination to find a basis for a vector space spanned by a set of vectors. Assume we are given a set $\{\tilde{\boldsymbol{b}}_1, \ldots, \tilde{\boldsymbol{b}}_n\}$ of non-orthogonal and unnormalized basis vectors. We concatenate them into a matrix $\tilde{\boldsymbol{B}} = [\tilde{\boldsymbol{b}}_1, \ldots, \tilde{\boldsymbol{b}}_n]$ and apply Gaussian elimination to the augmented matrix (Section 2.3.2) $[\tilde{\boldsymbol{B}}\tilde{\boldsymbol{B}}^\top | \tilde{\boldsymbol{B}}]$ to obtain an orthonormal basis. This constructive way to iteratively build an orthonormal basis $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ is called the *Gram-Schmidt process* (Strang, 2003).

Example 3.8 (Orthonormal Basis)

The canonical/standard basis for a Euclidean vector space $\mathbb{R}^n$ is an orthonormal basis, where the inner product is the dot product of vectors.

In $\mathbb{R}^2$, the vectors

$$\boldsymbol{b}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{b}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{3.35}$$

form an orthonormal basis since $\boldsymbol{b}_1^\top \boldsymbol{b}_2 = 0$ and $\|\boldsymbol{b}_1\| = 1 = \|\boldsymbol{b}_2\|$.

We will exploit the concept of an orthonormal basis in Chapter 12 and Chapter 10 when we discuss support vector machines and principal component analysis.

3.6 Orthogonal Complement

Having defined orthogonality, we will now look at vector spaces that are orthogonal to each other. This will play an important role in Chapter 10, when we discuss linear dimensionality reduction from a geometric perspective.

Consider a D-dimensional vector space V and an M-dimensional subspace $U \subseteq V$. Then its *orthogonal complement* $U^\perp$ is a $(D-M)$-dimensional subspace of V and contains all vectors in V that are orthogonal to every vector in U. Furthermore, $U \cap U^\perp = \{\boldsymbol{0}\}$ so that any vector $\boldsymbol{x} \in V$ can be

orthogonal complement

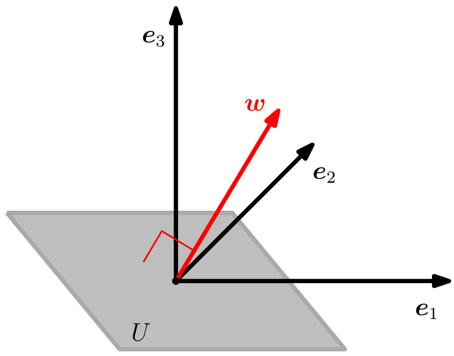


Figure 3.1 A plane U in a three-dimensional vector space can be described by its normal vector, which spans its orthogonal complement $U^{\perp}$.

uniquely decomposed into

$$\boldsymbol{x} = \sum_{m=1}^{M} \lambda_m \boldsymbol{b}_m + \sum_{j=1}^{D-M} \psi_j \boldsymbol{b}_j^{\perp}, \quad \lambda_m,\ \psi_j \in \mathbb{R}, \tag{3.36}$$

where $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_M)$ is a basis of U and $(\boldsymbol{b}_1^{\perp}, \dots, \boldsymbol{b}_{D-M}^{\perp})$ is a basis of $U^{\perp}$.

Therefore, the orthogonal complement can also be used to describe a plane U (two-dimensional subspace) in a three-dimensional vector space. More specifically, the vector $\boldsymbol{w}$ with $\|\boldsymbol{w}\| = 1$, which is orthogonal to the plane U, is the basis vector of $U^{\perp}$. Figure 3.1 illustrates this setting. All vectors that are orthogonal to $\boldsymbol{w}$ must (by construction) lie in the plane U. The vector $\boldsymbol{w}$ is called the *normal vector* of U.

normal vector

Generally, orthogonal complements can be used to describe hyperplanes in n-dimensional vector and affine spaces.

3.7 Inner Product of Functions

Thus far, we looked at properties of inner products to compute lengths, angles and distances. We focused on inner products of finite-dimensional vectors. In the following, we will look at an example of inner products of a different type of vectors: inner products of functions.

The inner products we discussed so far were defined for vectors with a finite number of entries. We can think of a vector $\boldsymbol{x} \in \mathbb{R}^n$ as a function with n function values. The concept of an inner product can be generalized to vectors with an infinite number of entries (countably infinite) and also continuous-valued functions (uncountably infinite). Then the sum over individual components of vectors (see Equation (3.5) for example) turns into an integral.

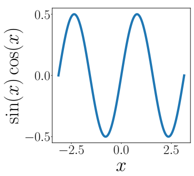
An inner product of two functions $u : \mathbb{R} \to \mathbb{R}$ and $v : \mathbb{R} \to \mathbb{R}$ can be defined as the definite integral

$$\langle u, v \rangle := \int_a^b u(x)v(x)dx \tag{3.37}$$

for lower and upper limits $a, b < \infty$, respectively. As with our usual inner product, we can define norms and orthogonality by looking at the inner product. If (3.37) evaluates to 0, the functions u and v are orthogonal. To make the preceding inner product mathematically precise, we need to take care of measures and the definition of integrals, leading to the definition of a Hilbert space. Furthermore, unlike inner products on finite-dimensional vectors, inner products on functions may diverge (have infinite value). All this requires diving into some more intricate details of real and functional analysis, which we do not cover in this book.

Example 3.9 (Inner Product of Functions)
If we choose $u = \sin(x)$ and $v = \cos(x)$, the integrand $f(x) = u(x)v(x)$ of (3.37), is shown in Figure 3.2. We see that this function is odd, i.e., $f(-x) = -f(x)$. Therefore, the integral with limits $a = -\pi, b = \pi$ of this product evaluates to 0. Therefore, $\sin$ and $\cos$ are orthogonal functions.

Figure 3.2 $f(x) = \sin(x)\cos(x)$.



Remark. It also holds that the collection of functions

$$\{1, \cos(x), \cos(2x), \cos(3x), \dots\} \tag{3.38}$$

is orthogonal if we integrate from $-\pi$ to π, i.e., any pair of functions are orthogonal to each other. The collection of functions in (3.38) spans a large subspace of the functions that are even and periodic on $[-\pi, \pi)$, and projecting functions onto this subspace is the fundamental idea behind Fourier series. ◇

In Section 6.4.6, we will have a look at a second type of unconventional inner products: the inner product of random variables.

3.8 Orthogonal Projections

Projections are an important class of linear transformations (besides rotations and reflections) and play an important role in graphics, coding theory, statistics and machine learning. In machine learning, we often deal with data that is high-dimensional. High-dimensional data is often hard to analyze or visualize. However, high-dimensional data quite often possesses the property that only a few dimensions contain most information, and most other dimensions are not essential to describe key properties of the data. When we compress or visualize high-dimensional data, we will lose information. To minimize this compression loss, we ideally find the most informative dimensions in the data. As discussed in Chapter 1, data can be represented as vectors, and in this chapter, we will discuss some of the fundamental tools for data compression. More specifically, we can project the original high-dimensional data onto a lower-dimensional feature space and work in this lower-dimensional space to learn more about the dataset and extract relevant patterns. For example, machine

"Feature" is a common expression for data representation.

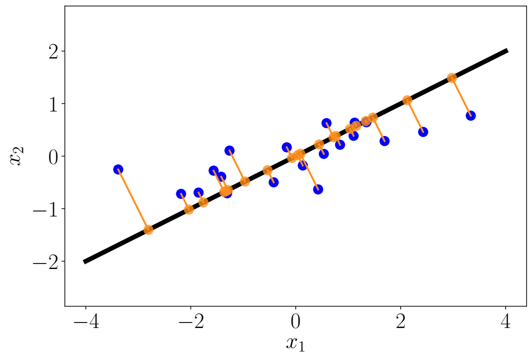


Figure 3.1 Orthogonal projection (orange dots) of a two-dimensional dataset (blue dots) onto a one-dimensional subspace (straight line).

learning algorithms, such as principal component analysis (PCA) by Pearson (1901) and Hotelling (1933) and deep neural networks (e.g., deep auto-encoders (Deng et al., 2010)), heavily exploit the idea of dimensionality reduction. In the following, we will focus on orthogonal projections, which we will use in Chapter 10 for linear dimensionality reduction and in Chapter 12 for classification. Even linear regression, which we discuss in Chapter 9, can be interpreted using orthogonal projections. For a given lower-dimensional subspace, orthogonal projections of high-dimensional data retain as much information as possible and minimize the difference/error between the original data and the corresponding projection. An illustration of such an orthogonal projection is given in Figure 3.1. Before we detail how to obtain these projections, let us define what a projection actually is.

projection

Definition 3.10 (Projection)**.** Let V be a vector space and $U \subseteq V$ a subspace of V. A linear mapping $\pi : V \to U$ is called a *projection* if $\pi^2 = \pi \circ \pi = \pi$.

projection matrix

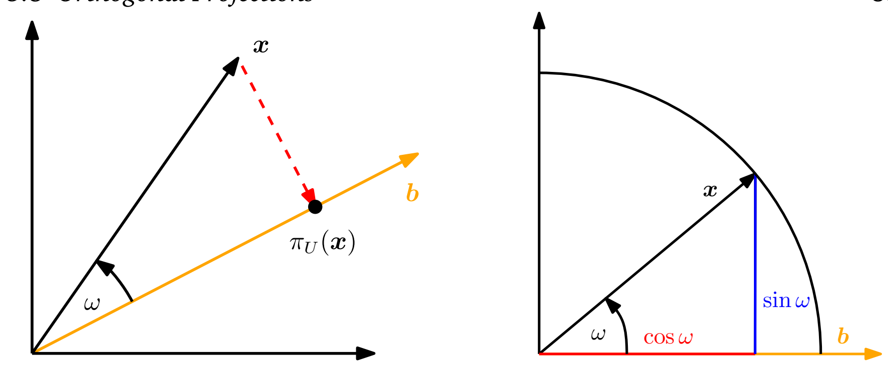
Since linear mappings can be expressed by transformation matrices (see Section 2.7), the preceding definition applies equally to a special kind of transformation matrices, the *projection matrices* $\boldsymbol{P}_\pi$, which exhibit the property that $\boldsymbol{P}_\pi^2 = \boldsymbol{P}_\pi$.

line

In the following, we will derive orthogonal projections of vectors in the inner product space $(\mathbb{R}^n, \langle \cdot, \cdot \rangle)$ onto subspaces. We will start with one-dimensional subspaces, which are also called *lines*. If not mentioned otherwise, we assume the dot product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\top \boldsymbol{y}$ as the inner product.

3.8.1 Projection onto One-Dimensional Subspaces (Lines)

Assume we are given a line (one-dimensional subspace) through the origin with basis vector $\boldsymbol{b} \in \mathbb{R}^n$. The line is a one-dimensional subspace $U \subseteq \mathbb{R}^n$ spanned by $\boldsymbol{b}$. When we project $\boldsymbol{x} \in \mathbb{R}^n$ onto U, we seek the vector $\pi_U(\boldsymbol{x}) \in U$ that is closest to $\boldsymbol{x}$. Using geometric arguments, let us



(a) Projection of $\boldsymbol{x} \in \mathbb{R}^2$ onto a subspace U with basis vector $\boldsymbol{b}$.

(b) Projection of a two-dimensional vector $\boldsymbol{x}$ with $\|\boldsymbol{x}\| = 1$ onto a one-dimensional subspace spanned by $\boldsymbol{b}$.

Figure 3.2 Examples of projections onto one-dimensional subspaces.

characterize some properties of the projection $\pi_U(\boldsymbol{x})$ (Figure 3.2(a) serves as an illustration):

- The projection $\pi_U(\boldsymbol{x})$ is closest to $\boldsymbol{x}$, where "closest" implies that the distance $\|\boldsymbol{x} - \pi_U(\boldsymbol{x})\|$ is minimal. It follows that the segment $\pi_U(\boldsymbol{x}) - \boldsymbol{x}$ from $\pi_U(\boldsymbol{x})$ to $\boldsymbol{x}$ is orthogonal to U, and therefore the basis vector $\boldsymbol{b}$ of U. The orthogonality condition yields $\langle \pi_U(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{b}\rangle = 0$ since angles between vectors are defined via the inner product.
- The projection $\pi_U(\boldsymbol{x})$ of $\boldsymbol{x}$ onto U must be an element of U and, therefore, a multiple of the basis vector $\boldsymbol{b}$ that spans U. Hence, $\pi_U(\boldsymbol{x}) = \lambda \boldsymbol{b}$, for some $\lambda \in \mathbb{R}$.

λ is then the coordinate of $\pi_U(\boldsymbol{x})$ with respect to $\boldsymbol{b}$.

In the following three steps, we determine the coordinate λ, the projection $\pi_U(\boldsymbol{x}) \in U$, and the projection matrix $\boldsymbol{P}_\pi$ that maps any $\boldsymbol{x} \in \mathbb{R}^n$ onto U:

1. Finding the coordinate λ. The orthogonality condition yields

$$\langle \boldsymbol{x} - \pi_U(\boldsymbol{x}), \boldsymbol{b}\rangle = 0 \overset{\pi_U(\boldsymbol{x}) = \lambda \boldsymbol{b}}{\iff} \langle \boldsymbol{x} - \lambda\boldsymbol{b}, \boldsymbol{b}\rangle = 0\,. \tag{3.39}$$

We can now exploit the bilinearity of the inner product and arrive at

With a general inner product, we get $\lambda = \langle \boldsymbol{x}, \boldsymbol{b}\rangle$ if $\|\boldsymbol{b}\| = 1$.

$$\langle \boldsymbol{x}, \boldsymbol{b}\rangle - \lambda \langle \boldsymbol{b}, \boldsymbol{b}\rangle = 0 \iff \lambda = \frac{\langle \boldsymbol{x}, \boldsymbol{b}\rangle}{\langle \boldsymbol{b}, \boldsymbol{b}\rangle} = \frac{\langle \boldsymbol{b}, \boldsymbol{x}\rangle}{\|\boldsymbol{b}\|^2}\,. \tag{3.40}$$

In the last step, we exploited the fact that inner products are symmetric. If we choose $\langle \cdot, \cdot \rangle$ to be the dot product, we obtain

$$\lambda = \frac{\boldsymbol{b}^\top \boldsymbol{x}}{\boldsymbol{b}^\top \boldsymbol{b}} = \frac{\boldsymbol{b}^\top \boldsymbol{x}}{\|\boldsymbol{b}\|^2}\,. \tag{3.41}$$

If $\|\boldsymbol{b}\| = 1$, then the coordinate λ of the projection is given by $\boldsymbol{b}^\top \boldsymbol{x}$.

2. Finding the projection point $\pi_U(\boldsymbol{x}) \in U$. Since $\pi_U(\boldsymbol{x}) = \lambda \boldsymbol{b}$, we immediately obtain with (3.40) that

$$\pi_U(\boldsymbol{x}) = \lambda \boldsymbol{b} = \frac{\langle \boldsymbol{x}, \boldsymbol{b} \rangle}{\|\boldsymbol{b}\|^2} \boldsymbol{b} = \frac{\boldsymbol{b}^\top \boldsymbol{x}}{\|\boldsymbol{b}\|^2} \boldsymbol{b}\,, \tag{3.42}$$

where the last equality holds for the dot product only. We can also compute the length of $\pi_U(\boldsymbol{x})$ by means of Definition 3.1 as

$$\|\pi_U(\boldsymbol{x})\| = \|\lambda \boldsymbol{b}\| = |\lambda|\, \|\boldsymbol{b}\|\,. \tag{3.43}$$

Hence, our projection is of length $|\lambda|$ times the length of $\boldsymbol{b}$. This also adds the intuition that λ is the coordinate of $\pi_U(\boldsymbol{x})$ with respect to the basis vector $\boldsymbol{b}$ that spans our one-dimensional subspace U.

If we use the dot product as an inner product, we get

$$\|\pi_U(\boldsymbol{x})\| \overset{(3.42)}{=} \frac{|\boldsymbol{b}^\top \boldsymbol{x}|}{\|\boldsymbol{b}\|^2} \|\boldsymbol{b}\| \overset{(3.25)}{=} |\cos \omega|\, \|\boldsymbol{x}\|\, \|\boldsymbol{b}\| \frac{\|\boldsymbol{b}\|}{\|\boldsymbol{b}\|^2} = |\cos \omega|\, \|\boldsymbol{x}\|\,. \tag{3.44}$$

Here, ω is the angle between $\boldsymbol{x}$ and $\boldsymbol{b}$. This equation should be familiar from trigonometry: If $\|\boldsymbol{x}\| = 1$, then $\boldsymbol{x}$ lies on the unit circle. It follows that the projection onto the horizontal axis spanned by $\boldsymbol{b}$ is exactly $\cos \omega$, and the length of the corresponding vector $\pi_U(\boldsymbol{x}) = |\cos \omega|$. An illustration is given in Figure 3.2(b).

The horizontal axis is a one-dimensional subspace.

3. Finding the projection matrix $\boldsymbol{P}_\pi$. We know that a projection is a linear mapping (see Definition 3.10). Therefore, there exists a projection matrix $\boldsymbol{P}_\pi$, such that $\pi_U(\boldsymbol{x}) = \boldsymbol{P}_\pi \boldsymbol{x}$. With the dot product as inner product and

$$\pi_U(\boldsymbol{x}) = \lambda \boldsymbol{b} = \boldsymbol{b} \lambda = \boldsymbol{b} \frac{\boldsymbol{b}^\top \boldsymbol{x}}{\|\boldsymbol{b}\|^2} = \frac{\boldsymbol{b}\boldsymbol{b}^\top}{\|\boldsymbol{b}\|^2} \boldsymbol{x}\,, \tag{3.45}$$

we immediately see that

$$\boldsymbol{P}_\pi = \frac{\boldsymbol{b}\boldsymbol{b}^\top}{\|\boldsymbol{b}\|^2}\,. \tag{3.46}$$

Note that $\boldsymbol{b}\boldsymbol{b}^\top$ (and, consequently, $\boldsymbol{P}_\pi$) is a symmetric matrix (of rank 1), and $\|\boldsymbol{b}\|^2 = \langle \boldsymbol{b}, \boldsymbol{b} \rangle$ is a scalar.

Projection matrices are always symmetric.

The projection matrix $\boldsymbol{P}_\pi$ projects any vector $\boldsymbol{x} \in \mathbb{R}^n$ onto the line through the origin with direction $\boldsymbol{b}$ (equivalently, the subspace U spanned by $\boldsymbol{b}$).

Remark. The projection $\pi_U(\boldsymbol{x}) \in \mathbb{R}^n$ is still an n-dimensional vector and not a scalar. However, we no longer require n coordinates to represent the projection, but only a single one if we want to express it with respect to the basis vector $\boldsymbol{b}$ that spans the subspace U: λ. $\diamondsuit$

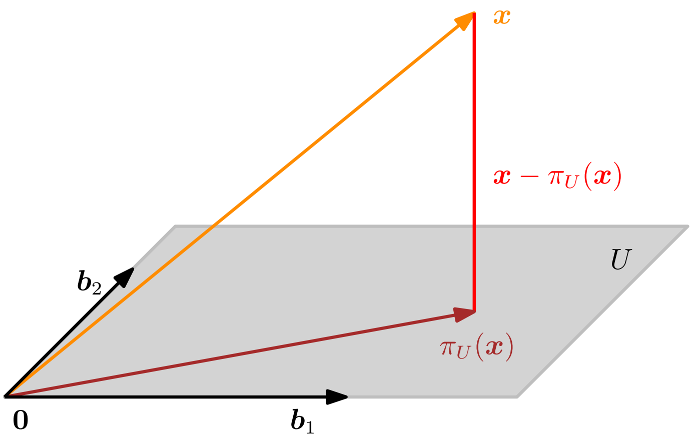


Figure 3.1 Projection onto a two-dimensional subspace U with basis b_1, b_2. The projection $\pi_U(x)$ of $x \in \mathbb{R}^3$ onto U can be expressed as a linear combination of b_1, b_2 and the displacement vector $x - \pi_U(x)$ is orthogonal to both b_1 and b_2.

Example 3.10 (Projection onto a Line)
Find the projection matrix P_π onto the line through the origin spanned by $b = \begin{bmatrix}1 & 2 & 2\end{bmatrix}^\top$. b is a direction and a basis of the one-dimensional subspace (line through origin).

With (3.46), we obtain

$$P_\pi = \frac{bb^\top}{b^\top b} = \frac{1}{9}\begin{bmatrix}1\\2\\2\end{bmatrix}\begin{bmatrix}1 & 2 & 2\end{bmatrix} = \frac{1}{9}\begin{bmatrix}1 & 2 & 2\\2 & 4 & 4\\2 & 4 & 4\end{bmatrix}. \tag{3.47}$$

Let us now choose a particular x and see whether it lies in the subspace spanned by b. For $x = \begin{bmatrix}1 & 1 & 1\end{bmatrix}^\top$, the projection is

$$\pi_U(x) = P_\pi x = \frac{1}{9}\begin{bmatrix}1 & 2 & 2\\2 & 4 & 4\\2 & 4 & 4\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = \frac{1}{9}\begin{bmatrix}5\\10\\10\end{bmatrix} \in \operatorname{span}\left[\begin{bmatrix}1\\2\\2\end{bmatrix}\right]. \tag{3.48}$$

Note that the application of P_π to $\pi_U(x)$ does not change anything, i.e., $P_\pi \pi_U(x) = \pi_U(x)$. This is expected because according to Definition 3.10, we know that a projection matrix P_π satisfies $P_\pi^2 x = P_\pi x$ for all x.

Remark. With the results from Chapter 4, we can show that $\pi_U(x)$ is an eigenvector of P_π, and the corresponding eigenvalue is 1. $\diamondsuit$

3.8.2 Projection onto General Subspaces

In the following, we look at orthogonal projections of vectors $x \in \mathbb{R}^n$ onto lower-dimensional subspaces $U \subseteq \mathbb{R}^n$ with $\dim(U) = m \geqslant 1$. An illustration is given in Figure 3.1.

If U is given by a set of spanning vectors, which are not a basis, make sure you determine a basis $b_1, \dots, b_m$ before proceeding.

Assume that $(b_1, \dots, b_m)$ is an ordered basis of U. Any projection $\pi_U(x)$ onto U is necessarily an element of U. Therefore, they can be represented as linear combinations of the basis vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ of U, such that $\pi_U(\boldsymbol{x}) = \sum_{i=1}^{m} \lambda_i \boldsymbol{b}_i$.

The basis vectors form the columns of $\boldsymbol{B} \in \mathbb{R}^{n \times m}$, where $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m]$.

As in the 1D case, we follow a three-step procedure to find the projection $\pi_U(\boldsymbol{x})$ and the projection matrix $\boldsymbol{P}_\pi$:

1. Find the coordinates $\lambda_1, \ldots, \lambda_m$ of the projection (with respect to the basis of U), such that the linear combination

$$\pi_U(\boldsymbol{x}) = \sum_{i=1}^{m} \lambda_i \boldsymbol{b}_i = \boldsymbol{B}\boldsymbol{\lambda}\,, \tag{3.49}$$

$$\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m] \in \mathbb{R}^{n \times m}, \quad \boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_m]^\top \in \mathbb{R}^m\,, \tag{3.50}$$

is closest to $\boldsymbol{x} \in \mathbb{R}^n$. As in the 1D case, "closest" means "minimum distance", which implies that the vector connecting $\pi_U(\boldsymbol{x}) \in U$ and $\boldsymbol{x} \in \mathbb{R}^n$ must be orthogonal to all basis vectors of U. Therefore, we obtain m simultaneous conditions (assuming the dot product as the inner product)

$$\langle \boldsymbol{b}_1, \boldsymbol{x} - \pi_U(\boldsymbol{x}) \rangle = \boldsymbol{b}_1^\top (\boldsymbol{x} - \pi_U(\boldsymbol{x})) = 0 \tag{3.51}$$

$$\vdots$$

$$\langle \boldsymbol{b}_m, \boldsymbol{x} - \pi_U(\boldsymbol{x}) \rangle = \boldsymbol{b}_m^\top (\boldsymbol{x} - \pi_U(\boldsymbol{x})) = 0 \tag{3.52}$$

which, with $\pi_U(\boldsymbol{x}) = \boldsymbol{B}\boldsymbol{\lambda}$, can be written as

$$\boldsymbol{b}_1^\top (\boldsymbol{x} - \boldsymbol{B}\boldsymbol{\lambda}) = 0 \tag{3.53}$$

$$\vdots$$

$$\boldsymbol{b}_m^\top (\boldsymbol{x} - \boldsymbol{B}\boldsymbol{\lambda}) = 0 \tag{3.54}$$

such that we obtain a homogeneous linear equation system

$$\begin{bmatrix} \boldsymbol{b}_1^\top \\ \vdots \\ \boldsymbol{b}_m^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{x} - \boldsymbol{B}\boldsymbol{\lambda} \end{bmatrix} = \boldsymbol{0} \iff \boldsymbol{B}^\top (\boldsymbol{x} - \boldsymbol{B}\boldsymbol{\lambda}) = \boldsymbol{0} \tag{3.55}$$

$$\iff \boldsymbol{B}^\top \boldsymbol{B}\boldsymbol{\lambda} = \boldsymbol{B}^\top \boldsymbol{x}\,. \tag{3.56}$$

normal equation

The last expression is called *normal equation*. Since $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ are a basis of U and, therefore, linearly independent, $\boldsymbol{B}^\top \boldsymbol{B} \in \mathbb{R}^{m \times m}$ is regular and can be inverted. This allows us to solve for the coefficients/coordinates

$$\boldsymbol{\lambda} = (\boldsymbol{B}^\top \boldsymbol{B})^{-1} \boldsymbol{B}^\top \boldsymbol{x}\,. \tag{3.57}$$

pseudo-inverse

The matrix $(\boldsymbol{B}^\top \boldsymbol{B})^{-1} \boldsymbol{B}^\top$ is also called the *pseudo-inverse* of $\boldsymbol{B}$, which can be computed for non-square matrices $\boldsymbol{B}$. It only requires that $\boldsymbol{B}^\top \boldsymbol{B}$ is positive definite, which is the case if $\boldsymbol{B}$ is full rank. In practical applications (e.g., linear regression), we often add a "jitter term" $\epsilon \boldsymbol{I}$ to

$\boldsymbol{B}^\top\boldsymbol{B}$ to guarantee increased numerical stability and positive definiteness. This "ridge" can be rigorously derived using Bayesian inference. See Chapter 9 for details.

2. Find the projection $\pi_U(\boldsymbol{x}) \in U$. We already established that $\pi_U(\boldsymbol{x}) = \boldsymbol{B}\boldsymbol{\lambda}$. Therefore, with (3.57)

$$\pi_U(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B})^{-1}\boldsymbol{B}^\top\boldsymbol{x}\,. \tag{3.58}$$

3. Find the projection matrix $\boldsymbol{P}_\pi$. From (3.58), we can immediately see that the projection matrix that solves $\boldsymbol{P}_\pi\boldsymbol{x} = \pi_U(\boldsymbol{x})$ must be

$$\boldsymbol{P}_\pi = \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B})^{-1}\boldsymbol{B}^\top\,. \tag{3.59}$$

Remark. The solution for projecting onto general subspaces includes the 1D case as a special case: If $\dim(U) = 1$, then $\boldsymbol{B}^\top\boldsymbol{B} \in \mathbb{R}$ is a scalar and we can rewrite the projection matrix in (3.59) $\boldsymbol{P}_\pi = \boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B})^{-1}\boldsymbol{B}^\top$ as $\boldsymbol{P}_\pi = \frac{\boldsymbol{B}\boldsymbol{B}^\top}{\boldsymbol{B}^\top\boldsymbol{B}}$, which is exactly the projection matrix in (3.46). ♢

Example 3.11 (Projection onto a Two-dimensional Subspace)

For a subspace $U = \mathrm{span}[\begin{bmatrix}1\\1\\1\end{bmatrix}, \begin{bmatrix}0\\1\\2\end{bmatrix}] \subseteq \mathbb{R}^3$ and $\boldsymbol{x} = \begin{bmatrix}6\\0\\0\end{bmatrix} \in \mathbb{R}^3$ find the coordinates $\boldsymbol{\lambda}$ of $\boldsymbol{x}$ in terms of the subspace U, the projection point $\pi_U(\boldsymbol{x})$ and the projection matrix $\boldsymbol{P}_\pi$.

First, we see that the generating set of U is a basis (linear independence) and write the basis vectors of U into a matrix $\boldsymbol{B} = \begin{bmatrix}1 & 0\\1 & 1\\1 & 2\end{bmatrix}$.

Second, we compute the matrix $\boldsymbol{B}^\top\boldsymbol{B}$ and the vector $\boldsymbol{B}^\top\boldsymbol{x}$ as

$$\boldsymbol{B}^\top\boldsymbol{B} = \begin{bmatrix}1 & 1 & 1\\0 & 1 & 2\end{bmatrix}\begin{bmatrix}1 & 0\\1 & 1\\1 & 2\end{bmatrix} = \begin{bmatrix}3 & 3\\3 & 5\end{bmatrix}, \quad \boldsymbol{B}^\top\boldsymbol{x} = \begin{bmatrix}1 & 1 & 1\\0 & 1 & 2\end{bmatrix}\begin{bmatrix}6\\0\\0\end{bmatrix} = \begin{bmatrix}6\\0\end{bmatrix}. \tag{3.60}$$

Third, we solve the normal equation $\boldsymbol{B}^\top\boldsymbol{B}\boldsymbol{\lambda} = \boldsymbol{B}^\top\boldsymbol{x}$ to find $\boldsymbol{\lambda}$:

$$\begin{bmatrix}3 & 3\\3 & 5\end{bmatrix}\begin{bmatrix}\lambda_1\\\lambda_2\end{bmatrix} = \begin{bmatrix}6\\0\end{bmatrix} \iff \boldsymbol{\lambda} = \begin{bmatrix}5\\-3\end{bmatrix}. \tag{3.61}$$

Fourth, the projection $\pi_U(\boldsymbol{x})$ of $\boldsymbol{x}$ onto U, i.e., into the column space of $\boldsymbol{B}$, can be directly computed via

$$\pi_U(\boldsymbol{x}) = \boldsymbol{B}\boldsymbol{\lambda} = \begin{bmatrix}5\\2\\-1\end{bmatrix}. \tag{3.62}$$

projection error

The projection error is also called the *reconstruction error*.

The corresponding *projection error* is the norm of the difference vector between the original vector and its projection onto U, i.e.,

$$\|\boldsymbol{x} - \pi_U(\boldsymbol{x})\| = \left\| \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^\top \right\| = \sqrt{6}\,. \tag{3.63}$$

Fifth, the projection matrix (for any $\boldsymbol{x} \in \mathbb{R}^3$) is given by

$$\boldsymbol{P}_\pi = \boldsymbol{B}(\boldsymbol{B}^\top \boldsymbol{B})^{-1}\boldsymbol{B}^\top = \frac{1}{6}\begin{bmatrix} 5 & 2 & -1 \\ 2 & 2 & 2 \\ -1 & 2 & 5 \end{bmatrix}. \tag{3.64}$$

To verify the results, we can (a) check whether the displacement vector $\pi_U(\boldsymbol{x}) - \boldsymbol{x}$ is orthogonal to all basis vectors of U, and (b) verify that $\boldsymbol{P}_\pi = \boldsymbol{P}_\pi^2$ (see Definition 3.10).

Remark. The projections $\pi_U(\boldsymbol{x})$ are still vectors in $\mathbb{R}^n$ although they lie in an m-dimensional subspace $U \subseteq \mathbb{R}^n$. However, to represent a projected vector we only need the m coordinates $\lambda_1, \ldots, \lambda_m$ with respect to the basis vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ of U. ♢

Remark. In vector spaces with general inner products, we have to pay attention when computing angles and distances, which are defined by means of the inner product. ♢

We can find approximate solutions to unsolvable linear equation systems using projections.

Projections allow us to look at situations where we have a linear system $\boldsymbol{Ax} = \boldsymbol{b}$ without a solution. Recall that this means that $\boldsymbol{b}$ does not lie in the span of $\boldsymbol{A}$, i.e., the vector $\boldsymbol{b}$ does not lie in the subspace spanned by the columns of $\boldsymbol{A}$. Given that the linear equation cannot be solved exactly, we can find an *approximate solution*. The idea is to find the vector in the subspace spanned by the columns of $\boldsymbol{A}$ that is closest to $\boldsymbol{b}$, i.e., we compute the orthogonal projection of $\boldsymbol{b}$ onto the subspace spanned by the columns of $\boldsymbol{A}$. This problem arises often in practice, and the solution is called the *least-squares solution* (assuming the dot product as the inner product) of an overdetermined system. This is discussed further in Section 9.4. Using reconstruction errors (3.63) is one possible approach to derive principal component analysis (Section 10.3).

least-squares solution

Remark. We just looked at projections of vectors $\boldsymbol{x}$ onto a subspace U with basis vectors $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$. If this basis is an ONB, i.e., (3.33) and (3.34) are satisfied, the projection equation (3.58) simplifies greatly to

$$\pi_U(\boldsymbol{x}) = \boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{x} \tag{3.65}$$

since $\boldsymbol{B}^\top \boldsymbol{B} = \boldsymbol{I}$ with coordinates

$$\boldsymbol{\lambda} = \boldsymbol{B}^\top \boldsymbol{x}\,. \tag{3.66}$$

This means that we no longer have to compute the inverse from (3.58), which saves computation time. ♢

3.8.3 Gram-Schmidt Orthogonalization

Projections are at the core of the Gram-Schmidt method that allows us to constructively transform any basis $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_n)$ of an n-dimensional vector space V into an orthogonal/orthonormal basis $(\boldsymbol{u}_1, \dots, \boldsymbol{u}_n)$ of V. This basis always exists (Liesen and Mehrmann, 2015) and $\mathrm{span}[\boldsymbol{b}_1, \dots, \boldsymbol{b}_n] = \mathrm{span}[\boldsymbol{u}_1, \dots, \boldsymbol{u}_n]$. The *Gram-Schmidt orthogonalization* method iteratively constructs an orthogonal basis $(\boldsymbol{u}_1, \dots, \boldsymbol{u}_n)$ from any basis $(\boldsymbol{b}_1, \dots, \boldsymbol{b}_n)$ of V as follows:

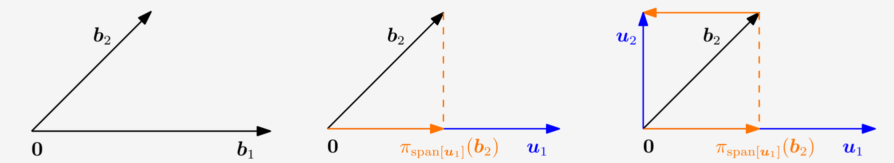
Gram-Schmidt orthogonalization

$$\boldsymbol{u}_1 := \boldsymbol{b}_1 \tag{3.67}$$

$$\boldsymbol{u}_k := \boldsymbol{b}_k - \pi_{\mathrm{span}[\boldsymbol{u}_1, \dots, \boldsymbol{u}_{k-1}]}(\boldsymbol{b}_k)\,, \quad k = 2, \dots, n\,. \tag{3.68}$$

In (3.68), the kth basis vector $\boldsymbol{b}_k$ is projected onto the subspace spanned by the first $k-1$ constructed orthogonal vectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_{k-1}$; see Section 3.8.2. This projection is then subtracted from $\boldsymbol{b}_k$ and yields a vector $\boldsymbol{u}_k$ that is orthogonal to the $(k-1)$-dimensional subspace spanned by $\boldsymbol{u}_1, \dots, \boldsymbol{u}_{k-1}$. Repeating this procedure for all n basis vectors $\boldsymbol{b}_1, \dots, \boldsymbol{b}_n$ yields an orthogonal basis $(\boldsymbol{u}_1, \dots, \boldsymbol{u}_n)$ of V. If we normalize the $\boldsymbol{u}_k$, we obtain an ONB where $\|\boldsymbol{u}_k\| = 1$ for $k = 1, \dots, n$.

Example 3.12 (Gram-Schmidt Orthogonalization)



(a) Original non-orthogonal basis vectors $\boldsymbol{b}_1, \boldsymbol{b}_2$.

(b) First new basis vector $\boldsymbol{u}_1 = \boldsymbol{b}_1$ and projection of $\boldsymbol{b}_2$ onto the subspace spanned by $\boldsymbol{u}_1$.

(c) Orthogonal basis vectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2 = \boldsymbol{b}_2 - \pi_{\mathrm{span}[\boldsymbol{u}_1]}(\boldsymbol{b}_2)$.

Figure 3.2 Gram-Schmidt orthogonalization. (a) non-orthogonal basis $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ of $\mathbb{R}^2$; (b) first constructed basis vector $\boldsymbol{u}_1$ and orthogonal projection of $\boldsymbol{b}_2$ onto $\mathrm{span}[\boldsymbol{u}_1]$; (c) orthogonal basis $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ of $\mathbb{R}^2$.

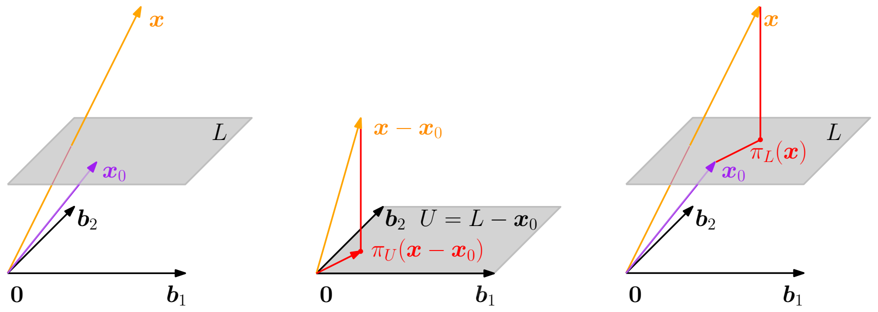
Consider a basis $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ of $\mathbb{R}^2$, where

$$\boldsymbol{b}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad \boldsymbol{b}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \tag{3.69}$$

see also Figure 3.2(a). Using the Gram-Schmidt method, we construct an orthogonal basis $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ of $\mathbb{R}^2$ as follows (assuming the dot product as the inner product):

$$\boldsymbol{u}_1 := \boldsymbol{b}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \tag{3.70}$$

$$\boldsymbol{u}_2 := \boldsymbol{b}_2 - \pi_{\mathrm{span}[\boldsymbol{u}_1]}(\boldsymbol{b}_2) \overset{(3.45)}{=} \boldsymbol{b}_2 - \frac{\boldsymbol{u}_1 \boldsymbol{u}_1^\top}{\|\boldsymbol{u}_1\|^2} \boldsymbol{b}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.71}$$



(a) Setting.

(b) Reduce problem to projection π_U onto vector subspace.

(c) Add support point back in to get affine projection π_L.

Figure 3.3 Projection onto an affine space. (a) original setting; (b) setting shifted by $-\boldsymbol{x}_0$ so that $\boldsymbol{x}-\boldsymbol{x}_0$ can be projected onto the direction space U; (c) projection is translated back to $\boldsymbol{x}_0 + \pi_U(\boldsymbol{x}-\boldsymbol{x}_0)$, which gives the final orthogonal projection $\pi_L(\boldsymbol{x})$.

These steps are illustrated in Figures 3.2(b) and (c). We immediately see that $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are orthogonal, i.e., $\boldsymbol{u}_1^\top \boldsymbol{u}_2 = 0$.

3.8.4 Projection onto Affine Subspaces

Thus far, we discussed how to project a vector onto a lower-dimensional subspace U. In the following, we provide a solution to projecting a vector onto an affine subspace.

Consider the setting in Figure 3.3(a). We are given an affine space $L = \boldsymbol{x}_0 + U$, where $\boldsymbol{b}_1, \boldsymbol{b}_2$ are basis vectors of U. To determine the orthogonal projection $\pi_L(\boldsymbol{x})$ of $\boldsymbol{x}$ onto L, we transform the problem into a problem that we know how to solve: the projection onto a vector subspace. In order to get there, we subtract the support point $\boldsymbol{x}_0$ from $\boldsymbol{x}$ and from L, so that $L - \boldsymbol{x}_0 = U$ is exactly the vector subspace U. We can now use the orthogonal projections onto a subspace we discussed in Section 3.8.2 and obtain the projection $\pi_U(\boldsymbol{x}-\boldsymbol{x}_0)$, which is illustrated in Figure 3.3(b). This projection can now be translated back into L by adding $\boldsymbol{x}_0$, such that we obtain the orthogonal projection onto an affine space L as

$$\pi_L(\boldsymbol{x}) = \boldsymbol{x}_0 + \pi_U(\boldsymbol{x}-\boldsymbol{x}_0)\,, \tag{3.72}$$

where $\pi_U(\cdot)$ is the orthogonal projection onto the subspace U, i.e., the direction space of L; see Figure 3.3(c).

From Figure 3.3, it is also evident that the distance of $\boldsymbol{x}$ from the affine space L is identical to the distance of $\boldsymbol{x}-\boldsymbol{x}_0$ from U, i.e.,

$$d(\boldsymbol{x}, L) = \|\boldsymbol{x} - \pi_L(\boldsymbol{x})\| = \|\boldsymbol{x} - (\boldsymbol{x}_0 + \pi_U(\boldsymbol{x}-\boldsymbol{x}_0))\| \tag{3.73a}$$

$$= d(\boldsymbol{x}-\boldsymbol{x}_0, \pi_U(\boldsymbol{x}-\boldsymbol{x}_0)) = d(\boldsymbol{x}-\boldsymbol{x}_0, U)\,. \tag{3.73b}$$

We will use projections onto an affine subspace to derive the concept of a separating hyperplane in Section 12.1.

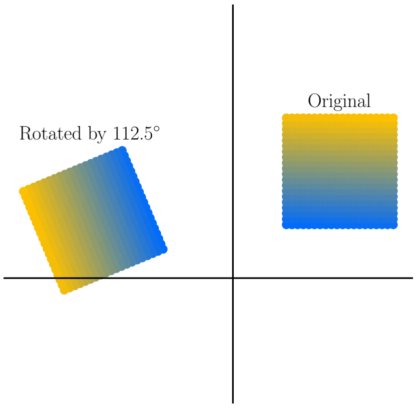


Figure 3.2 A rotation rotates objects in a plane about the origin. If the rotation angle is positive, we rotate counterclockwise.

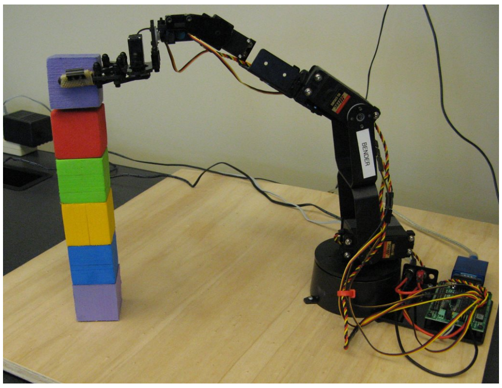


Figure 3.1 The robotic arm needs to rotate its joints in order to pick up objects or to place them correctly. Figure taken from (Deisenroth et al., 2015).

3.9 Rotations

Length and angle preservation, as discussed in Section 3.4, are the two characteristics of linear mappings with orthogonal transformation matrices. In the following, we will have a closer look at specific orthogonal transformation matrices, which describe rotations.

A *rotation* is a linear mapping (more specifically, an automorphism of a Euclidean vector space) that rotates a plane by an angle θ about the origin, i.e., the origin is a fixed point. For a positive angle $\theta > 0$, by common convention, we rotate in a counterclockwise direction. An example is shown in Figure 3.2, where the transformation matrix is

rotation

$$\boldsymbol{R} = \begin{bmatrix} -0.38 & -0.92 \\ 0.92 & -0.38 \end{bmatrix} . \tag{3.74}$$

Important application areas of rotations include computer graphics and robotics. For example, in robotics, it is often important to know how to rotate the joints of a robotic arm in order to pick up or place an object, see Figure 3.1.

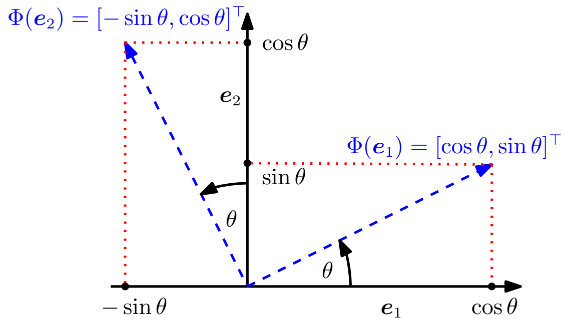


Figure 3.2 Rotation of the standard basis in $\mathbb{R}^2$ by an angle θ.

3.9.1 Rotations in $\mathbb{R}^2$

Consider the standard basis $\left\{ \boldsymbol{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$ of $\mathbb{R}^2$, which defines the standard coordinate system in $\mathbb{R}^2$. We aim to rotate this coordinate system by an angle θ as illustrated in Figure 3.2. Note that the rotated vectors are still linearly independent and, therefore, are a basis of $\mathbb{R}^2$. This means that the rotation performs a basis change.

Rotations Φ are linear mappings so that we can express them by a *rotation matrix* $\boldsymbol{R}(\theta)$. Trigonometry (see Figure 3.2) allows us to determine the coordinates of the rotated axes (the image of Φ) with respect to the standard basis in $\mathbb{R}^2$. We obtain

rotation matrix

$$\Phi(\boldsymbol{e}_1) = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}, \quad \Phi(\boldsymbol{e}_2) = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix}. \tag{3.75}$$

Therefore, the rotation matrix that performs the basis change into the rotated coordinates $\boldsymbol{R}(\theta)$ is given as

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \Phi(\boldsymbol{e}_1) & \Phi(\boldsymbol{e}_2) \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{3.76}$$

3.9.2 Rotations in $\mathbb{R}^3$

In contrast to the $\mathbb{R}^2$ case, in $\mathbb{R}^3$ we can rotate any two-dimensional plane about a one-dimensional axis. The easiest way to specify the general rotation matrix is to specify how the images of the standard basis $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ are supposed to be rotated, and making sure these images $\boldsymbol{Re}_1, \boldsymbol{Re}_2, \boldsymbol{Re}_3$ are orthonormal to each other. We can then obtain a general rotation matrix $\boldsymbol{R}$ by combining the images of the standard basis.

To have a meaningful rotation angle, we have to define what "counterclockwise" means when we operate in more than two dimensions. We use the convention that a "counterclockwise" (planar) rotation about an axis refers to a rotation about an axis when we look at the axis "head on, from the end toward the origin". In $\mathbb{R}^3$, there are therefore three (planar) rotations about the three standard basis vectors (see Figure 3.2):

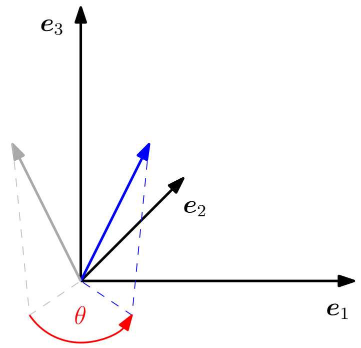


Figure 3.2 Rotation of a vector (gray) in $\mathbb{R}^3$ by an angle θ about the e_3-axis. The rotated vector is shown in blue.

- Rotation about the e_1-axis

$$\boldsymbol{R}_1(\theta) = \begin{bmatrix}\Phi(\boldsymbol{e}_1) & \Phi(\boldsymbol{e}_2) & \Phi(\boldsymbol{e}_3)\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & \cos\theta & -\sin\theta\\ 0 & \sin\theta & \cos\theta\end{bmatrix}. \tag{3.77}$$

  Here, the $\boldsymbol{e}_1$ coordinate is fixed, and the counterclockwise rotation is performed in the $\boldsymbol{e}_2\boldsymbol{e}_3$ plane.

- Rotation about the $\boldsymbol{e}_2$-axis

$$\boldsymbol{R}_2(\theta) = \begin{bmatrix}\cos\theta & 0 & \sin\theta\\ 0 & 1 & 0\\ -\sin\theta & 0 & \cos\theta\end{bmatrix}. \tag{3.78}$$

  If we rotate the $\boldsymbol{e}_1\boldsymbol{e}_3$ plane about the $\boldsymbol{e}_2$ axis, we need to look at the $\boldsymbol{e}_2$ axis from its “tip” toward the origin.

- Rotation about the $\boldsymbol{e}_3$-axis

$$\boldsymbol{R}_3(\theta) = \begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}. \tag{3.79}$$

  Figure 3.2 illustrates this.

3.9.3 Rotations in n Dimensions

The generalization of rotations from 2D and 3D to n-dimensional Euclidean vector spaces can be intuitively described as fixing $n-2$ dimensions and restrict the rotation to a two-dimensional plane in the n-dimensional space. As in the three-dimensional case, we can rotate any plane (two-dimensional subspace of $\mathbb{R}^n$).

Definition 3.11 (Givens Rotation)**.** Let V be an n-dimensional Euclidean vector space and $\Phi : V \to V$ an automorphism with transformation matrix

$$\boldsymbol{R}_{ij}(\theta) := \begin{bmatrix} \boldsymbol{I}_{i-1} & \boldsymbol{0} & \cdots & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \cos\theta & \boldsymbol{0} & -\sin\theta & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{j-i-1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \sin\theta & \boldsymbol{0} & \cos\theta & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \cdots & \boldsymbol{0} & \boldsymbol{I}_{n-j} \end{bmatrix} \in \mathbb{R}^{n\times n}\,, \tag{3.80}$$

Givens rotation

for $1 \leqslant i < j \leqslant n$ and $\theta \in \mathbb{R}$. Then $\boldsymbol{R}_{ij}(\theta)$ is called a *Givens rotation*. Essentially, $\boldsymbol{R}_{ij}(\theta)$ is the identity matrix $\boldsymbol{I}_n$ with

$$r_{ii} = \cos\theta\,, \quad r_{ij} = -\sin\theta\,, \quad r_{ji} = \sin\theta\,, \quad r_{jj} = \cos\theta\,. \tag{3.81}$$

In two dimensions (i.e., $n = 2$), we obtain (3.76) as a special case.

3.9.4 Properties of Rotations

Rotations exhibit a number of useful properties, which can be derived by considering them as orthogonal matrices (Definition 3.8):

- Rotations preserve distances, i.e., $\|\boldsymbol{x}-\boldsymbol{y}\| = \|\boldsymbol{R}_\theta(\boldsymbol{x})-\boldsymbol{R}_\theta(\boldsymbol{y})\|$. In other words, rotations leave the distance between any two points unchanged after the transformation.
- Rotations preserve angles, i.e., the angle between $\boldsymbol{R}_\theta\boldsymbol{x}$ and $\boldsymbol{R}_\theta\boldsymbol{y}$ equals the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$.
- Rotations in three (or more) dimensions are generally not commutative. Therefore, the order in which rotations are applied is important, even if they rotate about the same point. Only in two dimensions vector rotations are commutative, such that $\boldsymbol{R}(\phi)\boldsymbol{R}(\theta) = \boldsymbol{R}(\theta)\boldsymbol{R}(\phi)$ for all $\phi, \theta \in [0, 2\pi)$. They form an Abelian group (with multiplication) only if they rotate about the same point (e.g., the origin).

3.10 Further Reading

In this chapter, we gave a brief overview of some of the important concepts of analytic geometry, which we will use in later chapters of the book. For a broader and more in-depth overview of some of the concepts we presented, we refer to the following excellent books: Axler (2015) and Boyd and Vandenberghe (2018).

Inner products allow us to determine specific bases of vector (sub)spaces, where each vector is orthogonal to all others (orthogonal bases) using the Gram-Schmidt method. These bases are important in optimization and numerical algorithms for solving linear equation systems. For instance, Krylov subspace methods, such as conjugate gradients or the generalized minimal residual method (GMRES), minimize residual errors that are orthogonal to each other (Stoer and Burlirsch, 2002).

In machine learning, inner products are important in the context of kernel methods (Schölkopf and Smola, 2002). Kernel methods exploit the fact that many linear algorithms can be expressed purely by inner product computations. Then, the "kernel trick" allows us to compute these inner products implicitly in a (potentially infinite-dimensional) feature space, without even knowing this feature space explicitly. This allowed the "non-linearization" of many algorithms used in machine learning, such as kernel-PCA (Schölkopf et al., 1997) for dimensionality reduction. Gaussian processes (Rasmussen and Williams, 2006) also fall into the category of kernel methods and are the current state of the art in probabilistic regression (fitting curves to data points). The idea of kernels is explored further in Chapter 12.

Projections are often used in computer graphics, e.g., to generate shadows. In optimization, orthogonal projections are often used to (iteratively) minimize residual errors. This also has applications in machine learning, e.g., in linear regression where we want to find a (linear) function that minimizes the residual errors, i.e., the lengths of the orthogonal projections of the data onto the linear function (Bishop, 2006). We will investigate this further in Chapter 9. PCA (Pearson, 1901; Hotelling, 1933) also uses projections to reduce the dimensionality of high-dimensional data. We will discuss this in more detail in Chapter 10.

Exercises

3.1 Show that $\langle \cdot, \cdot \rangle$ defined for all $\boldsymbol{x} = [x_1, x_2]^\top \in \mathbb{R}^2$ and $\boldsymbol{y} = [y_1, y_2]^\top \in \mathbb{R}^2$ by

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle := x_1 y_1 - (x_1 y_2 + x_2 y_1) + 2(x_2 y_2)$$

is an inner product.

3.2 Consider $\mathbb{R}^2$ with $\langle \cdot, \cdot \rangle$ defined for all $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}^2$ as

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \underbrace{\begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix}}_{=:\boldsymbol{A}} \boldsymbol{y}\,.$$

Is $\langle \cdot, \cdot \rangle$ an inner product?

3.3 Compute the distance between

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix}$$

using

a. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{y}$

b. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{y}\,, \quad \boldsymbol{A} := \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}$

3.4 Compute the angle between

$$\boldsymbol{x} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

using

a. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{y}$

b. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{B} \boldsymbol{y}\,, \quad \boldsymbol{B} := \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$

3.5 Consider the Euclidean vector space $\mathbb{R}^5$ with the dot product. A subspace $U \subseteq \mathbb{R}^5$ and $\boldsymbol{x} \in \mathbb{R}^5$ are given by

$$U = \mathrm{span}[\begin{bmatrix} 0 \\ -1 \\ 2 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ -3 \\ 1 \\ -1 \\ 2 \end{bmatrix}, \begin{bmatrix} -3 \\ 4 \\ 1 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ -3 \\ 5 \\ 0 \\ 7 \end{bmatrix}], \quad \boldsymbol{x} = \begin{bmatrix} -1 \\ -9 \\ -1 \\ 4 \\ 1 \end{bmatrix}.$$

a. Determine the orthogonal projection $\pi_U(\boldsymbol{x})$ of $\boldsymbol{x}$ onto U

b. Determine the distance $d(\boldsymbol{x}, U)$

3.6 Consider $\mathbb{R}^3$ with the inner product

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \boldsymbol{y}\,.$$

Furthermore, we define $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ as the standard/canonical basis in $\mathbb{R}^3$.

a. Determine the orthogonal projection $\pi_U(\boldsymbol{e}_2)$ of $\boldsymbol{e}_2$ onto

$$U = \mathrm{span}[\boldsymbol{e}_1, \boldsymbol{e}_3]\,.$$

Hint: Orthogonality is defined through the inner product.

b. Compute the distance $d(\boldsymbol{e}_2, U)$.

c. Draw the scenario: standard basis vectors and $\pi_U(\boldsymbol{e}_2)$

3.7 Let V be a vector space and π an endomorphism of V.

a. Prove that π is a projection if and only if $\mathrm{id}_V - \pi$ is a projection, where id_V is the identity endomorphism on V.

b. Assume now that π is a projection. Calculate $\mathrm{Im}(\mathrm{id}_V - \pi)$ and $\ker(\mathrm{id}_V - \pi)$ as a function of $\mathrm{Im}(\pi)$ and $\ker(\pi)$.

3.8 Using the Gram-Schmidt method, turn the basis $B = (\boldsymbol{b}_1, \boldsymbol{b}_2)$ of a two-dimensional subspace $U \subseteq \mathbb{R}^3$ into an ONB $C = (\boldsymbol{c}_1, \boldsymbol{c}_2)$ of U, where

$$\boldsymbol{b}_1 := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{b}_2 := \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix}.$$

3.9 Let $n \in \mathbb{N}$ and let $x_1, \ldots, x_n > 0$ be n positive real numbers so that $x_1 + \ldots + x_n = 1$. Use the Cauchy-Schwarz inequality and show that

a. $\sum_{i=1}^{n} x_i^2 \geqslant \frac{1}{n}$

b. $\sum_{i=1}^{n} \frac{1}{x_i} \geqslant n^2$

Hint: Think about the dot product on $\mathbb{R}^n$. Then, choose specific vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and apply the Cauchy-Schwarz inequality.

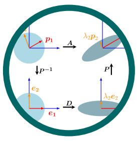
3.10 Rotate the vectors

$$\boldsymbol{x}_1 := \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \boldsymbol{x}_2 := \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

by $30°$.

4

Matrix Decompositions



In Chapters 2 and 3, we studied ways to manipulate and measure vectors, projections of vectors, and linear mappings. Mappings and transformations of vectors can be conveniently described as operations performed by matrices. Moreover, data is often represented in matrix form as well, e.g., where the rows of the matrix represent different people and the columns describe different features of the people, such as weight, height, and socioeconomic status. In this chapter, we present three aspects of matrices: how to summarize matrices, how matrices can be decomposed, and how these decompositions can be used for matrix approximations.

We first consider methods that allow us to describe matrices with just a few numbers that characterize the overall properties of matrices. We will do this in the sections on determinants (Section 4.1) and eigenvalues (Section 4.2) for the important special case of square matrices. These characteristic numbers have important mathematical consequences and allow us to quickly grasp what useful properties a matrix has. From here we will proceed to matrix decomposition methods: An analogy for matrix decomposition is the factoring of numbers, such as the factoring of 21 into prime numbers $7 \cdot 3$. For this reason matrix decomposition is also often referred to as *matrix factorization*. Matrix decompositions are used to describe a matrix by means of a different representation using factors of interpretable matrices.

matrix factorization

We will first cover a square-root-like operation for symmetric, positive definite matrices, the Cholesky decomposition (Section 4.3). From here we will look at two related methods for factorizing matrices into canonical forms. The first one is known as matrix diagonalization (Section 4.4), which allows us to represent the linear mapping using a diagonal transformation matrix if we choose an appropriate basis. The second method, singular value decomposition (Section 4.5), extends this factorization to non-square matrices, and it is considered one of the fundamental concepts in linear algebra. These decompositions are helpful, as matrices representing numerical data are often very large and hard to analyze. We conclude the chapter with a systematic overview of the types of matrices and the characteristic properties that distinguish them in the form of a matrix taxonomy (Section 4.7).

The methods that we cover in this chapter will become important in

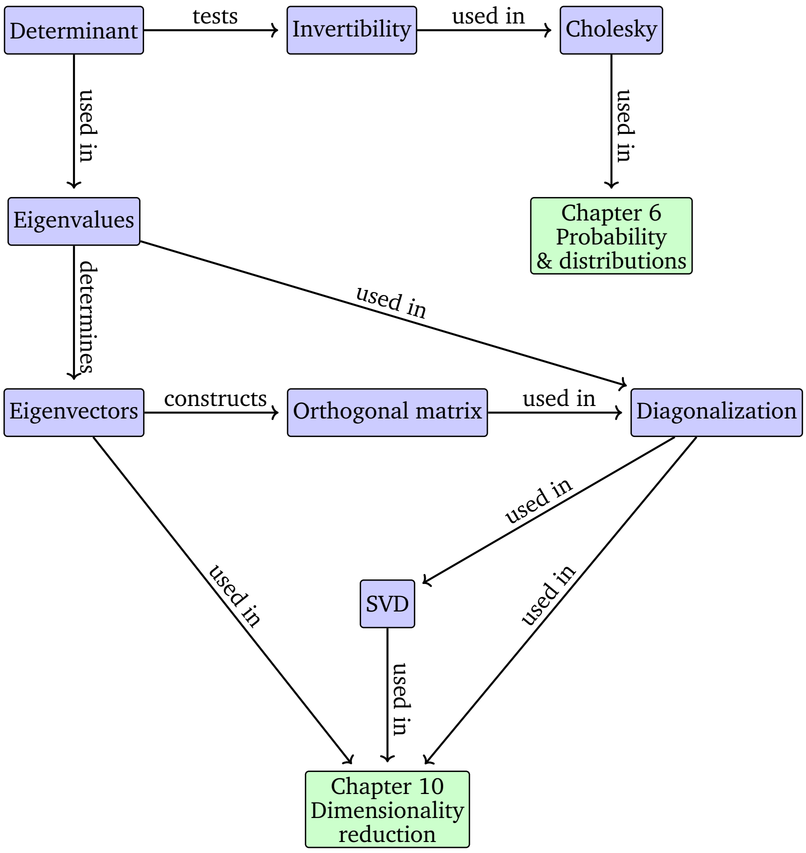


Figure 4.2 A mind map of the concepts introduced in this chapter, along with where they are used in other parts of the book.

both subsequent mathematical chapters, such as Chapter 6, but also in applied chapters, such as dimensionality reduction in Chapters 10 or density estimation in Chapter 11. This chapter's overall structure is depicted in the mind map of Figure 4.2.

4.1 Determinant and Trace

The determinant notation $|\boldsymbol{A}|$ must not be confused with the absolute value.

Determinants are important concepts in linear algebra. A determinant is a mathematical object in the analysis and solution of systems of linear equations. Determinants are only defined for square matrices $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, i.e., matrices with the same number of rows and columns. In this book, we write the determinant as $\det(\boldsymbol{A})$ or sometimes as $|\boldsymbol{A}|$ so that

$$\det(\boldsymbol{A}) = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} . \tag{4.1}$$

The *determinant* of a square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is a function that maps $\boldsymbol{A}$

determinant

onto a real number. Before providing a definition of the determinant for general $n \times n$ matrices, let us have a look at some motivating examples, and define determinants for some special matrices.

Example 4.1 (Testing for Matrix Invertibility)
Let us begin with exploring if a square matrix $\boldsymbol{A}$ is invertible (see Section 2.2.2). For the smallest cases, we already know when a matrix is invertible. If $\boldsymbol{A}$ is a 1×1 matrix, i.e., it is a scalar number, then $\boldsymbol{A} = a \implies \boldsymbol{A}^{-1} = \frac{1}{a}$. Thus $a\,\frac{1}{a} = 1$ holds, if and only if $a \neq 0$.

For 2×2 matrices, by the definition of the inverse (Definition 2.3), we know that $\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}$. Then, with (2.24), the inverse of $\boldsymbol{A}$ is

$$\boldsymbol{A}^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}. \tag{4.2}$$

Hence, $\boldsymbol{A}$ is invertible if and only if

$$a_{11}a_{22} - a_{12}a_{21} \neq 0\,. \tag{4.3}$$

This quantity is the determinant of $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$, i.e.,

$$\det(\boldsymbol{A}) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}\,. \tag{4.4}$$

Example 4.1 points already at the relationship between determinants and the existence of inverse matrices. The next theorem states the same result for $n \times n$ matrices.

Theorem 4.1. *For any square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ it holds that $\boldsymbol{A}$ is invertible if and only if $\det(\boldsymbol{A}) \neq 0$.*

We have explicit (closed-form) expressions for determinants of small matrices in terms of the elements of the matrix. For $n = 1$,

$$\det(\boldsymbol{A}) = \det(a_{11}) = a_{11}\,. \tag{4.5}$$

For $n = 2$,

$$\det(\boldsymbol{A}) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}\,, \tag{4.6}$$

which we have observed in the preceding example.

For $n = 3$ (known as Sarrus' rule),

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{21}a_{32}a_{13} + a_{31}a_{12}a_{23} \tag{4.7}$$
$$- a_{31}a_{22}a_{13} - a_{11}a_{32}a_{23} - a_{21}a_{12}a_{33}\,.$$

For a memory aid of the product terms in Sarrus' rule, try tracing the elements of the triple products in the matrix.

We call a square matrix $\boldsymbol{T}$ an *upper-triangular matrix* if $T_{ij} = 0$ for $i > j$, i.e., the matrix is zero below its diagonal. Analogously, we define a *lower-triangular matrix* as a matrix with zeros above its diagonal. For a triangular matrix $\boldsymbol{T} \in \mathbb{R}^{n\times n}$, the determinant is the product of the diagonal elements, i.e.,

upper-triangular matrix

lower-triangular matrix

$$\det(\boldsymbol{T}) = \prod_{i=1}^{n} T_{ii} \,. \tag{4.8}$$

The determinant is the signed volume of the parallelepiped formed by the columns of the matrix.

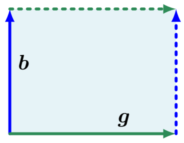


Figure 4.3 The area of the parallelogram (shaded region) spanned by the vectors $\boldsymbol{b}$ and $\boldsymbol{g}$ is $|\det([\boldsymbol{b}, \ \boldsymbol{g}])|$.

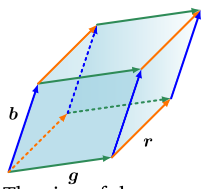


Figure 4.4 The volume of the parallelepiped (shaded volume) spanned by vectors $\boldsymbol{r}, \boldsymbol{b}, \boldsymbol{g}$ is $|\det([\boldsymbol{r}, \ \boldsymbol{b}, \ \boldsymbol{g}])|$.

Example 4.2 (Determinants as Measures of Volume)

The notion of a determinant is natural when we consider it as a mapping from a set of n vectors spanning an object in $\mathbb{R}^n$. It turns out that the determinant $\det(\boldsymbol{A})$ is the signed volume of an n-dimensional parallelepiped formed by columns of the matrix $\boldsymbol{A}$.

For $n = 2$, the columns of the matrix form a parallelogram; see Figure 4.3. As the angle between vectors gets smaller, the area of a parallelogram shrinks, too. Consider two vectors $\boldsymbol{b}, \boldsymbol{g}$ that form the columns of a matrix $\boldsymbol{A} = [\boldsymbol{b}, \boldsymbol{g}]$. Then, the absolute value of the determinant of $\boldsymbol{A}$ is the area of the parallelogram with vertices $\boldsymbol{0}, \boldsymbol{b}, \boldsymbol{g}, \boldsymbol{b}+\boldsymbol{g}$. In particular, if $\boldsymbol{b}, \boldsymbol{g}$ are linearly dependent so that $\boldsymbol{b} = \lambda\boldsymbol{g}$ for some $\lambda \in \mathbb{R}$, they no longer form a two-dimensional parallelogram. Therefore, the corresponding area is 0. On the contrary, if $\boldsymbol{b}, \boldsymbol{g}$ are linearly independent and are multiples of the canonical basis vectors $\boldsymbol{e}_1, \boldsymbol{e}_2$ then they can be written as $\boldsymbol{b} = \begin{bmatrix} b \\ 0 \end{bmatrix}$ and $\boldsymbol{g} = \begin{bmatrix} 0 \\ g \end{bmatrix}$, and the determinant is $\begin{vmatrix} b & 0 \\ 0 & g \end{vmatrix} = bg - 0 = bg$.

The sign of the determinant indicates the orientation of the spanning vectors $\boldsymbol{b}, \boldsymbol{g}$ with respect to the standard basis $(\boldsymbol{e}_1, \boldsymbol{e}_2)$. In our figure, flipping the order to $\boldsymbol{g}, \boldsymbol{b}$ swaps the columns of $\boldsymbol{A}$ and reverses the orientation of the shaded area. This becomes the familiar formula: area = height $\times$ length. This intuition extends to higher dimensions. In $\mathbb{R}^3$, we consider three vectors $\boldsymbol{r}, \boldsymbol{b}, \boldsymbol{g} \in \mathbb{R}^3$ spanning the edges of a parallelepiped, i.e., a solid with faces that are parallel parallelograms (see Figure 4.4). The absolute value of the determinant of the 3×3 matrix $[\boldsymbol{r}, \ \boldsymbol{b}, \ \boldsymbol{g}]$ is the volume of the solid. Thus, the determinant acts as a function that measures the signed volume formed by column vectors composed in a matrix.

The sign of the determinant indicates the orientation of the spanning vectors.

Consider the three linearly independent vectors $\boldsymbol{r}, \boldsymbol{g}, \boldsymbol{b} \in \mathbb{R}^3$ given as

$$\boldsymbol{r} = \begin{bmatrix} 2 \\ 0 \\ -8 \end{bmatrix}, \quad \boldsymbol{g} = \begin{bmatrix} 6 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 4 \\ -1 \end{bmatrix}. \tag{4.9}$$

Writing these vectors as the columns of a matrix

$$\boldsymbol{A} = [\boldsymbol{r},\ \boldsymbol{g},\ \boldsymbol{b}] = \begin{bmatrix} 2 & 6 & 1 \\ 0 & 1 & 4 \\ -8 & 0 & -1 \end{bmatrix} \tag{4.10}$$

allows us to compute the desired volume as

$$V = |\det(\boldsymbol{A})| = 186\,. \tag{4.11}$$

Computing the determinant of an $n \times n$ matrix requires a general algorithm to solve the cases for $n > 3$, which we are going to explore in the following. Theorem 4.2 below reduces the problem of computing the determinant of an $n \times n$ matrix to computing the determinant of $(n-1) \times (n-1)$ matrices. By recursively applying the Laplace expansion (Theorem 4.2), we can therefore compute determinants of $n \times n$ matrices by ultimately computing determinants of 2×2 matrices.

Laplace expansion

Theorem 4.2 (Laplace Expansion)**.** *Consider a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. Then, for all $j = 1, \ldots, n$:*

$\det(\boldsymbol{A}_{k,j})$ is called a *minor* and $(-1)^{k+j} \det(\boldsymbol{A}_{k,j})$ a *cofactor*.

1. *Expansion along column j*

$$\det(\boldsymbol{A}) = \sum_{k=1}^{n} (-1)^{k+j} a_{kj} \det(\boldsymbol{A}_{k,j})\,. \tag{4.12}$$

2. *Expansion along row j*

$$\det(\boldsymbol{A}) = \sum_{k=1}^{n} (-1)^{k+j} a_{jk} \det(\boldsymbol{A}_{j,k})\,. \tag{4.13}$$

Here $\boldsymbol{A}_{k,j} \in \mathbb{R}^{(n-1)\times(n-1)}$ is the submatrix of $\boldsymbol{A}$ that we obtain when deleting row k and column j.

Example 4.3 (Laplace Expansion)
Let us compute the determinant of

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.14}$$

using the Laplace expansion along the first row. Applying (4.13) yields

$$\begin{aligned} \begin{vmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 0 & 0 & 1 \end{vmatrix} &= (-1)^{1+1} \cdot 1 \begin{vmatrix} 1 & 2 \\ 0 & 1 \end{vmatrix} \\ &+ (-1)^{1+2} \cdot 2 \begin{vmatrix} 3 & 2 \\ 0 & 1 \end{vmatrix} + (-1)^{1+3} \cdot 3 \begin{vmatrix} 3 & 1 \\ 0 & 0 \end{vmatrix}\,. \end{aligned} \tag{4.15}$$

We use (4.6) to compute the determinants of all 2×2 matrices and obtain

$$\det(\boldsymbol{A}) = 1(1-0) - 2(3-0) + 3(0-0) = -5\,. \tag{4.16}$$

For completeness we can compare this result to computing the determinant using Sarrus' rule (4.7):

$$\det(\boldsymbol{A}) = 1{\cdot}1{\cdot}1 + 3{\cdot}0{\cdot}3 + 0{\cdot}2{\cdot}2 - 0{\cdot}1{\cdot}3 - 1{\cdot}0{\cdot}2 - 3{\cdot}2{\cdot}1 = 1 - 6 = -5\,. \tag{4.17}$$

For $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ the determinant exhibits the following properties:

- The determinant of a matrix product is the product of the corresponding determinants, $\det(\boldsymbol{AB}) = \det(\boldsymbol{A})\det(\boldsymbol{B})$.
- Determinants are invariant to transposition, i.e., $\det(\boldsymbol{A}) = \det(\boldsymbol{A}^\top)$.
- If $\boldsymbol{A}$ is regular (invertible), then $\det(\boldsymbol{A}^{-1}) = \frac{1}{\det(\boldsymbol{A})}$.
- Similar matrices (Definition 2.22) possess the same determinant. Therefore, for a linear mapping $\Phi : V \to V$ all transformation matrices $\boldsymbol{A}_\Phi$ of Φ have the same determinant. Thus, the determinant is invariant to the choice of basis of a linear mapping.
- Adding a multiple of a column/row to another one does not change $\det(\boldsymbol{A})$.
- Multiplication of a column/row with $\lambda \in \mathbb{R}$ scales $\det(\boldsymbol{A})$ by λ. In particular, $\det(\lambda\boldsymbol{A}) = \lambda^n \det(\boldsymbol{A})$.
- Swapping two rows/columns changes the sign of $\det(\boldsymbol{A})$.

Because of the last three properties, we can use Gaussian elimination (see Section 2.1) to compute $\det(\boldsymbol{A})$ by bringing $\boldsymbol{A}$ into row-echelon form. We can stop Gaussian elimination when we have $\boldsymbol{A}$ in a triangular form where the elements below the diagonal are all 0. Recall from (4.8) that the determinant of a triangular matrix is the product of the diagonal elements.

Theorem 4.3. *A square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ has $\det(\boldsymbol{A}) \neq 0$ if and only if $\mathrm{rk}(\boldsymbol{A}) = n$. In other words, $\boldsymbol{A}$ is invertible if and only if it is full rank.*

When mathematics was mainly performed by hand, the determinant calculation was considered an essential way to analyze matrix invertibility. However, contemporary approaches in machine learning use direct numerical methods that superseded the explicit calculation of the determinant. For example, in Chapter 2, we learned that inverse matrices can be computed by Gaussian elimination. Gaussian elimination can thus be used to compute the determinant of a matrix.

Determinants will play an important theoretical role for the following sections, especially when we learn about eigenvalues and eigenvectors (Section 4.2) through the characteristic polynomial.

Definition 4.4. The *trace* of a square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is defined as

trace

$$\mathrm{tr}(\boldsymbol{A}) := \sum_{i=1}^{n} a_{ii}\,, \tag{4.18}$$

i.e. , the trace is the sum of the diagonal elements of $\boldsymbol{A}$.

The trace satisfies the following properties:

- $\mathrm{tr}(\boldsymbol{A}+\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{A}) + \mathrm{tr}(\boldsymbol{B})$ for $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$
- $\mathrm{tr}(\alpha\boldsymbol{A}) = \alpha\mathrm{tr}(\boldsymbol{A})\,, \alpha \in \mathbb{R}$ for $\boldsymbol{A} \in \mathbb{R}^{n\times n}$
- $\mathrm{tr}(\boldsymbol{I}_n) = n$
- $\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{B}\boldsymbol{A})$ for $\boldsymbol{A} \in \mathbb{R}^{n\times k}, \boldsymbol{B} \in \mathbb{R}^{k\times n}$

It can be shown that only one function satisfies these four properties together – the trace (Gohberg et al., 2012).

The properties of the trace of matrix products are more general. Specifically, the trace is invariant under cyclic permutations, i.e.,

The trace is invariant under cyclic permutations.

$$\mathrm{tr}(\boldsymbol{A}\boldsymbol{K}\boldsymbol{L}) = \mathrm{tr}(\boldsymbol{K}\boldsymbol{L}\boldsymbol{A}) \tag{4.19}$$

for matrices $\boldsymbol{A} \in \mathbb{R}^{a\times k}, \boldsymbol{K} \in \mathbb{R}^{k\times l}, \boldsymbol{L} \in \mathbb{R}^{l\times a}$. This property generalizes to products of an arbitrary number of matrices. As a special case of (4.19), it follows that for two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$

$$\mathrm{tr}(\boldsymbol{x}\boldsymbol{y}^\top) = \mathrm{tr}(\boldsymbol{y}^\top\boldsymbol{x}) = \boldsymbol{y}^\top\boldsymbol{x} \in \mathbb{R}\,. \tag{4.20}$$

Given a linear mapping $\Phi : V \to V$, where V is a vector space, we define the trace of this map by using the trace of matrix representation of Φ. For a given basis of V, we can describe Φ by means of the transformation matrix $\boldsymbol{A}$. Then the trace of Φ is the trace of $\boldsymbol{A}$. For a different basis of V, it holds that the corresponding transformation matrix $\boldsymbol{B}$ of Φ can be obtained by a basis change of the form $\boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{S}$ for suitable $\boldsymbol{S}$ (see Section 2.7.2). For the corresponding trace of Φ, this means

$$\mathrm{tr}(\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{S}) \overset{(4.19)}{=} \mathrm{tr}(\boldsymbol{A}\boldsymbol{S}\boldsymbol{S}^{-1}) = \mathrm{tr}(\boldsymbol{A})\,. \tag{4.21}$$

Hence, while matrix representations of linear mappings are basis dependent the trace of a linear mapping Φ is independent of the basis.

In this section, we covered determinants and traces as functions characterizing a square matrix. Taking together our understanding of determinants and traces we can now define an important equation describing a matrix $\boldsymbol{A}$ in terms of a polynomial, which we will use extensively in the following sections.

Definition 4.5 (Characteristic Polynomial)**.** For $\lambda \in \mathbb{R}$ and a square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$

$$p_{\boldsymbol{A}}(\lambda) := \det(\boldsymbol{A} - \lambda\boldsymbol{I}) \tag{4.22a}$$

$$= c_0 + c_1\lambda + c_2\lambda^2 + \cdots + c_{n-1}\lambda^{n-1} + (-1)^n\lambda^n\,, \tag{4.22b}$$

characteristic polynomial

$c_0, \ldots, c_{n-1} \in \mathbb{R}$, is the *characteristic polynomial* of $\boldsymbol{A}$. In particular,

$$c_0 = \det(\boldsymbol{A}), \tag{4.23}$$
$$c_{n-1} = (-1)^{n-1}\mathrm{tr}(\boldsymbol{A}). \tag{4.24}$$

The characteristic polynomial (4.22a) will allow us to compute eigenvalues and eigenvectors, covered in the next section.

4.2 Eigenvalues and Eigenvectors

We will now get to know a new way to characterize a matrix and its associated linear mapping. Recall from Section 2.7.1 that every linear mapping has a unique transformation matrix given an ordered basis. We can interpret linear mappings and their associated transformation matrices by performing an "eigen" analysis. As we will see, the eigenvalues of a linear mapping will tell us how a special set of vectors, the eigenvectors, is transformed by the linear mapping.

Eigen is a German word meaning "characteristic", "self", or "own".

Definition 4.6. Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a square matrix. Then $\lambda \in \mathbb{R}$ is an *eigenvalue* of $\boldsymbol{A}$ and $\boldsymbol{x} \in \mathbb{R}^n\backslash\{\boldsymbol{0}\}$ is the corresponding *eigenvector* of $\boldsymbol{A}$ if

eigenvalue
eigenvector

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}. \tag{4.25}$$

We call (4.25) the *eigenvalue equation*.

eigenvalue equation

Remark. In the linear algebra literature and software, it is often a convention that eigenvalues are sorted in descending order, so that the largest eigenvalue and associated eigenvector are called the first eigenvalue and its associated eigenvector, and the second largest called the second eigenvalue and its associated eigenvector, and so on. However, textbooks and publications may have different or no notion of orderings. We do not want to presume an ordering in this book if not stated explicitly. ♢

The following statements are equivalent:

- λ is an eigenvalue of $\boldsymbol{A} \in \mathbb{R}^{n\times n}$.
- There exists an $\boldsymbol{x} \in \mathbb{R}^n\backslash\{\boldsymbol{0}\}$ with $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$, or equivalently, $(\boldsymbol{A} - \lambda\boldsymbol{I}_n)\boldsymbol{x} = \boldsymbol{0}$ can be solved non-trivially, i.e., $\boldsymbol{x} \neq \boldsymbol{0}$.
- $\mathrm{rk}(\boldsymbol{A} - \lambda\boldsymbol{I}_n) < n$.
- $\det(\boldsymbol{A} - \lambda\boldsymbol{I}_n) = 0$.

Definition 4.7 (Collinearity and Codirection)**.** Two vectors that point in the same direction are called *codirected*. Two vectors are *collinear* if they point in the same or the opposite direction.

codirected
collinear

Remark (Non-uniqueness of eigenvectors). If $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$ associated with eigenvalue λ, then for any $c \in \mathbb{R}\backslash\{0\}$ it holds that $c\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$ with the same eigenvalue since

$$\boldsymbol{A}(c\boldsymbol{x}) = c\boldsymbol{A}\boldsymbol{x} = c\lambda\boldsymbol{x} = \lambda(c\boldsymbol{x}). \tag{4.26}$$

Thus, all vectors that are collinear to $\boldsymbol{x}$ are also eigenvectors of $\boldsymbol{A}$.

♢

Theorem 4.8. *$\lambda \in \mathbb{R}$ is an eigenvalue of $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ if and only if λ is a root of the characteristic polynomial $p_{\boldsymbol{A}}(\lambda)$ of $\boldsymbol{A}$.*

algebraic multiplicity

Definition 4.9. Let a square matrix $\boldsymbol{A}$ have an eigenvalue λ_i. The *algebraic multiplicity* of λ_i is the number of times the root appears in the characteristic polynomial.

eigenspace
eigenspectrum
spectrum

Definition 4.10 (Eigenspace and Eigenspectrum)**.** For $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, the set of all eigenvectors of $\boldsymbol{A}$ associated with an eigenvalue λ spans a subspace of $\mathbb{R}^n$, which is called the *eigenspace* of $\boldsymbol{A}$ with respect to λ and is denoted by E_λ. The set of all eigenvalues of $\boldsymbol{A}$ is called the *eigenspectrum*, or just *spectrum*, of $\boldsymbol{A}$.

If λ is an eigenvalue of $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, then the corresponding eigenspace E_λ is the solution space of the homogeneous system of linear equations $(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$. Geometrically, the eigenvector corresponding to a nonzero eigenvalue points in a direction that is stretched by the linear mapping. The eigenvalue is the factor by which it is stretched. If the eigenvalue is negative, the direction of the stretching is flipped.

Example 4.4 (The Case of the Identity Matrix)
The identity matrix $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ has characteristic polynomial $p_{\boldsymbol{I}}(\lambda) = \det(\boldsymbol{I} - \lambda\boldsymbol{I}) = (1-\lambda)^n = 0$, which has only one eigenvalue $\lambda = 1$ that occurs n times. Moreover, $\boldsymbol{I}\boldsymbol{x} = \lambda\boldsymbol{x} = 1\boldsymbol{x}$ holds for all vectors $\boldsymbol{x} \in \mathbb{R}^n\backslash\{\boldsymbol{0}\}$. Because of this, the sole eigenspace E_1 of the identity matrix spans n dimensions, and all n standard basis vectors of $\mathbb{R}^n$ are eigenvectors of $\boldsymbol{I}$.

Useful properties regarding eigenvalues and eigenvectors include the following:

- A matrix $\boldsymbol{A}$ and its transpose $\boldsymbol{A}^\top$ possess the same eigenvalues, but not necessarily the same eigenvectors.
- The eigenspace E_λ is the null space of $\boldsymbol{A} - \lambda\boldsymbol{I}$ since

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x} \iff \boldsymbol{A}\boldsymbol{x} - \lambda\boldsymbol{x} = \boldsymbol{0} \tag{4.27a}$$
$$\iff (\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{x} = \boldsymbol{0} \iff \boldsymbol{x} \in \ker(\boldsymbol{A} - \lambda\boldsymbol{I}). \tag{4.27b}$$

- Similar matrices (see Definition 2.22) possess the same eigenvalues. Therefore, a linear mapping Φ has eigenvalues that are independent of the choice of basis of its transformation matrix. This makes eigenvalues, together with the determinant and the trace, key characteristic parameters of a linear mapping as they are all invariant under basis change.
- Symmetric, positive definite matrices always have positive, real eigenvalues.

Example 4.5 (Computing Eigenvalues, Eigenvectors, and Eigenspaces)

Let us find the eigenvalues and eigenvectors of the 2×2 matrix

$$\boldsymbol{A} = \begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix}. \tag{4.28}$$

Step 1: Characteristic Polynomial. From our definition of the eigenvector $\boldsymbol{x} \neq \boldsymbol{0}$ and eigenvalue λ of $\boldsymbol{A}$, there will be a vector such that $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$, i.e., $(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{x} = \boldsymbol{0}$. Since $\boldsymbol{x} \neq \boldsymbol{0}$, this requires that the kernel (null space) of $\boldsymbol{A} - \lambda\boldsymbol{I}$ contains more elements than just $\boldsymbol{0}$. This means that $\boldsymbol{A} - \lambda\boldsymbol{I}$ is not invertible and therefore $\det(\boldsymbol{A} - \lambda\boldsymbol{I}) = 0$. Hence, we need to compute the roots of the characteristic polynomial (4.22a) to find the eigenvalues.

Step 2: Eigenvalues. The characteristic polynomial is

$$p_{\boldsymbol{A}}(\lambda) = \det(\boldsymbol{A} - \lambda\boldsymbol{I}) \tag{4.29a}$$

$$= \det\left(\begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right) = \begin{vmatrix} 4-\lambda & 2 \\ 1 & 3-\lambda \end{vmatrix} \tag{4.29b}$$

$$= (4-\lambda)(3-\lambda) - 2 \cdot 1\,. \tag{4.29c}$$

We factorize the characteristic polynomial and obtain

$$p(\lambda) = (4-\lambda)(3-\lambda) - 2 \cdot 1 = 10 - 7\lambda + \lambda^2 = (2-\lambda)(5-\lambda) \tag{4.30}$$

giving the roots $\lambda_1 = 2$ and $\lambda_2 = 5$.

Step 3: Eigenvectors and Eigenspaces. We find the eigenvectors that correspond to these eigenvalues by looking at vectors $\boldsymbol{x}$ such that

$$\begin{bmatrix} 4-\lambda & 2 \\ 1 & 3-\lambda \end{bmatrix} \boldsymbol{x} = \boldsymbol{0}\,. \tag{4.31}$$

For $\lambda = 5$ we obtain

$$\begin{bmatrix} 4-5 & 2 \\ 1 & 3-5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 & 2 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \boldsymbol{0}\,. \tag{4.32}$$

We solve this homogeneous system and obtain a solution space

$$E_5 = \operatorname{span}[\begin{bmatrix} 2 \\ 1 \end{bmatrix}]\,. \tag{4.33}$$

This eigenspace is one-dimensional as it possesses a single basis vector.

Analogously, we find the eigenvector for $\lambda = 2$ by solving the homogeneous system of equations

$$\begin{bmatrix} 4-2 & 2 \\ 1 & 3-2 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix} \boldsymbol{x} = \boldsymbol{0}\,. \tag{4.34}$$

This means any vector $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, where $x_2 = -x_1$, such as $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$, is an eigenvector with eigenvalue 2. The corresponding eigenspace is given as

$$E_2 = \text{span}[\begin{bmatrix} 1 \\ -1 \end{bmatrix}]. \tag{4.35}$$

The two eigenspaces E_5 and E_2 in Example 4.5 are one-dimensional as they are each spanned by a single vector. However, in other cases we may have multiple identical eigenvalues (see Definition 4.9) and the eigenspace may have more than one dimension.

geometric multiplicity

Definition 4.11. Let λ_i be an eigenvalue of a square matrix $\boldsymbol{A}$. Then the *geometric multiplicity* of λ_i is the number of linearly independent eigenvectors associated with λ_i. In other words, it is the dimensionality of the eigenspace spanned by the eigenvectors associated with λ_i.

Remark. A specific eigenvalue's geometric multiplicity must be at least one because every eigenvalue has at least one associated eigenvector. An eigenvalue's geometric multiplicity cannot exceed its algebraic multiplicity, but it may be lower. $\diamondsuit$

Example 4.6

The matrix $\boldsymbol{A} = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$ has two repeated eigenvalues $\lambda_1 = \lambda_2 = 2$ and an algebraic multiplicity of 2. The eigenvalue has, however, only one distinct unit eigenvector $\boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and, thus, geometric multiplicity 1.

Graphical Intuition in Two Dimensions

Let us gain some intuition for determinants, eigenvectors, and eigenvalues using different linear mappings. Figure 4.2 depicts five transformation matrices $\boldsymbol{A}_1, \dots, \boldsymbol{A}_5$ and their impact on a square grid of points, centered at the origin:

In geometry, the area-preserving properties of this type of shearing parallel to an axis is also known as Cavalieri's principle of equal areas for parallelograms (Katz, 2004).

- $\boldsymbol{A}_1 = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 2 \end{bmatrix}$. The direction of the two eigenvectors correspond to the canonical basis vectors in $\mathbb{R}^2$, i.e., to two cardinal axes. The vertical axis is extended by a factor of 2 (eigenvalue $\lambda_1 = 2$), and the horizontal axis is compressed by factor $\frac{1}{2}$ (eigenvalue $\lambda_2 = \frac{1}{2}$). The mapping is area preserving ($\det(\boldsymbol{A}_1) = 1 = 2 \cdot \frac{1}{2}$).
- $\boldsymbol{A}_2 = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{bmatrix}$ corresponds to a shearing mapping , i.e., it shears the points along the horizontal axis to the right if they are on the positive

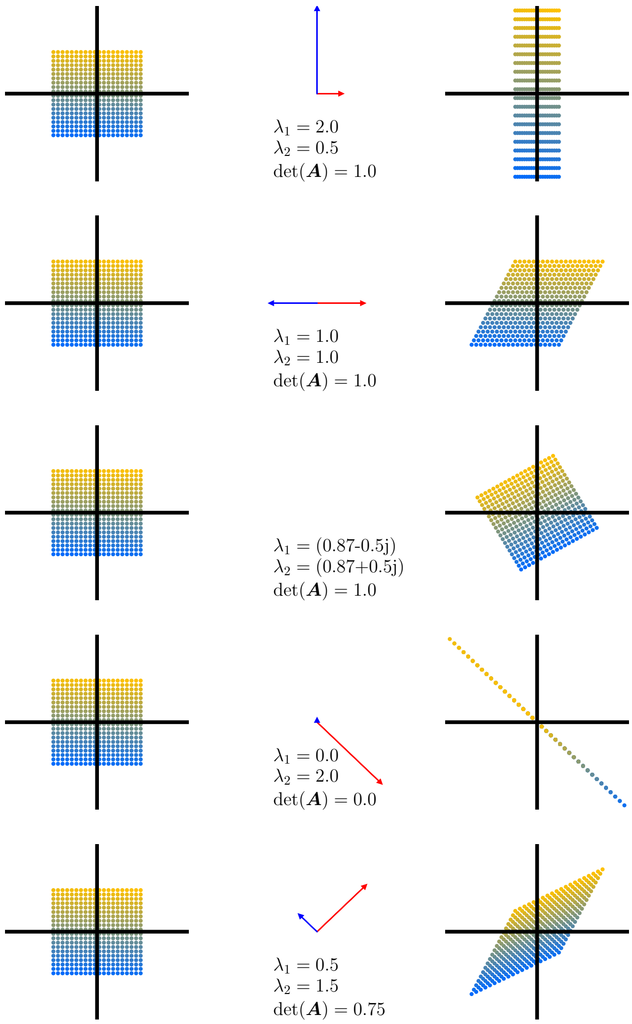


Figure 4.2 Determinants and eigenspaces. Overview of five linear mappings and their associated transformation matrices $\boldsymbol{A}_i \in \mathbb{R}^{2\times 2}$ projecting 400 color-coded points $\boldsymbol{x} \in \mathbb{R}^2$ (left column) onto target points $\boldsymbol{A}_i\boldsymbol{x}$ (right column). The central column depicts the first eigenvector, stretched by its associated eigenvalue λ_1, and the second eigenvector stretched by its eigenvalue λ_2. Each row depicts the effect of one of five transformation matrices $\boldsymbol{A}_i$ with respect to the standard basis.

half of the vertical axis, and to the left vice versa. This mapping is area preserving ($\det(\boldsymbol{A}_2) = 1$). The eigenvalue $\lambda_1 = 1 = \lambda_2$ is repeated and the eigenvectors are collinear (drawn here for emphasis in two opposite directions). This indicates that the mapping acts only along one direction (the horizontal axis).

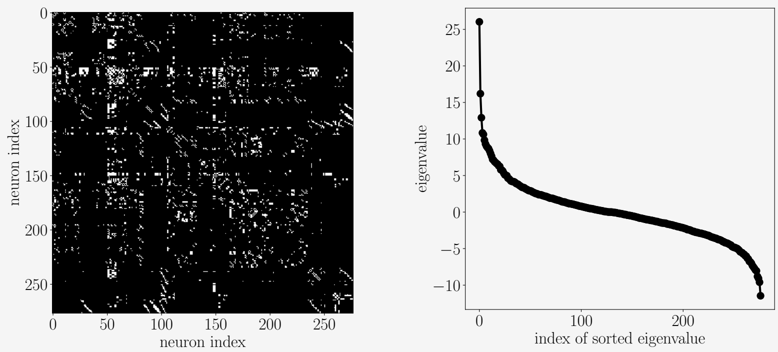
- $\boldsymbol{A}_3 = \begin{bmatrix} \cos(\frac{\pi}{6}) & -\sin(\frac{\pi}{6}) \\ \sin(\frac{\pi}{6}) & \cos(\frac{\pi}{6}) \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \sqrt{3} & -1 \\ 1 & \sqrt{3} \end{bmatrix}$ The matrix $\boldsymbol{A}_3$ rotates the points by $\frac{\pi}{6}$ rad $= 30^\circ$ counter-clockwise and has only complex eigenvalues, reflecting that the mapping is a rotation (hence, no eigenvectors are drawn). A rotation has to be volume preserving, and so the determinant is 1. For more details on rotations, we refer to Section 3.9.
- $\boldsymbol{A}_4 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$ represents a mapping in the standard basis that collapses a two-dimensional domain onto one dimension. Since one eigen-

value is 0, the space in direction of the (blue) eigenvector corresponding to $\lambda_1 = 0$ collapses, while the orthogonal (red) eigenvector stretches space by a factor $\lambda_2 = 2$. Therefore, the area of the image is 0.

- $\boldsymbol{A}_5 = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}$ is a shear-and-stretch mapping that scales space by 75% since $|\det(\boldsymbol{A}_5)| = \frac{3}{4}$. It stretches space along the (red) eigenvector of λ_2 by a factor 1.5 and compresses it along the orthogonal (blue) eigenvector by a factor 0.5.

Example 4.7 (Eigenspectrum of a Biological Neural Network)

Figure 4.3 Caenorhabditis elegans neural network (Kaiser and Hilgetag, 2006).(a) Symmetrized connectivity matrix; (b) Eigenspectrum.



(a) Connectivity matrix. (b) Eigenspectrum.

Methods to analyze and learn from network data are an essential component of machine learning methods. The key to understanding networks is the connectivity between network nodes, especially if two nodes are connected to each other or not. In data science applications, it is often useful to study the matrix that captures this connectivity data.

We build a connectivity/adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{277\times277}$ of the complete neural network of the worm *C.Elegans*. Each row/column represents one of the 277 neurons of this worm's brain. The connectivity matrix $\boldsymbol{A}$ has a value of $a_{ij} = 1$ if neuron i talks to neuron j through a synapse, and $a_{ij} = 0$ otherwise. The connectivity matrix is not symmetric, which implies that eigenvalues may not be real valued. Therefore, we compute a symmetrized version of the connectivity matrix as $\boldsymbol{A}_{sym} := \boldsymbol{A} + \boldsymbol{A}^\top$. This new matrix $\boldsymbol{A}_{sym}$ is shown in Figure 4.3(a) and has a nonzero value a_{ij} if and only if two neurons are connected (white pixels), irrespective of the direction of the connection. In Figure 4.3(b), we show the corresponding eigenspectrum of $\boldsymbol{A}_{sym}$. The horizontal axis shows the index of the eigenvalues, sorted in descending order. The vertical axis shows the corresponding eigenvalue. The S-like shape of this eigenspectrum is typical for many biological neural networks. The underlying mechanism responsible for this is an area of active neuroscience research.

Theorem 4.12. *The eigenvectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ with n distinct eigenvalues $\lambda_1, \ldots, \lambda_n$ are linearly independent.*

This theorem states that eigenvectors of a matrix with n distinct eigenvalues form a basis of $\mathbb{R}^n$.

Definition 4.13. A square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is *defective* if it possesses fewer than n linearly independent eigenvectors.

defective

A non-defective matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ does not necessarily require n distinct eigenvalues, but it does require that the eigenvectors form a basis of $\mathbb{R}^n$. Looking at the eigenspaces of a defective matrix, it follows that the sum of the dimensions of the eigenspaces is less than n. Specifically, a defective matrix has at least one eigenvalue λ_i with an algebraic multiplicity $m > 1$ and a geometric multiplicity of less than m.

Remark. A defective matrix cannot have n distinct eigenvalues, as distinct eigenvalues have linearly independent eigenvectors (Theorem 4.12). $\diamondsuit$

Theorem 4.14. *Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, we can always obtain a symmetric, positive semidefinite matrix $\boldsymbol{S} \in \mathbb{R}^{n\times n}$ by defining*

$$\boldsymbol{S} := \boldsymbol{A}^\top \boldsymbol{A}\,. \tag{4.36}$$

Remark. If $\mathrm{rk}(\boldsymbol{A}) = n$, then $\boldsymbol{S} := \boldsymbol{A}^\top\boldsymbol{A}$ is symmetric, positive definite. $\diamondsuit$

Understanding why Theorem 4.14 holds is insightful for how we can use symmetrized matrices: Symmetry requires $\boldsymbol{S} = \boldsymbol{S}^\top$, and by inserting (4.36) we obtain $\boldsymbol{S} = \boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{A}^\top(\boldsymbol{A}^\top)^\top = (\boldsymbol{A}^\top\boldsymbol{A})^\top = \boldsymbol{S}^\top$. Moreover, positive semidefiniteness (Section 3.2.3) requires that $\boldsymbol{x}^\top\boldsymbol{S}\boldsymbol{x} \geqslant 0$ and inserting (4.36) we obtain $\boldsymbol{x}^\top\boldsymbol{S}\boldsymbol{x} = \boldsymbol{x}^\top\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} = (\boldsymbol{x}^\top\boldsymbol{A}^\top)(\boldsymbol{A}\boldsymbol{x}) = (\boldsymbol{A}\boldsymbol{x})^\top(\boldsymbol{A}\boldsymbol{x}) \geqslant 0$, because the dot product computes a sum of squares (which are themselves non-negative).

spectral theorem

Theorem 4.15 (Spectral Theorem)**.** *If $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is symmetric, there exists an orthonormal basis of the corresponding vector space V consisting of eigenvectors of $\boldsymbol{A}$, and each eigenvalue is real.*

A direct implication of the spectral theorem is that the eigendecomposition of a symmetric matrix $\boldsymbol{A}$ exists (with real eigenvalues), and that we can find an ONB of eigenvectors so that $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^\top$, where $\boldsymbol{D}$ is diagonal and the columns of $\boldsymbol{P}$ contain the eigenvectors.

Example 4.8
Consider the matrix

$$\boldsymbol{A} = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{bmatrix}. \tag{4.37}$$

The characteristic polynomial of $\boldsymbol{A}$ is

$$p_{\boldsymbol{A}}(\lambda) = -(\lambda - 1)^2(\lambda - 7)\,, \tag{4.38}$$

so that we obtain the eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 7$, where λ_1 is a repeated eigenvalue. Following our standard procedure for computing eigenvectors, we obtain the eigenspaces

$$E_1 = \mathrm{span}[\underbrace{\begin{bmatrix}-1\\1\\0\end{bmatrix}}_{=:\boldsymbol{x}_1}, \underbrace{\begin{bmatrix}-1\\0\\1\end{bmatrix}}_{=:\boldsymbol{x}_2}], \quad E_7 = \mathrm{span}[\underbrace{\begin{bmatrix}1\\1\\1\end{bmatrix}}_{=:\boldsymbol{x}_3}]\,. \tag{4.39}$$

We see that $\boldsymbol{x}_3$ is orthogonal to both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. However, since $\boldsymbol{x}_1^\top \boldsymbol{x}_2 = 1 \neq 0$, they are not orthogonal. The spectral theorem (Theorem 4.15) states that there exists an orthogonal basis, but the one we have is not orthogonal. However, we can construct one.

To construct such a basis, we exploit the fact that $\boldsymbol{x}_1, \boldsymbol{x}_2$ are eigenvectors associated with the same eigenvalue λ. Therefore, for any $\alpha, \beta \in \mathbb{R}$ it holds that

$$\boldsymbol{A}(\alpha\boldsymbol{x}_1 + \beta\boldsymbol{x}_2) = \boldsymbol{A}\boldsymbol{x}_1\alpha + \boldsymbol{A}\boldsymbol{x}_2\beta = \lambda(\alpha\boldsymbol{x}_1 + \beta\boldsymbol{x}_2)\,, \tag{4.40}$$

i.e., any linear combination of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is also an eigenvector of $\boldsymbol{A}$ associated with λ. The Gram-Schmidt algorithm (Section 3.8.3) is a method for iteratively constructing an orthogonal/orthonormal basis from a set of basis vectors using such linear combinations. Therefore, even if $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are not orthogonal, we can apply the Gram-Schmidt algorithm and find eigenvectors associated with $\lambda_1 = 1$ that are orthogonal to each other (and to $\boldsymbol{x}_3$). In our example, we will obtain

$$\boldsymbol{x}_1' = \begin{bmatrix}-1\\1\\0\end{bmatrix}, \quad \boldsymbol{x}_2' = \frac{1}{2}\begin{bmatrix}-1\\-1\\2\end{bmatrix}, \tag{4.41}$$

which are orthogonal to each other, orthogonal to $\boldsymbol{x}_3$, and eigenvectors of $\boldsymbol{A}$ associated with $\lambda_1 = 1$.

Before we conclude our considerations of eigenvalues and eigenvectors it is useful to tie these matrix characteristics together with the concepts of the determinant and the trace.

Theorem 4.16. *The determinant of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is the product of its eigenvalues, i.e.,*

$$\det(\boldsymbol{A}) = \prod_{i=1}^{n} \lambda_i\,, \tag{4.42}$$

where $\lambda_i \in \mathbb{C}$ are (possibly repeated) eigenvalues of $\boldsymbol{A}$.

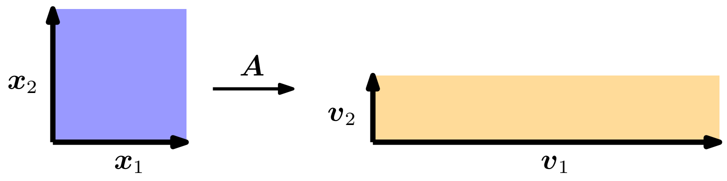


Figure 4.1 Geometric interpretation of eigenvalues. The eigenvectors of $\boldsymbol{A}$ get stretched by the corresponding eigenvalues. The area of the unit square changes by $|\lambda_1\lambda_2|$, the perimeter changes by a factor of $\frac{1}{2}(|\lambda_1| + |\lambda_2|)$.

Theorem 4.17. *The trace of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is the sum of its eigenvalues, i.e.,*

$$tr(\boldsymbol{A}) = \sum_{i=1}^{n} \lambda_i \,, \tag{4.43}$$

where $\lambda_i \in \mathbb{C}$ are (possibly repeated) eigenvalues of $\boldsymbol{A}$.

Let us provide a geometric intuition of these two theorems. Consider a matrix $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ that possesses two linearly independent eigenvectors $\boldsymbol{x}_1, \boldsymbol{x}_2$. For this example, we assume $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ are an ONB of $\mathbb{R}^2$ so that they are orthogonal and the area of the square they span is 1; see Figure 4.1. From Section 4.1, we know that the determinant computes the change of area of unit square under the transformation $\boldsymbol{A}$. In this example, we can compute the change of area explicitly: Mapping the eigenvectors using $\boldsymbol{A}$ gives us vectors $\boldsymbol{v}_1 = \boldsymbol{A}\boldsymbol{x}_1 = \lambda_1\boldsymbol{x}_1$ and $\boldsymbol{v}_2 = \boldsymbol{A}\boldsymbol{x}_2 = \lambda_2\boldsymbol{x}_2$, i.e., the new vectors $\boldsymbol{v}_i$ are scaled versions of the eigenvectors $\boldsymbol{x}_i$, and the scaling factors are the corresponding eigenvalues λ_i. $\boldsymbol{v}_1, \boldsymbol{v}_2$ are still orthogonal, and the area of the rectangle they span is $|\lambda_1\lambda_2|$.

Given that $\boldsymbol{x}_1, \boldsymbol{x}_2$ (in our example) are orthonormal, we can directly compute the perimeter of the unit square as $2(1+1)$. Mapping the eigenvectors using $\boldsymbol{A}$ creates a rectangle whose perimeter is $2(|\lambda_1| + |\lambda_2|)$. Therefore, the sum of the absolute values of the eigenvalues tells us how the perimeter of the unit square changes under the transformation matrix $\boldsymbol{A}$.

Example 4.9 (Google's PageRank – Webpages as Eigenvectors)
Google uses the eigenvector corresponding to the maximal eigenvalue of a matrix $\boldsymbol{A}$ to determine the rank of a page for search. The idea for the PageRank algorithm, developed at Stanford University by Larry Page and Sergey Brin in 1996, was that the importance of any web page can be approximated by the importance of pages that link to it. For this, they write down all web sites as a huge directed graph that shows which page links to which. PageRank computes the weight (importance) $x_i \geqslant 0$ of a web site a_i by counting the number of pages pointing to a_i. Moreover, PageRank takes into account the importance of the web sites that link to a_i. The navigation behavior of a user is then modeled by a transition matrix $\boldsymbol{A}$ of this graph that tells us with what (click) probability somebody will end up

PageRank

on a different web site. The matrix $\boldsymbol{A}$ has the property that for any initial rank/importance vector $\boldsymbol{x}$ of a web site the sequence $\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}^2\boldsymbol{x}, \ldots$ converges to a vector $\boldsymbol{x}^*$. This vector is called the *PageRank* and satisfies $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{x}^*$, i.e., it is an eigenvector (with corresponding eigenvalue 1) of $\boldsymbol{A}$. After normalizing $\boldsymbol{x}^*$, such that $\|\boldsymbol{x}^*\| = 1$, we can interpret the entries as probabilities. More details and different perspectives on PageRank can be found in the original technical report (Page et al., 1999).

4.3 Cholesky Decomposition

Cholesky decomposition
Cholesky factorization

There are many ways to factorize special types of matrices that we encounter often in machine learning. In the positive real numbers, we have the square-root operation that gives us a decomposition of the number into identical components, e.g., $9 = 3 \cdot 3$. For matrices, we need to be careful that we compute a square-root-like operation on positive quantities. For symmetric, positive definite matrices (see Section 3.2.3), we can choose from a number of square-root equivalent operations. The *Cholesky decomposition/Cholesky factorization* provides a square-root equivalent operation on symmetric, positive definite matrices that is useful in practice.

Theorem 4.18 (Cholesky Decomposition)**.** *A symmetric, positive definite matrix $\boldsymbol{A}$ can be factorized into a product $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^\top$, where $\boldsymbol{L}$ is a lower-triangular matrix with positive diagonal elements:*

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & \cdots & l_{n1} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & l_{nn} \end{bmatrix}. \tag{4.44}$$

Cholesky factor

$\boldsymbol{L}$ is called the Cholesky factor of $\boldsymbol{A}$, and $\boldsymbol{L}$ is unique.

Example 4.10 (Cholesky Factorization)

Consider a symmetric, positive definite matrix $\boldsymbol{A} \in \mathbb{R}^{3\times 3}$. We are interested in finding its Cholesky factorization $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^\top$, i.e.,

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \boldsymbol{L}\boldsymbol{L}^\top = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}. \tag{4.45}$$

Multiplying out the right-hand side yields

$$\boldsymbol{A} = \begin{bmatrix} l_{11}^2 & l_{21}l_{11} & l_{31}l_{11} \\ l_{21}l_{11} & l_{21}^2 + l_{22}^2 & l_{31}l_{21} + l_{32}l_{22} \\ l_{31}l_{11} & l_{31}l_{21} + l_{32}l_{22} & l_{31}^2 + l_{32}^2 + l_{33}^2 \end{bmatrix}. \tag{4.46}$$

Comparing the left-hand side of (4.45) and the right-hand side of (4.46) shows that there is a simple pattern in the diagonal elements l_{ii}:

$$l_{11} = \sqrt{a_{11}}\,, \quad l_{22} = \sqrt{a_{22} - l_{21}^2}\,, \quad l_{33} = \sqrt{a_{33} - (l_{31}^2 + l_{32}^2)}\,. \tag{4.47}$$

Similarly for the elements below the diagonal (l_{ij}, where $i > j$), there is also a repeating pattern:

$$l_{21} = \frac{1}{l_{11}} a_{21}\,, \quad l_{31} = \frac{1}{l_{11}} a_{31}\,, \quad l_{32} = \frac{1}{l_{22}} (a_{32} - l_{31} l_{21})\,. \tag{4.48}$$

Thus, we constructed the Cholesky decomposition for any symmetric, positive definite 3×3 matrix. The key realization is that we can backward calculate what the components l_{ij} for the $\boldsymbol{L}$ should be, given the values a_{ij} for $\boldsymbol{A}$ and previously computed values of l_{ij}.

The Cholesky decomposition is an important tool for the numerical computations underlying machine learning. Here, symmetric positive definite matrices require frequent manipulation, e.g., the covariance matrix of a multivariate Gaussian variable (see Section 6.5) is symmetric, positive definite. The Cholesky factorization of this covariance matrix allows us to generate samples from a Gaussian distribution. It also allows us to perform a linear transformation of random variables, which is heavily exploited when computing gradients in deep stochastic models, such as the variational auto-encoder (Jimenez Rezende et al., 2014; Kingma and Welling, 2014). The Cholesky decomposition also allows us to compute determinants very efficiently. Given the Cholesky decomposition $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^\top$, we know that $\det(\boldsymbol{A}) = \det(\boldsymbol{L})\det(\boldsymbol{L}^\top) = \det(\boldsymbol{L})^2$. Since $\boldsymbol{L}$ is a triangular matrix, the determinant is simply the product of its diagonal entries so that $\det(\boldsymbol{A}) = \prod_i l_{ii}^2$. Thus, many numerical software packages use the Cholesky decomposition to make computations more efficient.

4.4 Eigendecomposition and Diagonalization

A *diagonal matrix* is a matrix that has value zero on all off-diagonal elements, i.e., they are of the form diagonal matrix

$$\boldsymbol{D} = \begin{bmatrix} c_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & c_n \end{bmatrix}. \tag{4.49}$$

They allow fast computation of determinants, powers, and inverses. The determinant is the product of its diagonal entries, a matrix power $\boldsymbol{D}^k$ is given by each diagonal element raised to the power k, and the inverse $\boldsymbol{D}^{-1}$ is the reciprocal of its diagonal elements if all of them are nonzero.

In this section, we will discuss how to transform matrices into diagonal form. This is an important application of the basis change we discussed in Section 2.7.2 and eigenvalues from Section 4.2.

Recall that two matrices $\boldsymbol{A}, \boldsymbol{D}$ are similar (Definition 2.22) if there exists an invertible matrix $\boldsymbol{P}$, such that $\boldsymbol{D} = \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$. More specifically, we will look at matrices $\boldsymbol{A}$ that are similar to diagonal matrices $\boldsymbol{D}$ that contain the eigenvalues of $\boldsymbol{A}$ on the diagonal.

diagonalizable

Definition 4.19 (Diagonalizable)**.** A matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is *diagonalizable* if it is similar to a diagonal matrix, i.e., if there exists an invertible matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ such that $\boldsymbol{D} = \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$.

In the following, we will see that diagonalizing a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is a way of expressing the same linear mapping but in another basis (see Section 2.6.1), which will turn out to be a basis that consists of the eigenvectors of $\boldsymbol{A}$.

Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, let $\lambda_1, \dots, \lambda_n$ be a set of scalars, and let $\boldsymbol{p}_1, \dots, \boldsymbol{p}_n$ be a set of vectors in $\mathbb{R}^n$. We define $\boldsymbol{P} := [\boldsymbol{p}_1, \dots, \boldsymbol{p}_n]$ and let $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ be a diagonal matrix with diagonal entries $\lambda_1, \dots, \lambda_n$. Then we can show that

$$\boldsymbol{A}\boldsymbol{P} = \boldsymbol{P}\boldsymbol{D} \tag{4.50}$$

if and only if $\lambda_1, \dots, \lambda_n$ are the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{p}_1, \dots, \boldsymbol{p}_n$ are corresponding eigenvectors of $\boldsymbol{A}$.

We can see that this statement holds because

$$\boldsymbol{A}\boldsymbol{P} = \boldsymbol{A}[\boldsymbol{p}_1, \dots, \boldsymbol{p}_n] = [\boldsymbol{A}\boldsymbol{p}_1, \dots, \boldsymbol{A}\boldsymbol{p}_n]\,, \tag{4.51}$$

$$\boldsymbol{P}\boldsymbol{D} = [\boldsymbol{p}_1, \dots, \boldsymbol{p}_n]\begin{bmatrix}\lambda_1 & & 0\\ & \ddots & \\ 0 & & \lambda_n\end{bmatrix} = [\lambda_1\boldsymbol{p}_1, \dots, \lambda_n\boldsymbol{p}_n]\,. \tag{4.52}$$

Thus, (4.50) implies that

$$\boldsymbol{A}\boldsymbol{p}_1 = \lambda_1\boldsymbol{p}_1 \tag{4.53}$$

$$\vdots$$

$$\boldsymbol{A}\boldsymbol{p}_n = \lambda_n\boldsymbol{p}_n\,. \tag{4.54}$$

Therefore, the columns of $\boldsymbol{P}$ must be eigenvectors of $\boldsymbol{A}$.

Our definition of diagonalization requires that $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ is invertible, i.e., $\boldsymbol{P}$ has full rank (Theorem 4.3). This requires us to have n linearly independent eigenvectors $\boldsymbol{p}_1, \dots, \boldsymbol{p}_n$, i.e., the $\boldsymbol{p}_i$ form a basis of $\mathbb{R}^n$.

Theorem 4.20 (Eigendecomposition)**.** *A square matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ can be factored into*

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}\,, \tag{4.55}$$

where $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{D}$ is a diagonal matrix whose diagonal entries are the eigenvalues of $\boldsymbol{A}$, if and only if the eigenvectors of $\boldsymbol{A}$ form a basis of $\mathbb{R}^n$.

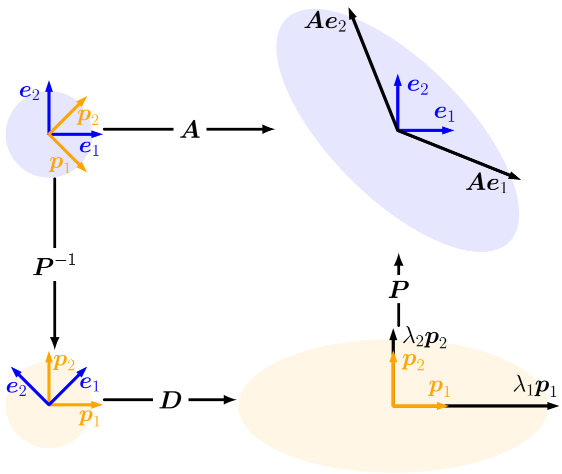


Figure 4.1 Intuition behind the eigendecomposition as sequential transformations. Top-left to bottom-left: P^{-1} performs a basis change (here drawn in $\mathbb{R}^2$ and depicted as a rotation-like operation) from the standard basis into the eigenbasis. Bottom-left to bottom-right: D performs a scaling along the remapped orthogonal eigenvectors, depicted here by a circle being stretched to an ellipse. Bottom-right to top-right: P undoes the basis change (depicted as a reverse rotation) and restores the original coordinate frame.

Theorem 4.20 implies that only non-defective matrices can be diagonalized and that the columns of $\boldsymbol{P}$ are the n eigenvectors of $\boldsymbol{A}$. For symmetric matrices we can obtain even stronger outcomes for the eigenvalue decomposition.

Theorem 4.21. *A symmetric matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$ can always be diagonalized.*

Theorem 4.21 follows directly from the spectral theorem 4.15. Moreover, the spectral theorem states that we can find an ONB of eigenvectors of $\mathbb{R}^n$. This makes $\boldsymbol{P}$ an orthogonal matrix so that $\boldsymbol{D} = \boldsymbol{P}^\top \boldsymbol{A} \boldsymbol{P}$.

Remark. The Jordan normal form of a matrix offers a decomposition that works for defective matrices (Lang, 1987) but is beyond the scope of this book. $\diamondsuit$

Geometric Intuition for the Eigendecomposition

We can interpret the eigendecomposition of a matrix as follows (see also Figure 4.1): Let $\boldsymbol{A}$ be the transformation matrix of a linear mapping with respect to the standard basis $\boldsymbol{e}_i$ (blue arrows). $\boldsymbol{P}^{-1}$ performs a basis change from the standard basis into the eigenbasis. Then, the diagonal $\boldsymbol{D}$ scales the vectors along these axes by the eigenvalues λ_i. Finally, $\boldsymbol{P}$ transforms these scaled vectors back into the standard/canonical coordinates yielding $\lambda_i \boldsymbol{p}_i$.

Example 4.11 (Eigendecomposition)

Let us compute the eigendecomposition of $\boldsymbol{A} = \frac{1}{2}\begin{bmatrix} 5 & -2 \\ -2 & 5 \end{bmatrix}$.

Step 1: Compute eigenvalues and eigenvectors. The characteristic polynomial of $\boldsymbol{A}$ is

$$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = \det\left(\begin{bmatrix} \frac{5}{2} - \lambda & -1 \\ -1 & \frac{5}{2} - \lambda \end{bmatrix}\right) \tag{4.56a}$$

$$= (\tfrac{5}{2} - \lambda)^2 - 1 = \lambda^2 - 5\lambda + \tfrac{21}{4} = (\lambda - \tfrac{7}{2})(\lambda - \tfrac{3}{2}) \,. \tag{4.56b}$$

Therefore, the eigenvalues of $\boldsymbol{A}$ are $\lambda_1 = \frac{7}{2}$ and $\lambda_2 = \frac{3}{2}$ (the roots of the characteristic polynomial), and the associated (normalized) eigenvectors are obtained via

$$\boldsymbol{A}\boldsymbol{p}_1 = \frac{7}{2}\boldsymbol{p}_1 \,, \quad \boldsymbol{A}\boldsymbol{p}_2 = \frac{3}{2}\boldsymbol{p}_2 \,. \tag{4.57}$$

This yields

$$\boldsymbol{p}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} , \quad \boldsymbol{p}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} . \tag{4.58}$$

Step 2: Check for existence. The eigenvectors $\boldsymbol{p}_1, \boldsymbol{p}_2$ form a basis of $\mathbb{R}^2$. Therefore, $\boldsymbol{A}$ can be diagonalized.

Step 3: Construct the matrix $\boldsymbol{P}$ to diagonalize $\boldsymbol{A}$. We collect the eigenvectors of $\boldsymbol{A}$ in $\boldsymbol{P}$ so that

$$\boldsymbol{P} = [\boldsymbol{p}_1, \ \boldsymbol{p}_2] = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} . \tag{4.59}$$

We then obtain

$$\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P} = \begin{bmatrix} \frac{7}{2} & 0 \\ 0 & \frac{3}{2} \end{bmatrix} = \boldsymbol{D} \,. \tag{4.60}$$

Figure 4.1 visualizes the eigendecomposition of $\boldsymbol{A} = \begin{bmatrix} 5 & -2 \\ -2 & 5 \end{bmatrix}$ as a sequence of linear transformations.

Equivalently, we get (exploiting that $\boldsymbol{P}^{-1} = \boldsymbol{P}^\top$ since the eigenvectors $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ in this example form an ONB)

$$\underbrace{\frac{1}{2}\begin{bmatrix} 5 & -2 \\ -2 & 5 \end{bmatrix}}_{\boldsymbol{A}} = \underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}}_{\boldsymbol{P}} \underbrace{\begin{bmatrix} \frac{7}{2} & 0 \\ 0 & \frac{3}{2} \end{bmatrix}}_{\boldsymbol{D}} \underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}}_{\boldsymbol{P}^{-1}} . \tag{4.61}$$

- Diagonal matrices $\boldsymbol{D}$ can efficiently be raised to a power. Therefore, we can find a matrix power for a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ via the eigenvalue decomposition (if it exists) so that

$$\boldsymbol{A}^k = (\boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1})^k = \boldsymbol{P}\boldsymbol{D}^k\boldsymbol{P}^{-1} \,. \tag{4.62}$$

  Computing $\boldsymbol{D}^k$ is efficient because we apply this operation individually to any diagonal element.
- Assume that the eigendecomposition $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}$ exists. Then,

$$\det(\boldsymbol{A}) = \det(\boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}) = \det(\boldsymbol{P})\det(\boldsymbol{D})\det(\boldsymbol{P}^{-1}) \tag{4.63a}$$

$$= \det(\boldsymbol{D}) = \prod_i d_{ii} \tag{4.63b}$$

allows for an efficient computation of the determinant of $\boldsymbol{A}$.

The eigenvalue decomposition requires square matrices. It would be useful to perform a decomposition on general matrices. In the next section, we introduce a more general matrix decomposition technique, the singular value decomposition.

4.5 Singular Value Decomposition

The singular value decomposition (SVD) of a matrix is a central matrix decomposition method in linear algebra. It has been referred to as the "fundamental theorem of linear algebra" (Strang, 1993) because it can be applied to all matrices, not only to square matrices, and it always exists. Moreover, as we will explore in the following, the SVD of a matrix $\boldsymbol{A}$, which represents a linear mapping $\Phi : V \to W$, quantifies the change between the underlying geometry of these two vector spaces. We recommend the work by Kalman (1996) and Roy and Banerjee (2014) for a deeper overview of the mathematics of the SVD.

SVD theorem

Theorem 4.22 (SVD Theorem)**.** *Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ be a rectangular matrix of rank $r \in [0, \min(m, n)]$. The SVD of $\boldsymbol{A}$ is a decomposition of the form*

SVD
singular value decomposition

$$ {}_{m}\overset{n}{\boxed{\boldsymbol{A}}} = {}_{m}\overset{m}{\boxed{\boldsymbol{U}}}\ {}_{m}\overset{n}{\boxed{\boldsymbol{\Sigma}}}\ \overset{n}{\boxed{\boldsymbol{V}^\top}}{}_{n} \tag{4.64}$$

with an orthogonal matrix $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ with column vectors $\boldsymbol{u}_i$, $i = 1, \ldots, m$, and an orthogonal matrix $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ with column vectors $\boldsymbol{v}_j$, $j = 1, \ldots, n$. Moreover, $\boldsymbol{\Sigma}$ is an $m \times n$ matrix with $\Sigma_{ii} = \sigma_i \geqslant 0$ and $\Sigma_{ij} = 0,\ i \neq j$.

The diagonal entries σ_i, $i = 1, \ldots, r$, of $\boldsymbol{\Sigma}$ are called the *singular values*, $\boldsymbol{u}_i$ are called the *left-singular vectors*, and $\boldsymbol{v}_j$ are called the *right-singular vectors*. By convention, the singular values are ordered, i.e., $\sigma_1 \geqslant \sigma_2 \geqslant \sigma_r \geqslant 0$.

singular values
left-singular vectors
right-singular vectors

The *singular value matrix* $\boldsymbol{\Sigma}$ is unique, but it requires some attention. Observe that the $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times n}$ is rectangular. In particular, $\boldsymbol{\Sigma}$ is of the same size as $\boldsymbol{A}$. This means that $\boldsymbol{\Sigma}$ has a diagonal submatrix that contains the singular values and needs additional zero padding. Specifically, if $m > n$, then the matrix $\boldsymbol{\Sigma}$ has diagonal structure up to row n and then consists of

singular value matrix

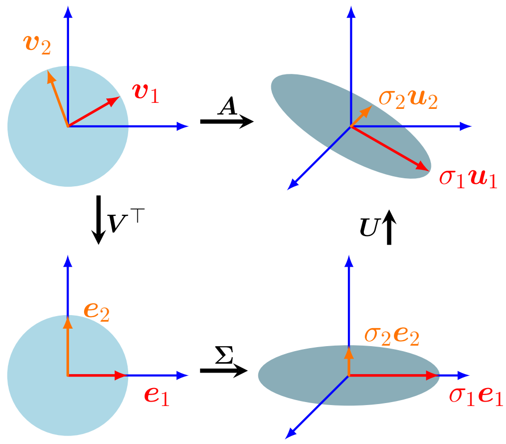


Figure 4.1 Intuition behind the SVD of a matrix $\boldsymbol{A} \in \mathbb{R}^{3\times 2}$ as sequential transformations. Top-left to bottom-left: $\boldsymbol{V}^\top$ performs a basis change in $\mathbb{R}^2$. Bottom-left to bottom-right: $\boldsymbol{\Sigma}$ scales and maps from $\mathbb{R}^2$ to $\mathbb{R}^3$. The ellipse in the bottom-right lives in $\mathbb{R}^3$. The third dimension is orthogonal to the surface of the elliptical disk. Bottom-right to top-right: $\boldsymbol{U}$ performs a basis change within $\mathbb{R}^3$.

$\boldsymbol{0}^\top$ row vectors from $n+1$ to m below so that

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n \\ 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{bmatrix} . \tag{4.65}$$

If $m < n$, the matrix $\boldsymbol{\Sigma}$ has a diagonal structure up to column m and columns that consist of $\boldsymbol{0}$ from $m+1$ to n:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & \vdots & & \vdots \\ 0 & 0 & \sigma_m & 0 & \dots & 0 \end{bmatrix} . \tag{4.66}$$

Remark. The SVD exists for any matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. $\diamondsuit$

4.5.1 Geometric Intuitions for the SVD

The SVD offers geometric intuitions to describe a transformation matrix $\boldsymbol{A}$. In the following, we will discuss the SVD as sequential linear transformations performed on the bases. In Example 4.12, we will then apply transformation matrices of the SVD to a set of vectors in $\mathbb{R}^2$, which allows us to visualize the effect of each transformation more clearly.

The SVD of a matrix can be interpreted as a decomposition of a corresponding linear mapping (recall Section 2.7.1) $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ into three operations; see Figure 4.1. The SVD intuition follows superficially a similar structure to our eigendecomposition intuition, see Figure 4.1: Broadly speaking, the SVD performs a basis change via $\boldsymbol{V}^\top$ followed by a scaling and augmentation (or reduction) in dimensionality via the singular value matrix $\boldsymbol{\Sigma}$. Finally, it performs a second basis change via $\boldsymbol{U}$. The SVD entails a number of important details and caveats, which is why we will review our intuition in more detail.

It is useful to review basis changes (Section 2.7.2), orthogonal matrices (Definition 3.8) and orthonormal bases (Section 3.5).

Assume we are given a transformation matrix of a linear mapping $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ with respect to the standard bases B and C of $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively. Moreover, assume a second basis $\tilde{B}$ of $\mathbb{R}^n$ and $\tilde{C}$ of $\mathbb{R}^m$. Then

1. The matrix $\boldsymbol{V}$ performs a basis change in the domain $\mathbb{R}^n$ from $\tilde{B}$ (represented by the red and orange vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ in the top-left of Figure 4.1) to the standard basis B. $\boldsymbol{V}^\top = \boldsymbol{V}^{-1}$ performs a basis change from B to $\tilde{B}$. The red and orange vectors are now aligned with the canonical basis in the bottom-left of Figure 4.1.
2. Having changed the coordinate system to $\tilde{B}$, $\boldsymbol{\Sigma}$ scales the new coordinates by the singular values σ_i (and adds or deletes dimensions), i.e., $\boldsymbol{\Sigma}$ is the transformation matrix of Φ with respect to $\tilde{B}$ and $\tilde{C}$, represented by the red and orange vectors being stretched and lying in the $\boldsymbol{e}_1$-$\boldsymbol{e}_2$ plane, which is now embedded in a third dimension in the bottom-right of Figure 4.1.
3. $\boldsymbol{U}$ performs a basis change in the codomain $\mathbb{R}^m$ from $\tilde{C}$ into the canonical basis of $\mathbb{R}^m$, represented by a rotation of the red and orange vectors out of the $\boldsymbol{e}_1$-$\boldsymbol{e}_2$ plane. This is shown in the top-right of Figure 4.1.

The SVD expresses a change of basis in both the domain and codomain. This is in contrast with the eigendecomposition that operates within the same vector space, where the same basis change is applied and then undone. What makes the SVD special is that these two different bases are simultaneously linked by the singular value matrix $\boldsymbol{\Sigma}$.

Example 4.12 (Vectors and the SVD)
Consider a mapping of a square grid of vectors $\mathcal{X} \in \mathbb{R}^2$ that fit in a box of size 2×2 centered at the origin. Using the standard basis, we map these vectors using

$$\boldsymbol{A} = \begin{bmatrix} 1 & -0.8 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \tag{4.67a}$$

$$= \begin{bmatrix} -0.79 & 0 & -0.62 \\ 0.38 & -0.78 & -0.49 \\ -0.48 & -0.62 & 0.62 \end{bmatrix} \begin{bmatrix} 1.62 & 0 \\ 0 & 1.0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -0.78 & 0.62 \\ -0.62 & -0.78 \end{bmatrix}. \tag{4.67b}$$

We start with a set of vectors $\mathcal{X}$ (colored dots; see top-left panel of Figure 4.2) arranged in a grid. We then apply $\boldsymbol{V}^\top \in \mathbb{R}^{2\times 2}$, which rotates $\mathcal{X}$. The rotated vectors are shown in the bottom-left panel of Figure 4.2. We now map these vectors using the singular value matrix $\boldsymbol{\Sigma}$ to the codomain $\mathbb{R}^3$ (see the bottom-right panel in Figure 4.2). Note that all vectors lie in the x_1-x_2 plane. The third coordinate is always 0. The vectors in the x_1-x_2 plane have been stretched by the singular values.

The direct mapping of the vectors $\mathcal{X}$ by $\boldsymbol{A}$ to the codomain $\mathbb{R}^3$ equals the transformation of $\mathcal{X}$ by $\boldsymbol{U\Sigma V}^\top$, where $\boldsymbol{U}$ performs a rotation within the codomain $\mathbb{R}^3$ so that the mapped vectors are no longer restricted to the x_1-x_2 plane; they still are on a plane as shown in the top-right panel of Figure 4.2.

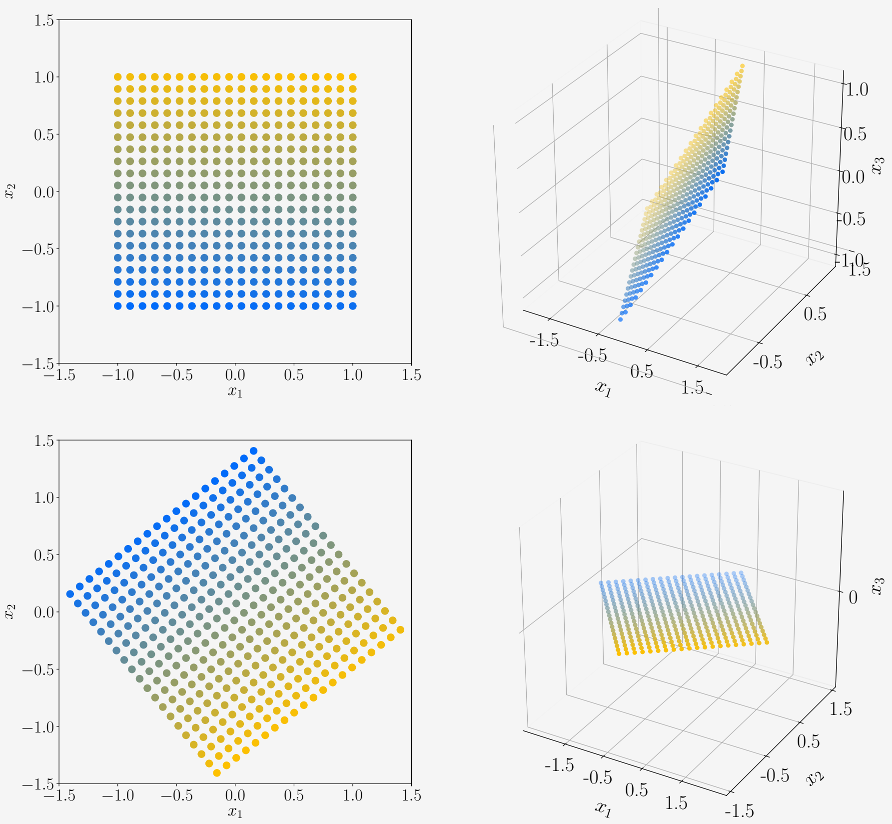


Figure 4.2 SVD and mapping of vectors (represented by discs). The panels follow the same anti-clockwise structure of Figure 4.1.

4.5.2 Construction of the SVD

We will next discuss why the SVD exists and show how to compute it in detail. The SVD of a general matrix shares some similarities with the eigendecomposition of a square matrix.

Remark. Compare the eigendecomposition of an SPD matrix

$$\boldsymbol{S} = \boldsymbol{S}^\top = \boldsymbol{PDP}^\top \tag{4.68}$$

with the corresponding SVD

$$S = U\Sigma V^\top . \tag{4.69}$$

If we set

$$U = P = V\,, \quad D = \Sigma\,, \tag{4.70}$$

we see that the SVD of SPD matrices is their eigendecomposition. $\diamondsuit$

In the following, we will explore why Theorem 4.22 holds and how the SVD is constructed. Computing the SVD of $A \in \mathbb{R}^{m\times n}$ is equivalent to finding two sets of orthonormal bases $U = (u_1, \dots, u_m)$ and $V = (v_1, \dots, v_n)$ of the codomain $\mathbb{R}^m$ and the domain $\mathbb{R}^n$, respectively. From these ordered bases, we will construct the matrices U and V.

Our plan is to start with constructing the orthonormal set of right-singular vectors $v_1, \dots, v_n \in \mathbb{R}^n$. We then construct the orthonormal set of left-singular vectors $u_1, \dots, u_m \in \mathbb{R}^m$. Thereafter, we will link the two and require that the orthogonality of the v_i is preserved under the transformation of A. This is important because we know that the images Av_i form a set of orthogonal vectors. We will then normalize these images by scalar factors, which will turn out to be the singular values.

Let us begin with constructing the right-singular vectors. The spectral theorem (Theorem 4.15) tells us that the eigenvectors of a symmetric matrix form an ONB, which also means it can be diagonalized. Moreover, from Theorem 4.14 we can always construct a symmetric, positive semidefinite matrix $A^\top A \in \mathbb{R}^{n\times n}$ from any rectangular matrix $A \in \mathbb{R}^{m\times n}$. Thus, we can always diagonalize $A^\top A$ and obtain

$$A^\top A = PDP^\top = P \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} P^\top , \tag{4.71}$$

where P is an orthogonal matrix, which is composed of the orthonormal eigenbasis. The $\lambda_i \geqslant 0$ are the eigenvalues of $A^\top A$. Let us assume the SVD of A exists and inject (4.64) into (4.71). This yields

$$A^\top A = (U\Sigma V^\top)^\top (U\Sigma V^\top) = V\Sigma^\top U^\top U\Sigma V^\top , \tag{4.72}$$

where U, V are orthogonal matrices. Therefore, with $U^\top U = I$ we obtain

$$A^\top A = V\Sigma^\top \Sigma V^\top = V \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n^2 \end{bmatrix} V^\top . \tag{4.73}$$

Comparing now (4.71) and (4.73), we identify

$$V^\top = P^\top , \tag{4.74}$$

$$\sigma_i^2 = \lambda_i . \tag{4.75}$$

Therefore, the eigenvectors of $\boldsymbol{A}^\top\boldsymbol{A}$ that compose $\boldsymbol{P}$ are the right-singular vectors $\boldsymbol{V}$ of $\boldsymbol{A}$ (see (4.74)). The eigenvalues of $\boldsymbol{A}^\top\boldsymbol{A}$ are the squared singular values of $\boldsymbol{\Sigma}$ (see (4.75)).

To obtain the left-singular vectors $\boldsymbol{U}$, we follow a similar procedure. We start by computing the SVD of the symmetric matrix $\boldsymbol{A}\boldsymbol{A}^\top \in \mathbb{R}^{m\times m}$ (instead of the previous $\boldsymbol{A}^\top\boldsymbol{A} \in \mathbb{R}^{n\times n}$). The SVD of $\boldsymbol{A}$ yields

$$\boldsymbol{A}\boldsymbol{A}^\top = (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)^\top = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{U}^\top \tag{4.76a}$$

$$= \boldsymbol{U}\begin{bmatrix}\sigma_1^2 & 0 & 0\\ 0 & \ddots & 0\\ 0 & 0 & \sigma_m^2\end{bmatrix}\boldsymbol{U}^\top . \tag{4.76b}$$

The spectral theorem tells us that $\boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{S}\boldsymbol{D}\boldsymbol{S}^\top$ can be diagonalized and we can find an ONB of eigenvectors of $\boldsymbol{A}\boldsymbol{A}^\top$, which are collected in $\boldsymbol{S}$. The orthonormal eigenvectors of $\boldsymbol{A}\boldsymbol{A}^\top$ are the left-singular vectors $\boldsymbol{U}$ and form an orthonormal basis in the codomain of the SVD.

This leaves the question of the structure of the matrix $\boldsymbol{\Sigma}$. Since $\boldsymbol{A}\boldsymbol{A}^\top$ and $\boldsymbol{A}^\top\boldsymbol{A}$ have the same nonzero eigenvalues (see page 106), the nonzero entries of the $\boldsymbol{\Sigma}$ matrices in the SVD for both cases have to be the same.

The last step is to link up all the parts we touched upon so far. We have an orthonormal set of right-singular vectors in $\boldsymbol{V}$. To finish the construction of the SVD, we connect them with the orthonormal vectors $\boldsymbol{U}$. To reach this goal, we use the fact the images of the $\boldsymbol{v}_i$ under $\boldsymbol{A}$ have to be orthogonal, too. We can show this by using the results from Section 3.4. We require that the inner product between $\boldsymbol{A}\boldsymbol{v}_i$ and $\boldsymbol{A}\boldsymbol{v}_j$ must be 0 for $i \neq j$. For any two orthogonal eigenvectors $\boldsymbol{v}_i, \boldsymbol{v}_j$, $i \neq j$, it holds that

$$(\boldsymbol{A}\boldsymbol{v}_i)^\top(\boldsymbol{A}\boldsymbol{v}_j) = \boldsymbol{v}_i^\top(\boldsymbol{A}^\top\boldsymbol{A})\boldsymbol{v}_j = \boldsymbol{v}_i^\top(\lambda_j\boldsymbol{v}_j) = \lambda_j\boldsymbol{v}_i^\top\boldsymbol{v}_j = 0 . \tag{4.77}$$

For the case $m \geqslant r$, it holds that $\{\boldsymbol{A}\boldsymbol{v}_1, \ldots, \boldsymbol{A}\boldsymbol{v}_r\}$ is a basis of an r-dimensional subspace of $\mathbb{R}^m$.

To complete the SVD construction, we need left-singular vectors that are ortho*normal*: We normalize the images of the right-singular vectors $\boldsymbol{A}\boldsymbol{v}_i$ and obtain

$$\boldsymbol{u}_i := \frac{\boldsymbol{A}\boldsymbol{v}_i}{\|\boldsymbol{A}\boldsymbol{v}_i\|} = \frac{1}{\sqrt{\lambda_i}}\boldsymbol{A}\boldsymbol{v}_i = \frac{1}{\sigma_i}\boldsymbol{A}\boldsymbol{v}_i , \tag{4.78}$$

where the last equality was obtained from (4.75) and (4.76b), showing us that the eigenvalues of $\boldsymbol{A}\boldsymbol{A}^\top$ are such that $\sigma_i^2 = \lambda_i$.

Therefore, the eigenvectors of $\boldsymbol{A}^\top\boldsymbol{A}$, which we know are the right-singular vectors $\boldsymbol{v}_i$, and their normalized images under $\boldsymbol{A}$, the left-singular vectors $\boldsymbol{u}_i$, form two self-consistent ONBs that are connected through the singular value matrix $\boldsymbol{\Sigma}$.

singular value equation

Let us rearrange (4.78) to obtain the *singular value equation*

$$\boldsymbol{A}\boldsymbol{v}_i = \sigma_i\boldsymbol{u}_i , \quad i = 1, \ldots, r . \tag{4.79}$$

This equation closely resembles the eigenvalue equation (4.25), but the vectors on the left- and the right-hand sides are not the same.

For $n < m$, (4.79) holds only for $i \leqslant n$, but (4.79) says nothing about the $\boldsymbol{u}_i$ for $i > n$. However, we know by construction that they are orthonormal. Conversely, for $m < n$, (4.79) holds only for $i \leqslant m$. For $i > m$, we have $\boldsymbol{A}\boldsymbol{v}_i = \boldsymbol{0}$ and we still know that the $\boldsymbol{v}_i$ form an orthonormal set. This means that the SVD also supplies an orthonormal basis of the kernel (null space) of $\boldsymbol{A}$, the set of vectors $\boldsymbol{x}$ with $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$ (see Section 2.7.3).

Concatenating the $\boldsymbol{v}_i$ as the columns of $\boldsymbol{V}$ and the $\boldsymbol{u}_i$ as the columns of $\boldsymbol{U}$ yields

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma}\,, \tag{4.80}$$

where $\boldsymbol{\Sigma}$ has the same dimensions as $\boldsymbol{A}$ and a diagonal structure for rows $1, \dots, r$. Hence, right-multiplying with $\boldsymbol{V}^\top$ yields $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, which is the SVD of $\boldsymbol{A}$.

Example 4.13 (Computing the SVD)
Let us find the singular value decomposition of

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix}. \tag{4.81}$$

The SVD requires us to compute the right-singular vectors $\boldsymbol{v}_j$, the singular values σ_k, and the left-singular vectors $\boldsymbol{u}_i$.

Step 1: Right-singular vectors as the eigenbasis of $\boldsymbol{A}^\top\boldsymbol{A}$.
We start by computing

$$\boldsymbol{A}^\top\boldsymbol{A} = \begin{bmatrix} 1 & -2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 5 & -2 & 1 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}. \tag{4.82}$$

We compute the singular values and right-singular vectors $\boldsymbol{v}_j$ through the eigenvalue decomposition of $\boldsymbol{A}^\top\boldsymbol{A}$, which is given as

$$\boldsymbol{A}^\top\boldsymbol{A} = \begin{bmatrix} \frac{5}{\sqrt{30}} & 0 & \frac{-1}{\sqrt{6}} \\ \frac{-2}{\sqrt{30}} & \frac{1}{\sqrt{5}} & \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} 6 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{5}{\sqrt{30}} & \frac{-2}{\sqrt{30}} & \frac{1}{\sqrt{30}} \\ 0 & \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{-1}{\sqrt{6}} & \frac{-2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{bmatrix} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^\top\,, \tag{4.83}$$

and we obtain the right-singular vectors as the columns of $\boldsymbol{P}$ so that

$$\boldsymbol{V} = \boldsymbol{P} = \begin{bmatrix} \frac{5}{\sqrt{30}} & 0 & \frac{-1}{\sqrt{6}} \\ \frac{-2}{\sqrt{30}} & \frac{1}{\sqrt{5}} & \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{6}} \end{bmatrix}. \tag{4.84}$$

Step 2: Singular-value matrix.
As the singular values σ_i are the square roots of the eigenvalues of

$\boldsymbol{A}^\top\boldsymbol{A}$ we obtain them straight from $\boldsymbol{D}$. Since $\mathrm{rk}(\boldsymbol{A}) = 2$, there are only two nonzero singular values: $\sigma_1 = \sqrt{6}$ and $\sigma_2 = 1$. The singular value matrix must be the same size as $\boldsymbol{A}$, and we obtain

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sqrt{6} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} . \tag{4.85}$$

Step 3: Left-singular vectors as the normalized image of the right-singular vectors.

We find the left-singular vectors by computing the image of the right-singular vectors under $\boldsymbol{A}$ and normalizing them by dividing them by their corresponding singular value. We obtain

$$\boldsymbol{u}_1 = \frac{1}{\sigma_1}\boldsymbol{A}\boldsymbol{v}_1 = \frac{1}{\sqrt{6}}\begin{bmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix}\begin{bmatrix} \frac{5}{\sqrt{30}} \\ \frac{-2}{\sqrt{30}} \\ \frac{1}{\sqrt{30}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{5}} \\ -\frac{2}{\sqrt{5}} \end{bmatrix} , \tag{4.86}$$

$$\boldsymbol{u}_2 = \frac{1}{\sigma_2}\boldsymbol{A}\boldsymbol{v}_2 = \frac{1}{1}\begin{bmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix} , \tag{4.87}$$

$$\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2] = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix} . \tag{4.88}$$

Note that on a computer the approach illustrated here has poor numerical behavior, and the SVD of $\boldsymbol{A}$ is normally computed without resorting to the eigenvalue decomposition of $\boldsymbol{A}^\top\boldsymbol{A}$.

4.5.3 Eigenvalue Decomposition vs. Singular Value Decomposition

Let us consider the eigendecomposition $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}$ and the SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ and review the core elements of the past sections.

- The SVD always exists for any matrix $\mathbb{R}^{m\times n}$. The eigendecomposition is only defined for square matrices $\mathbb{R}^{n\times n}$ and only exists if we can find a basis of eigenvectors of $\mathbb{R}^n$.
- The vectors in the eigendecomposition matrix $\boldsymbol{P}$ are not necessarily orthogonal, i.e., the change of basis is not a simple rotation and scaling. On the other hand, the vectors in the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ in the SVD are orthonormal, so they do represent rotations.
- Both the eigendecomposition and the SVD are compositions of three linear mappings:
  1. Change of basis in the domain
  2. Independent scaling of each new basis vector and mapping from domain to codomain
  3. Change of basis in the codomain

$$\begin{array}{rc} & \begin{array}{ccc}\text{Ali} & \text{Beatrix} & \text{Chandra}\end{array} \\ \begin{array}{r}\text{Star Wars}\\ \text{Blade Runner}\\ \text{Amelie}\\ \text{Delicatessen}\end{array} & \begin{bmatrix}5&4&1\\5&5&0\\0&0&5\\1&0&4\end{bmatrix}\end{array} = \begin{bmatrix}-0.6710 & 0.0236 & 0.4647 & -0.5774\\ -0.7197 & 0.2054 & -0.4759 & 0.4619\\ -0.0939 & -0.7705 & -0.5268 & -0.3464\\ -0.1515 & -0.6030 & 0.5293 & -0.5774\end{bmatrix}$$

$$\begin{bmatrix}9.6438 & 0 & 0\\ 0 & 6.3639 & 0\\ 0 & 0 & 0.7056\\ 0 & 0 & 0\end{bmatrix}$$

$$\begin{bmatrix}-0.7367 & -0.6515 & -0.1811\\ 0.0852 & 0.1762 & -0.9807\\ 0.6708 & -0.7379 & -0.0743\end{bmatrix}$$

Figure 4.1 Movie ratings of three people for four movies and its SVD decomposition.

A key difference between the eigendecomposition and the SVD is that in the SVD, domain and codomain can be vector spaces of different dimensions.

- In the SVD, the left- and right-singular vector matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are generally not inverse of each other (they perform basis changes in different vector spaces). In the eigendecomposition, the basis change matrices $\boldsymbol{P}$ and $\boldsymbol{P}^{-1}$ are inverses of each other.
- In the SVD, the entries in the diagonal matrix $\boldsymbol{\Sigma}$ are all real and nonnegative, which is not generally true for the diagonal matrix in the eigendecomposition.
- The SVD and the eigendecomposition are closely related through their projections
  - The left-singular vectors of $\boldsymbol{A}$ are eigenvectors of $\boldsymbol{A}\boldsymbol{A}^\top$
  - The right-singular vectors of $\boldsymbol{A}$ are eigenvectors of $\boldsymbol{A}^\top\boldsymbol{A}$.
  - The nonzero singular values of $\boldsymbol{A}$ are the square roots of the nonzero eigenvalues of both $\boldsymbol{A}\boldsymbol{A}^\top$ and $\boldsymbol{A}^\top\boldsymbol{A}$.
- For symmetric matrices $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, the eigenvalue decomposition and the SVD are one and the same, which follows from the spectral theorem 4.15.

Example 4.14 (Finding Structure in Movie Ratings and Consumers)

Let us add a practical interpretation of the SVD by analyzing data on people and their preferred movies. Consider three viewers (Ali, Beatrix, Chandra) rating four different movies (*Star Wars*, *Blade Runner*, *Amelie*, *Delicatessen*). Their ratings are values between 0 (worst) and 5 (best) and encoded in a data matrix $\boldsymbol{A} \in \mathbb{R}^{4\times 3}$ as shown in Figure 4.1. Each row represents a movie and each column a user. Thus, the column vectors of movie ratings, one for each viewer, are $\boldsymbol{x}_{\text{Ali}}$, $\boldsymbol{x}_{\text{Beatrix}}$, $\boldsymbol{x}_{\text{Chandra}}$.

Factoring $\boldsymbol{A}$ using the SVD offers us a way to capture the relationships of how people rate movies, and especially if there is a structure linking which people like which movies. Applying the SVD to our data matrix $\boldsymbol{A}$ makes a number of assumptions:

1. All viewers rate movies consistently using the same linear mapping.
2. There are no errors or noise in the ratings.
3. We interpret the left-singular vectors $\boldsymbol{u}_i$ as stereotypical movies and the right-singular vectors $\boldsymbol{v}_j$ as stereotypical viewers.

We then make the assumption that any viewer's specific movie preferences can be expressed as a linear combination of the $\boldsymbol{v}_j$. Similarly, any movie's like-ability can be expressed as a linear combination of the $\boldsymbol{u}_i$. Therefore, a vector in the domain of the SVD can be interpreted as a viewer in the "space" of stereotypical viewers, and a vector in the codomain of the SVD correspondingly as a movie in the "space" of stereotypical movies. Let us inspect the SVD of our movie-user matrix. The first left-singular vector $\boldsymbol{u}_1$ has large absolute values for the two science fiction movies and a large first singular value (red shading in Figure 4.1). Thus, this groups a type of users with a specific set of movies (science fiction theme). Similarly, the first right-singular $\boldsymbol{v}_1$ shows large absolute values for Ali and Beatrix, who give high ratings to science fiction movies (green shading in Figure 4.1). This suggests that $\boldsymbol{v}_1$ reflects the notion of a science fiction lover.

These two "spaces" are only meaningfully spanned by the respective viewer and movie data if the data itself covers a sufficient diversity of viewers and movies.

Similarly, $\boldsymbol{u}_2$, seems to capture a French art house film theme, and $\boldsymbol{v}_2$ indicates that Chandra is close to an idealized lover of such movies. An idealized science fiction lover is a purist and only loves science fiction movies, so a science fiction lover $\boldsymbol{v}_1$ gives a rating of zero to everything but science fiction themed—this logic is implied by the diagonal substructure for the singular value matrix $\boldsymbol{\Sigma}$. A specific movie is therefore represented by how it decomposes (linearly) into its stereotypical movies. Likewise, a person would be represented by how they decompose (via linear combination) into movie themes.

It is worth to briefly discuss SVD terminology and conventions, as there are different versions used in the literature. While these differences can be confusing, the mathematics remains invariant to them.

- For convenience in notation and abstraction, we use an SVD notation where the SVD is described as having two square left- and right-singular vector matrices, but a non-square singular value matrix. Our definition (4.64) for the SVD is sometimes called the *full SVD*.

full SVD

- Some authors define the SVD a bit differently and focus on square singular matrices. Then, for $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $m \geqslant n$,

$$\underset{m\times n}{\boldsymbol{A}} = \underset{m\times n}{\boldsymbol{U}}\ \underset{n\times n}{\boldsymbol{\Sigma}}\ \underset{n\times n}{\boldsymbol{V}^{\top}}\,. \tag{4.89}$$

Sometimes this formulation is called the *reduced SVD* (e.g., Datta (2010)) or *the* SVD (e.g., Press et al. (2007)). This alternative format changes merely how the matrices are constructed but leaves the mathematical structure of the SVD unchanged. The convenience of this alternative formulation is that $\boldsymbol{\Sigma}$ is diagonal, as in the eigenvalue decomposition.

reduced SVD

- In Section 4.6, we will learn about matrix approximation techniques using the SVD, which is also called the *truncated SVD*.

truncated SVD

- It is possible to define the SVD of a rank-r matrix $\boldsymbol{A}$ so that $\boldsymbol{U}$ is an $m \times r$ matrix, $\boldsymbol{\Sigma}$ a diagonal matrix $r \times r$, and $\boldsymbol{V}$ an $r \times n$ matrix. This construction is very similar to our definition, and ensures that the diagonal matrix $\boldsymbol{\Sigma}$ has only nonzero entries along the diagonal. The main convenience of this alternative notation is that $\boldsymbol{\Sigma}$ is diagonal, as in the eigenvalue decomposition.
- A restriction that the SVD for $\boldsymbol{A}$ only applies to $m \times n$ matrices with $m > n$ is practically unnecessary. When $m < n$, the SVD decomposition will yield $\boldsymbol{\Sigma}$ with more zero columns than rows and, consequently, the singular values $\sigma_{m+1}, \ldots, \sigma_n$ are 0.

The SVD is used in a variety of applications in machine learning from least-squares problems in curve fitting to solving systems of linear equations. These applications harness various important properties of the SVD, its relation to the rank of a matrix, and its ability to approximate matrices of a given rank with lower-rank matrices. Substituting a matrix with its SVD has often the advantage of making calculation more robust to numerical rounding errors. As we will explore in the next section, the SVD's ability to approximate matrices with "simpler" matrices in a principled manner opens up machine learning applications ranging from dimensionality reduction and topic modeling to data compression and clustering.

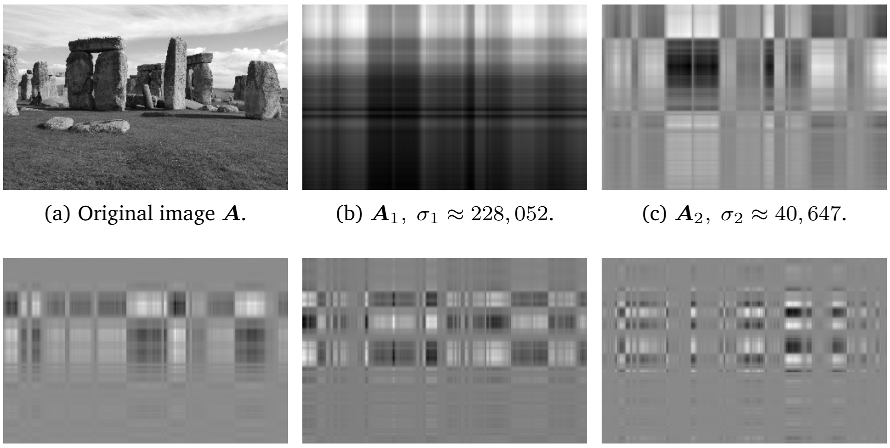
4.6 Matrix Approximation

We considered the SVD as a way to factorize $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \in \mathbb{R}^{m\times n}$ into the product of three matrices, where $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ are orthogonal and $\boldsymbol{\Sigma}$ contains the singular values on its main diagonal. Instead of doing the full SVD factorization, we will now investigate how the SVD allows us to represent a matrix $\boldsymbol{A}$ as a sum of simpler (low-rank) matrices $\boldsymbol{A}_i$, which lends itself to a matrix approximation scheme that is cheaper to compute than the full SVD.

We construct a rank-1 matrix $\boldsymbol{A}_i \in \mathbb{R}^{m\times n}$ as

$$\boldsymbol{A}_i := \boldsymbol{u}_i \boldsymbol{v}_i^\top , \tag{4.90}$$

which is formed by the outer product of the ith orthogonal column vector of $\boldsymbol{U}$ and $\boldsymbol{V}$. Figure 4.2 shows an image of Stonehenge, which can be represented by a matrix $\boldsymbol{A} \in \mathbb{R}^{1432\times 1910}$, and some outer products $\boldsymbol{A}_i$, as defined in (4.90).



(a) Original image $\boldsymbol{A}$. (b) $\boldsymbol{A}_1,\ \sigma_1 \approx 228,052$. (c) $\boldsymbol{A}_2,\ \sigma_2 \approx 40,647$.

(d) $\boldsymbol{A}_3,\ \sigma_3 \approx 26,125$. (e) $\boldsymbol{A}_4,\ \sigma_4 \approx 20,232$. (f) $\boldsymbol{A}_5,\ \sigma_5 \approx 15,436$.

Figure 4.2 Image processing with the SVD. (a) The original grayscale image is a $1,432 \times 1,910$ matrix of values between 0 (black) and 1 (white). (b)–(f) Rank-1 matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_5$ and their corresponding singular values $\sigma_1, \ldots, \sigma_5$. The grid-like structure of each rank-1 matrix is imposed by the outer-product of the left and right-singular vectors.

A matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ of rank r can be written as a sum of rank-1 matrices $\boldsymbol{A}_i$ so that

$$\boldsymbol{A} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top = \sum_{i=1}^{r} \sigma_i \boldsymbol{A}_i \,, \tag{4.91}$$

where the outer-product matrices $\boldsymbol{A}_i$ are weighted by the ith singular value σ_i. We can see why (4.91) holds: The diagonal structure of the singular value matrix $\boldsymbol{\Sigma}$ multiplies only matching left- and right-singular vectors $\boldsymbol{u}_i \boldsymbol{v}_i^\top$ and scales them by the corresponding singular value σ_i. All terms $\Sigma_{ij} \boldsymbol{u}_i \boldsymbol{v}_j^\top$ vanish for $i \neq j$ because $\boldsymbol{\Sigma}$ is a diagonal matrix. Any terms $i > r$ vanish because the corresponding singular values are 0.

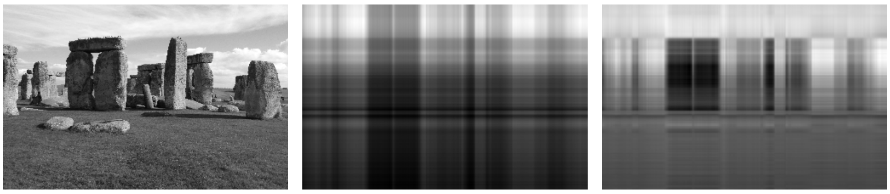
In (4.90), we introduced rank-1 matrices $\boldsymbol{A}_i$. We summed up the r individual rank-1 matrices to obtain a rank-r matrix $\boldsymbol{A}$; see (4.91). If the sum does not run over all matrices $\boldsymbol{A}_i$, $i = 1, \ldots, r$, but only up to an intermediate value $k < r$, we obtain a *rank-k approximation*

rank-k approximation

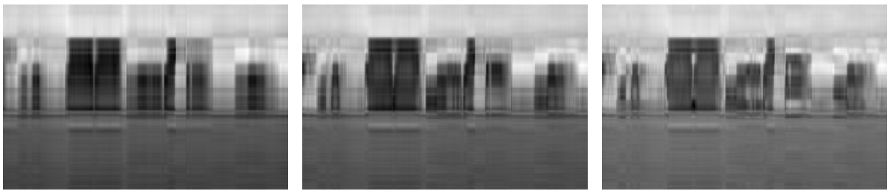
$$\widehat{\boldsymbol{A}}(k) := \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top = \sum_{i=1}^{k} \sigma_i \boldsymbol{A}_i \tag{4.92}$$

of $\boldsymbol{A}$ with $\mathrm{rk}(\widehat{\boldsymbol{A}}(k)) = k$. Figure 4.3 shows low-rank approximations $\widehat{\boldsymbol{A}}(k)$ of an original image $\boldsymbol{A}$ of Stonehenge. The shape of the rocks becomes increasingly visible and clearly recognizable in the rank-5 approximation. While the original image requires $1,432 \cdot 1,910 = 2,735,120$ numbers, the rank-5 approximation requires us only to store the five singular values and the five left- and right-singular vectors ($1,432$ and $1,910$-dimensional each) for a total of $5 \cdot (1,432 + 1,910 + 1) = 16,715$ numbers – just above 0.6% of the original.

To measure the difference (error) between $\boldsymbol{A}$ and its rank-k approximation $\widehat{\boldsymbol{A}}(k)$, we need the notion of a norm. In Section 3.1, we already used



(a) Original image $\boldsymbol{A}$. (b) Rank-1 approximation $\widehat{\boldsymbol{A}}(1)$. (c) Rank-2 approximation $\widehat{\boldsymbol{A}}(2)$.



(d) Rank-3 approximation $\widehat{\boldsymbol{A}}(3)$. (e) Rank-4 approximation $\widehat{\boldsymbol{A}}(4)$. (f) Rank-5 approximation $\widehat{\boldsymbol{A}}(5)$.

Figure 4.3 Image reconstruction with the SVD. (a) Original image. (b)–(f) Image reconstruction using the low-rank approximation of the SVD, where the rank-k approximation is given by $\widehat{\boldsymbol{A}}(k) = \sum_{i=1}^{k} \sigma_i \boldsymbol{A}_i$.

norms on vectors that measure the length of a vector. By analogy we can also define norms on matrices.

spectral norm

Definition 4.23 (Spectral Norm of a Matrix)**.** For $\boldsymbol{x} \in \mathbb{R}^n \backslash \{\boldsymbol{0}\}$, the *spectral norm* of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is defined as

$$\|\boldsymbol{A}\|_2 := \max_{\boldsymbol{x}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \,. \tag{4.93}$$

We introduce the notation of a subscript in the matrix norm (left-hand side), similar to the Euclidean norm for vectors (right-hand side), which has subscript 2. The spectral norm (4.93) determines how long any vector $\boldsymbol{x}$ can at most become when multiplied by $\boldsymbol{A}$.

Theorem 4.24. *The spectral norm of $\boldsymbol{A}$ is its largest singular value σ_1.*

We leave the proof of this theorem as an exercise.

Eckart-Young theorem

Theorem 4.25 (Eckart-Young Theorem (Eckart and Young, 1936))**.** *Consider a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ of rank r and let $\boldsymbol{B} \in \mathbb{R}^{m \times n}$ be a matrix of rank k. For any $k \leqslant r$ with $\widehat{\boldsymbol{A}}(k) = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$ it holds that*

$$\widehat{\boldsymbol{A}}(k) = \operatorname{argmin}_{\operatorname{rk}(\boldsymbol{B})=k} \|\boldsymbol{A} - \boldsymbol{B}\|_2 \,, \tag{4.94}$$

$$\left\|\boldsymbol{A} - \widehat{\boldsymbol{A}}(k)\right\|_2 = \sigma_{k+1} \,. \tag{4.95}$$

The Eckart-Young theorem states explicitly how much error we introduce by approximating $\boldsymbol{A}$ using a rank-k approximation. We can interpret the rank-k approximation obtained with the SVD as a projection of the full-rank matrix $\boldsymbol{A}$ onto a lower-dimensional space of rank-at-most-k matrices. Of all possible projections, the SVD minimizes the error (with respect to the spectral norm) between $\boldsymbol{A}$ and any rank-k approximation.

We can retrace some of the steps to understand why (4.95) should hold.

We observe that the difference between $\boldsymbol{A} - \widehat{\boldsymbol{A}}(k)$ is a matrix containing the sum of the remaining rank-1 matrices

$$\boldsymbol{A} - \widehat{\boldsymbol{A}}(k) = \sum_{i=k+1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top . \tag{4.96}$$

By Theorem 4.24, we immediately obtain σ_{k+1} as the spectral norm of the difference matrix. Let us have a closer look at (4.94). If we assume that there is another matrix $\boldsymbol{B}$ with $\mathrm{rk}(\boldsymbol{B}) \leqslant k$, such that

$$\|\boldsymbol{A} - \boldsymbol{B}\|_2 < \left\|\boldsymbol{A} - \widehat{\boldsymbol{A}}(k)\right\|_2 , \tag{4.97}$$

then there exists an at least $(n-k)$-dimensional null space $Z \subseteq \mathbb{R}^n$, such that $\boldsymbol{x} \in Z$ implies that $\boldsymbol{Bx} = \boldsymbol{0}$. Then it follows that

$$\|\boldsymbol{Ax}\|_2 = \|(\boldsymbol{A} - \boldsymbol{B})\boldsymbol{x}\|_2 , \tag{4.98}$$

and by using a version of the Cauchy-Schwartz inequality (3.17) that encompasses norms of matrices, we obtain

$$\|\boldsymbol{Ax}\|_2 \leqslant \|\boldsymbol{A} - \boldsymbol{B}\|_2 \|\boldsymbol{x}\|_2 < \sigma_{k+1} \|\boldsymbol{x}\|_2 . \tag{4.99}$$

However, there exists a $(k+1)$-dimensional subspace where $\|\boldsymbol{Ax}\|_2 \geqslant \sigma_{k+1} \|\boldsymbol{x}\|_2$, which is spanned by the right-singular vectors $\boldsymbol{v}_j, j \leqslant k+1$ of $\boldsymbol{A}$. Adding up dimensions of these two spaces yields a number greater than n, as there must be a nonzero vector in both spaces. This is a contradiction of the rank-nullity theorem (Theorem 2.24) in Section 2.7.3.

The Eckart-Young theorem implies that we can use SVD to reduce a rank-r matrix $\boldsymbol{A}$ to a rank-k matrix $\widehat{\boldsymbol{A}}$ in a principled, optimal (in the spectral norm sense) manner. We can interpret the approximation of $\boldsymbol{A}$ by a rank-k matrix as a form of lossy compression. Therefore, the low-rank approximation of a matrix appears in many machine learning applications, e.g., image processing, noise filtering, and regularization of ill-posed problems. Furthermore, it plays a key role in dimensionality reduction and principal component analysis, as we will see in Chapter 10.

Example 4.15 (Finding Structure in Movie Ratings and Consumers (continued))

Coming back to our movie-rating example, we can now apply the concept of low-rank approximations to approximate the original data matrix. Recall that our first singular value captures the notion of science fiction theme in movies and science fiction lovers. Thus, by using only the first singular value term in a rank-1 decomposition of the movie-rating matrix, we obtain the predicted ratings

$$\boldsymbol{A}_1 = \boldsymbol{u}_1 \boldsymbol{v}_1^\top = \begin{bmatrix} -0.6710 \\ -0.7197 \\ -0.0939 \\ -0.1515 \end{bmatrix} \begin{bmatrix} -0.7367 & -0.6515 & -0.1811 \end{bmatrix} \tag{4.100a}$$

$$= \begin{bmatrix} 0.4943 & 0.4372 & 0.1215 \\ 0.5302 & 0.4689 & 0.1303 \\ 0.0692 & 0.0612 & 0.0170 \\ 0.1116 & 0.0987 & 0.0274 \end{bmatrix}. \tag{4.100b}$$

This first rank-1 approximation $\boldsymbol{A}_1$ is insightful: it tells us that Ali and Beatrix like science fiction movies, such as *Star Wars* and *Bladerunner* (entries have values > 0.4), but fails to capture the ratings of the other movies by Chandra. This is not surprising, as Chandra's type of movies is not captured by the first singular value. The second singular value gives us a better rank-1 approximation for those movie-theme lovers:

$$\boldsymbol{A}_2 = \boldsymbol{u}_2\boldsymbol{v}_2^\top = \begin{bmatrix} 0.0236 \\ 0.2054 \\ -0.7705 \\ -0.6030 \end{bmatrix} \begin{bmatrix} 0.0852 & 0.1762 & -0.9807 \end{bmatrix} \tag{4.101a}$$

$$= \begin{bmatrix} 0.0020 & 0.0042 & -0.0231 \\ 0.0175 & 0.0362 & -0.2014 \\ -0.0656 & -0.1358 & 0.7556 \\ -0.0514 & -0.1063 & 0.5914 \end{bmatrix}. \tag{4.101b}$$

In this second rank-1 approximation $\boldsymbol{A}_2$, we capture Chandra's ratings and movie types well, but not the science fiction movies. This leads us to consider the rank-2 approximation $\widehat{\boldsymbol{A}}(2)$, where we combine the first two rank-1 approximations

$$\widehat{\boldsymbol{A}}(2) = \sigma_1\boldsymbol{A}_1 + \sigma_2\boldsymbol{A}_2 = \begin{bmatrix} 4.7801 & 4.2419 & 1.0244 \\ 5.2252 & 4.7522 & -0.0250 \\ 0.2493 & -0.2743 & 4.9724 \\ 0.7495 & 0.2756 & 4.0278 \end{bmatrix}. \tag{4.102}$$

$\widehat{\boldsymbol{A}}(2)$ is similar to the original movie ratings table

$$\boldsymbol{A} = \begin{bmatrix} 5 & 4 & 1 \\ 5 & 5 & 0 \\ 0 & 0 & 5 \\ 1 & 0 & 4 \end{bmatrix}, \tag{4.103}$$

and this suggests that we can ignore the contribution of $\boldsymbol{A}_3$. We can interpret this so that in the data table there is no evidence of a third movie-theme/movie-lovers category. This also means that the entire space of movie-themes/movie-lovers in our example is a two-dimensional space spanned by science fiction and French art house movies and lovers.

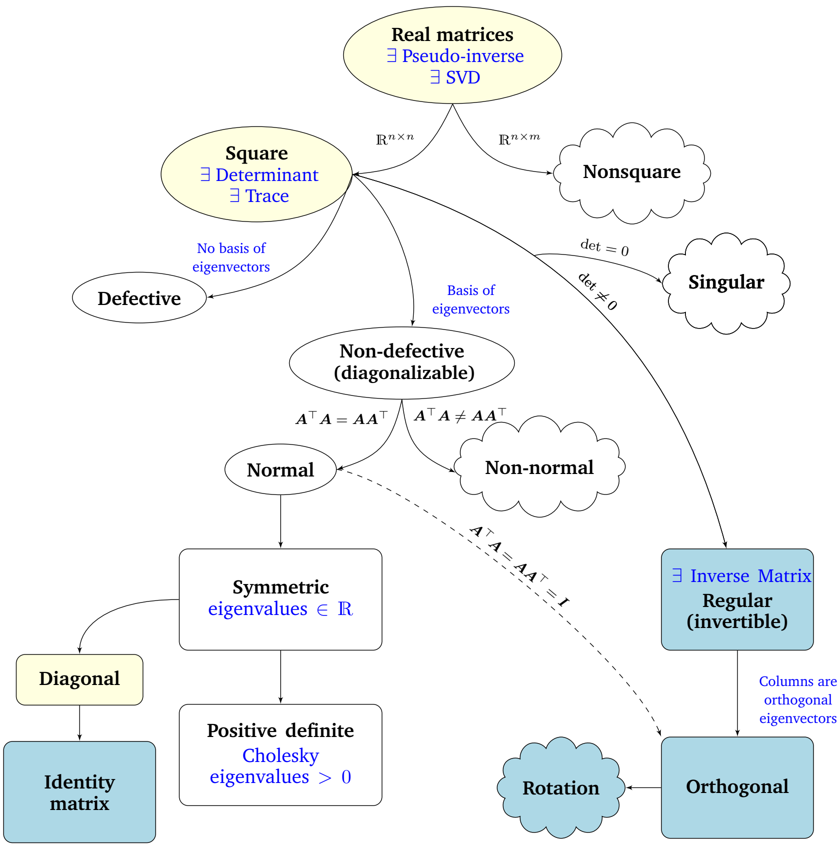


Figure 4.2 A functional phylogeny of matrices encountered in machine learning.

4.7 Matrix Phylogeny

In Chapters 2 and 3, we covered the basics of linear algebra and analytic geometry. In this chapter, we looked at fundamental characteristics of matrices and linear mappings. Figure 4.2 depicts the phylogenetic tree of relationships between different types of matrices (black arrows indicating "is a subset of") and the covered operations we can perform on them (in blue). We consider all *real matrices* $\boldsymbol{A} \in \mathbb{R}^{n\times m}$. For non-square matrices (where $n \neq m$), the SVD always exists, as we saw in this chapter. Focusing on *square matrices* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, the *determinant* informs us whether a square matrix possesses an *inverse matrix*, i.e., whether it belongs to the class of regular, invertible matrices. If the square $n \times n$ matrix possesses n linearly independent eigenvectors, then the matrix is *non-defective* and an *eigendecomposition* exists (Theorem 4.12). We know that repeated eigenvalues may result in defective matrices, which cannot be diagonalized.

The word "phylogenetic" describes how we capture the relationships among individuals or groups and derived from the Greek words for "tribe" and "source".

Non-singular and non-defective matrices are not the same. For example, a rotation matrix will be invertible (determinant is nonzero) but not diagonalizable in the real numbers (eigenvalues are not guaranteed to be real numbers).

We dive further into the branch of non-defective square $n \times n$ matrices. $\boldsymbol{A}$ is *normal* if the condition $\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^\top$ holds. Moreover, if the more restrictive condition holds that $\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{I}$, then $\boldsymbol{A}$ is called *orthogona*l (see Definition 3.8). The set of orthogonal matrices is a subset of the regular (invertible) matrices and satisfies $\boldsymbol{A}^\top = \boldsymbol{A}^{-1}$.

Normal matrices have a frequently encountered subset, the symmetric matrices $\boldsymbol{S} \in \mathbb{R}^{n \times n}$, which satisfy $\boldsymbol{S} = \boldsymbol{S}^\top$. Symmetric matrices have only real eigenvalues. A subset of the symmetric matrices consists of the positive definite matrices $\boldsymbol{P}$ that satisfy the condition of $\boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} > 0$ for all $\boldsymbol{x} \in \mathbb{R}^n \backslash \{\boldsymbol{0}\}$. In this case, a unique *Cholesky decomposition* exists (Theorem 4.18). Positive definite matrices have only positive eigenvalues and are always invertible (i.e., have a nonzero determinant).

Another subset of symmetric matrices consists of the *diagonal matrices* $\boldsymbol{D}$. Diagonal matrices are closed under multiplication and addition, but do not necessarily form a group (this is only the case if all diagonal entries are nonzero so that the matrix is invertible). A special diagonal matrix is the identity matrix $\boldsymbol{I}$.

4.8 Further Reading

Most of the content in this chapter establishes underlying mathematics and connects them to methods for studying mappings, many of which are at the heart of machine learning at the level of underpinning software solutions and building blocks for almost all machine learning theory. Matrix characterization using determinants, eigenspectra, and eigenspaces provides fundamental features and conditions for categorizing and analyzing matrices. This extends to all forms of representations of data and mappings involving data, as well as judging the numerical stability of computational operations on such matrices (Press et al., 2007).

Determinants are fundamental tools in order to invert matrices and compute eigenvalues "by hand". However, for almost all but the smallest instances, numerical computation by Gaussian elimination outperforms determinants (Press et al., 2007). Determinants remain nevertheless a powerful theoretical concept, e.g., to gain intuition about the orientation of a basis based on the sign of the determinant. Eigenvectors can be used to perform basis changes to transform data into the coordinates of meaningful orthogonal, feature vectors. Similarly, matrix decomposition methods, such as the Cholesky decomposition, reappear often when we compute or simulate random events (Rubinstein and Kroese, 2016). Therefore, the Cholesky decomposition enables us to compute the *reparametrization trick* where we want to perform continuous differentiation over random variables, e.g., in variational autoencoders (Jimenez Rezende et al., 2014; Kingma and Welling, 2014).

Eigendecomposition is fundamental in enabling us to extract meaningful and interpretable information that characterizes linear mappings.

Therefore, the eigendecomposition underlies a general class of machine learning algorithms called *spectral methods* that perform eigendecomposition of a positive-definite kernel. These spectral decomposition methods encompass classical approaches to statistical data analysis, such as the following:

principal component analysis

- *Principal component analysis* (PCA (Pearson, 1901), see also Chapter 10), in which a low-dimensional subspace, which explains most of the variability in the data, is sought.

Fisher discriminant analysis

- *Fisher discriminant analysis*, which aims to determine a separating hyperplane for data classification (Mika et al., 1999).

multidimensional scaling

- *Multidimensional scaling* (MDS) (Carroll and Chang, 1970).

Isomap

Laplacian eigenmaps

Hessian eigenmaps

spectral clustering

The computational efficiency of these methods typically comes from finding the best rank-k approximation to a symmetric, positive semidefinite matrix. More contemporary examples of spectral methods have different origins, but each of them requires the computation of the eigenvectors and eigenvalues of a positive-definite kernel, such as *Isomap* (Tenenbaum et al., 2000), *Laplacian eigenmaps* (Belkin and Niyogi, 2003), *Hessian eigenmaps* (Donoho and Grimes, 2003), and *spectral clustering* (Shi and Malik, 2000). The core computations of these are generally underpinned by low-rank matrix approximation techniques (Belabbas and Wolfe, 2009) as we encountered here via the SVD.

The SVD allows us to discover some of the same kind of information as the eigendecomposition. However, the SVD is more generally applicable to non-square matrices and data tables. These matrix factorization methods become relevant whenever we want to identify heterogeneity in data when we want to perform data compression by approximation, e.g., instead of storing $n \times m$ values just storing $(n+m)k$ values, or when we want to perform data pre-processing, e.g., to decorrelate predictor variables of a design matrix (Ormoneit et al., 2001). The SVD operates on matrices, which we can interpret as rectangular arrays with two indices (rows and columns). The extension of matrix-like structure to higher-dimensional arrays are called tensors. It turns out that the SVD is the special case of a more general family of decompositions that operate on such tensors (Kolda and Bader, 2009). SVD-like operations and low-rank approximations on tensors are, for example, the *Tucker decomposition* (Tucker, 1966) or the *CP decomposition* (Carroll and Chang, 1970).

Tucker decomposition

CP decomposition

The SVD low-rank approximation is frequently used in machine learning for computational efficiency reasons. This is because it reduces the amount of memory and operations with nonzero multiplications we need to perform on potentially very large matrices of data (Trefethen and Bau III, 1997). Moreover, low-rank approximations are used to operate on matrices that may contain missing values as well as for purposes of lossy compression and dimensionality reduction (Moonen and De Moor, 1995; Markovsky, 2011).

Exercises

4.1 Compute the determinant using the Laplace expansion (using the first row) and the Sarrus rule for

$$\boldsymbol{A} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 0 & 2 & 4 \end{bmatrix}.$$

4.2 Compute the following determinant efficiently:

$$\begin{bmatrix} 2 & 0 & 1 & 2 & 0 \\ 2 & -1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 1 & 2 \\ -2 & 0 & 2 & -1 & 2 \\ 2 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

4.3 Compute the eigenspaces of

a.

$$\boldsymbol{A} := \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

b.

$$\boldsymbol{B} := \begin{bmatrix} -2 & 2 \\ 2 & 1 \end{bmatrix}$$

4.4 Compute all eigenspaces of

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

4.5 Diagonalizability of a matrix is unrelated to its invertibility. Determine for the following four matrices whether they are diagonalizable and/or invertible

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

4.6 Compute the eigenspaces of the following transformation matrices. Are they diagonalizable?

a. For

$$\boldsymbol{A} = \begin{bmatrix} 2 & 3 & 0 \\ 1 & 4 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

b. For

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

4.7 Are the following matrices diagonalizable? If yes, determine their diagonal form and a basis with respect to which the transformation matrices are diagonal. If no, give reasons why they are not diagonalizable.

a.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -8 & 4 \end{bmatrix}$$

b.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

c.

$$\boldsymbol{A} = \begin{bmatrix} 5 & 4 & 2 & 1 \\ 0 & 1 & -1 & -1 \\ -1 & -1 & 3 & 0 \\ 1 & 1 & -1 & 2 \end{bmatrix}$$

d.

$$\boldsymbol{A} = \begin{bmatrix} 5 & -6 & -6 \\ -1 & 4 & 2 \\ 3 & -6 & -4 \end{bmatrix}$$

4.8 Find the SVD of the matrix

$$\boldsymbol{A} = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix} .$$

4.9 Find the singular value decomposition of

$$\boldsymbol{A} = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} .$$

4.10 Find the rank-1 approximation of

$$\boldsymbol{A} = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$$

4.11 Show that for any $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ the matrices $\boldsymbol{A}^\top \boldsymbol{A}$ and $\boldsymbol{A}\boldsymbol{A}^\top$ possess the same nonzero eigenvalues.

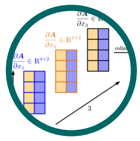
4.12 Show that for $\boldsymbol{x} \neq \boldsymbol{0}$ Theorem 4.24 holds, i.e., show that

$$\max_{\boldsymbol{x}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} = \sigma_1 ,$$

where σ_1 is the largest singular value of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$.

5

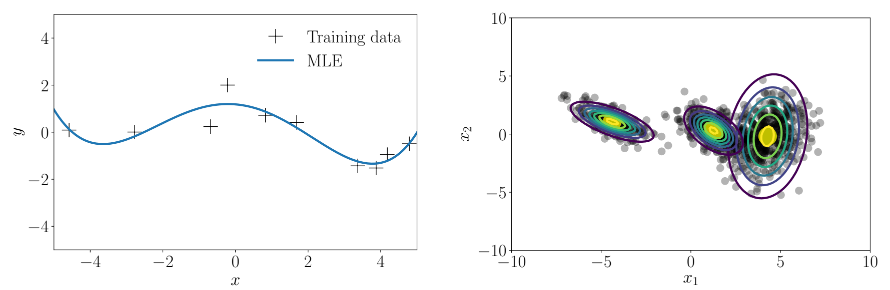
Vector Calculus



Many algorithms in machine learning optimize an objective function with respect to a set of desired model parameters that control how well a model explains the data: Finding good parameters can be phrased as an optimization problem (see Sections 8.2 and 8.3). Examples include: (i) linear regression (see Chapter 9), where we look at curve-fitting problems and optimize linear weight parameters to maximize the likelihood; (ii) neural-network auto-encoders for dimensionality reduction and data compression, where the parameters are the weights and biases of each layer, and where we minimize a reconstruction error by repeated application of the chain rule; and (iii) Gaussian mixture models (see Chapter 11) for modeling data distributions, where we optimize the location and shape parameters of each mixture component to maximize the likelihood of the model. Figure 5.1 illustrates some of these problems, which we typically solve by using optimization algorithms that exploit gradient information (Section 7.1). Figure 5.2 gives an overview of how concepts in this chapter are related and how they are connected to other chapters of the book.

Central to this chapter is the concept of a function. A function f is a quantity that relates two quantities to each other. In this book, these quantities are typically inputs $\boldsymbol{x} \in \mathbb{R}^D$ and targets (function values) $f(\boldsymbol{x})$, which we assume are real-valued if not stated otherwise. Here $\mathbb{R}^D$ is the *domain* of f, and the function values $f(\boldsymbol{x})$ are the *image/codomain* of f.

domain
image/codomain



(a) Regression problem: Find parameters, such that the curve explains the observations (crosses) well.

(b) Density estimation with a Gaussian mixture model: Find means and covariances, such that the data (dots) can be explained well.

Figure 5.1 Vector calculus plays a central role in (a) regression (curve fitting) and (b) density estimation, i.e., modeling data distributions.

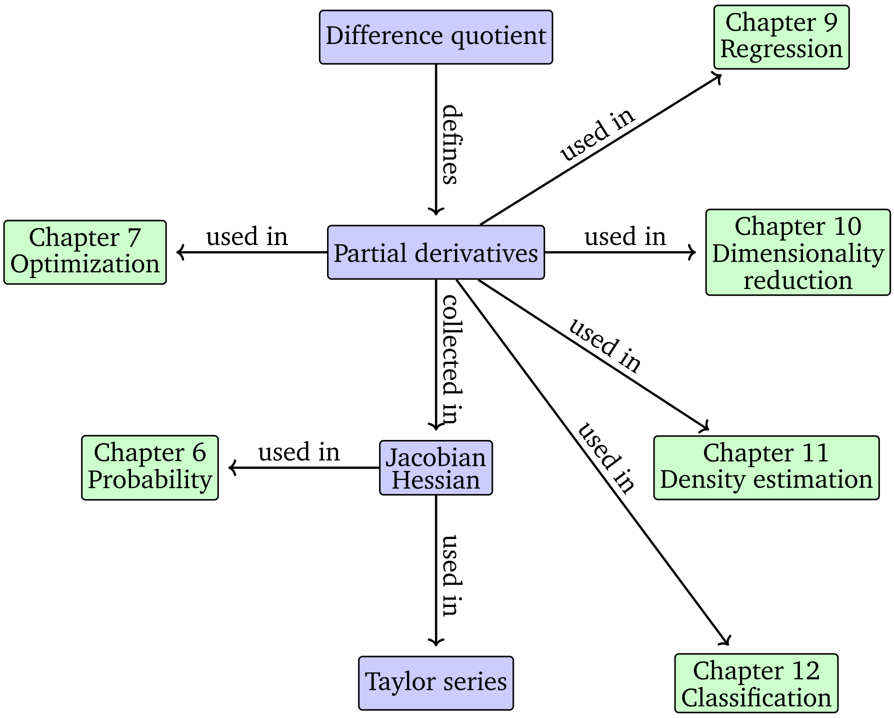


Figure 5.2 A mind map of the concepts introduced in this chapter, along with when they are used in other parts of the book.

Section 2.7.3 provides much more detailed discussion in the context of linear functions. We often write

$$f : \mathbb{R}^D \to \mathbb{R} \tag{5.1a}$$

$$\boldsymbol{x} \mapsto f(\boldsymbol{x}) \tag{5.1b}$$

to specify a function, where (5.1a) specifies that f is a mapping from $\mathbb{R}^D$ to $\mathbb{R}$ and (5.1b) specifies the explicit assignment of an input $\boldsymbol{x}$ to a function value $f(\boldsymbol{x})$. A function f assigns every input $\boldsymbol{x}$ exactly one function value $f(\boldsymbol{x})$.

Example 5.1

Recall the dot product as a special case of an inner product (Section 3.2). In the previous notation, the function $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{x}$, $\boldsymbol{x} \in \mathbb{R}^2$, would be specified as

$$f : \mathbb{R}^2 \to \mathbb{R} \tag{5.2a}$$

$$\boldsymbol{x} \mapsto x_1^2 + x_2^2 \,. \tag{5.2b}$$

In this chapter, we will discuss how to compute gradients of functions, which is often essential to facilitate learning in machine learning models since the gradient points in the direction of steepest ascent. Therefore,

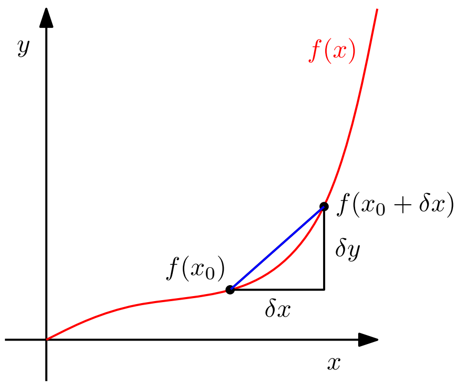


Figure 5.1 The average incline of a function f between x_0 and $x_0 + \delta x$ is the incline of the secant (blue) through $f(x_0)$ and $f(x_0 + \delta x)$ and given by $\delta y/\delta x$.

vector calculus is one of the fundamental mathematical tools we need in machine learning. Throughout this book, we assume that functions are differentiable. With some additional technical definitions, which we do not cover here, many of the approaches presented can be extended to sub-differentials (functions that are continuous but not differentiable at certain points). We will look at an extension to the case of functions with constraints in Chapter 7.

5.1 Differentiation of Univariate Functions

In the following, we briefly revisit differentiation of a univariate function, which may be familiar from high school mathematics. We start with the difference quotient of a univariate function $y = f(x),\ x, y \in \mathbb{R}$, which we will subsequently use to define derivatives.

difference quotient

Definition 5.1 (Difference Quotient)**.** The *difference quotient*

$$\frac{\delta y}{\delta x} := \frac{f(x + \delta x) - f(x)}{\delta x} \tag{5.3}$$

computes the slope of the secant line through two points on the graph of f. In Figure 5.1, these are the points with x-coordinates x_0 and $x_0 + \delta x$.

The difference quotient can also be considered the average slope of f between x and $x + \delta x$ if we assume f to be a linear function. In the limit for $\delta x \to 0$, we obtain the tangent of f at x, if f is differentiable. The tangent is then the derivative of f at x.

derivative

Definition 5.2 (Derivative)**.** More formally, for $h > 0$ the *derivative* of f at x is defined as the limit

$$\frac{\mathrm{d}f}{\mathrm{d}x} := \lim_{h \to 0} \frac{f(x + h) - f(x)}{h}, \tag{5.4}$$

and the secant in Figure 5.1 becomes a tangent.

The derivative of f points in the direction of steepest ascent of f.

Example 5.2 (Derivative of a Polynomial)
We want to compute the derivative of $f(x) = x^n, n \in \mathbb{N}$. We may already know that the answer will be nx^{n-1}, but we want to derive this result using the definition of the derivative as the limit of the difference quotient.

Using the definition of the derivative in (5.4), we obtain

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{5.5a}$$
$$= \lim_{h \to 0} \frac{(x+h)^n - x^n}{h} \tag{5.5b}$$
$$= \lim_{h \to 0} \frac{\sum_{i=0}^{n} \binom{n}{i} x^{n-i} h^i - x^n}{h} . \tag{5.5c}$$

We see that $x^n = \binom{n}{0} x^{n-0} h^0$. By starting the sum at 1, the x^n-term cancels, and we obtain

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{h \to 0} \frac{\sum_{i=1}^{n} \binom{n}{i} x^{n-i} h^i}{h} \tag{5.6a}$$
$$= \lim_{h \to 0} \sum_{i=1}^{n} \binom{n}{i} x^{n-i} h^{i-1} \tag{5.6b}$$
$$= \lim_{h \to 0} \binom{n}{1} x^{n-1} + \underbrace{\sum_{i=2}^{n} \binom{n}{i} x^{n-i} h^{i-1}}_{\to 0 \text{ as } h \to 0} \tag{5.6c}$$
$$= \frac{n!}{1!(n-1)!} x^{n-1} = nx^{n-1} . \tag{5.6d}$$

5.1.1 Taylor Series

The Taylor series is a representation of a function f as an infinite sum of terms. These terms are determined using derivatives of f evaluated at x_0.

Taylor polynomial

We define $t^0 := 1$ for all $t \in \mathbb{R}$.

Definition 5.3 (Taylor Polynomial)**.** The *Taylor polynomial* of degree n of $f : \mathbb{R} \to \mathbb{R}$ at x_0 is defined as

$$T_n(x) := \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k , \tag{5.7}$$

where $f^{(k)}(x_0)$ is the kth derivative of f at x_0 (which we assume exists) and $\frac{f^{(k)}(x_0)}{k!}$ are the coefficients of the polynomial.

Taylor series

Definition 5.4 (Taylor Series)**.** For a smooth function $f \in \mathcal{C}^\infty$, $f : \mathbb{R} \to \mathbb{R}$, the *Taylor series* of f at x_0 is defined as

$$T_\infty(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k\,. \tag{5.8}$$

For $x_0 = 0$, we obtain the *Maclaurin series* as a special instance of the Taylor series. If $f(x) = T_\infty(x)$, then f is called *analytic*.

$f \in \mathcal{C}^\infty$ means that f is continuously differentiable infinitely many times.
Maclaurin series
analytic

Remark. In general, a Taylor polynomial of degree n is an approximation of a function, which does not need to be a polynomial. The Taylor polynomial is similar to f in a neighborhood around x_0. However, a Taylor polynomial of degree n is an exact representation of a polynomial f of degree $k \leqslant n$ since all derivatives $f^{(i)}$, $i > k$ vanish. ♢

Example 5.3 (Taylor Polynomial)
We consider the polynomial

$$f(x) = x^4 \tag{5.9}$$

and seek the Taylor polynomial T_6, evaluated at $x_0 = 1$. We start by computing the coefficients $f^{(k)}(1)$ for $k = 0, \ldots, 6$:

$$f(1) = 1 \tag{5.10}$$
$$f'(1) = 4 \tag{5.11}$$
$$f''(1) = 12 \tag{5.12}$$
$$f^{(3)}(1) = 24 \tag{5.13}$$
$$f^{(4)}(1) = 24 \tag{5.14}$$
$$f^{(5)}(1) = 0 \tag{5.15}$$
$$f^{(6)}(1) = 0 \tag{5.16}$$

Therefore, the desired Taylor polynomial is

$$T_6(x) = \sum_{k=0}^{6} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k \tag{5.17a}$$
$$= 1 + 4(x-1) + 6(x-1)^2 + 4(x-1)^3 + (x-1)^4 + 0\,. \tag{5.17b}$$

Multiplying out and re-arranging yields

$$T_6(x) = (1 - 4 + 6 - 4 + 1) + x(4 - 12 + 12 - 4)$$
$$+ x^2(6 - 12 + 6) + x^3(4 - 4) + x^4 \tag{5.18a}$$
$$= x^4 = f(x)\,, \tag{5.18b}$$

i.e., we obtain an exact representation of the original function.

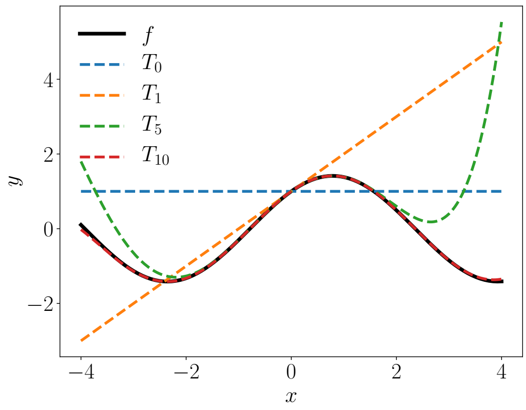


Figure 5.2 Taylor polynomials. The original function $f(x) = \sin(x) + \cos(x)$ (black, solid) is approximated by Taylor polynomials (dashed) around $x_0 = 0$. Higher-order Taylor polynomials approximate the function f better and more globally. T_{10} is already similar to f in $[-4, 4]$.

Example 5.4 (Taylor Series)
Consider the function in Figure 5.2 given by

$$f(x) = \sin(x) + \cos(x) \in \mathcal{C}^\infty . \tag{5.19}$$

We seek a Taylor series expansion of f at $x_0 = 0$, which is the Maclaurin series expansion of f. We obtain the following derivatives:

$$f(0) = \sin(0) + \cos(0) = 1 \tag{5.20}$$
$$f'(0) = \cos(0) - \sin(0) = 1 \tag{5.21}$$
$$f''(0) = -\sin(0) - \cos(0) = -1 \tag{5.22}$$
$$f^{(3)}(0) = -\cos(0) + \sin(0) = -1 \tag{5.23}$$
$$f^{(4)}(0) = \sin(0) + \cos(0) = f(0) = 1 \tag{5.24}$$
$$\vdots$$

We can see a pattern here: The coefficients in our Taylor series are only ± 1 (since $\sin(0) = 0$), each of which occurs twice before switching to the other one. Furthermore, $f^{(k+4)}(0) = f^{(k)}(0)$.

Therefore, the full Taylor series expansion of f at $x_0 = 0$ is given by

$$\begin{aligned}
T_\infty(x) &= \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k & (5.25a)\\
&= 1 + x - \frac{1}{2!}x^2 - \frac{1}{3!}x^3 + \frac{1}{4!}x^4 + \frac{1}{5!}x^5 - \cdots & (5.25b)\\
&= 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 \mp \cdots + x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 \mp \cdots & (5.25c)\\
&= \sum_{k=0}^{\infty} (-1)^k \frac{1}{(2k)!}x^{2k} + \sum_{k=0}^{\infty} (-1)^k \frac{1}{(2k+1)!}x^{2k+1} & (5.25d)\\
&= \cos(x) + \sin(x) , & (5.25e)
\end{aligned}$$

where we used the *power series representations*

power series representation

$$\cos(x) = \sum_{k=0}^{\infty} (-1)^k \frac{1}{(2k)!} x^{2k}, \tag{5.26}$$

$$\sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{1}{(2k+1)!} x^{2k+1}. \tag{5.27}$$

Figure 5.2 shows the corresponding first Taylor polynomials T_n for $n = 0, 1, 5, 10$.

Remark. A Taylor series is a special case of a power series

$$f(x) = \sum_{k=0}^{\infty} a_k (x - c)^k \tag{5.28}$$

where a_k are coefficients and c is a constant, which has the special form in Definition 5.4. ♢

5.1.2 Differentiation Rules

In the following, we briefly state basic differentiation rules, where we denote the derivative of f by f'.

Product rule: $$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x) \tag{5.29}$$

Quotient rule: $$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - f(x)g'(x)}{(g(x))^2} \tag{5.30}$$

Sum rule: $$(f(x) + g(x))' = f'(x) + g'(x) \tag{5.31}$$

Chain rule: $$\big(g(f(x))\big)' = (g \circ f)'(x) = g'(f(x))f'(x) \tag{5.32}$$

Here, $g \circ f$ denotes function composition $x \mapsto f(x) \mapsto g(f(x))$.

Example 5.5 (Chain Rule)
Let us compute the derivative of the function $h(x) = (2x + 1)^4$ using the chain rule. With

$$h(x) = (2x + 1)^4 = g(f(x)), \tag{5.33}$$

$$f(x) = 2x + 1, \tag{5.34}$$

$$g(f) = f^4, \tag{5.35}$$

we obtain the derivatives of f and g as

$$f'(x) = 2, \tag{5.36}$$

$$g'(f) = 4f^3, \tag{5.37}$$

such that the derivative of h is given as

$$h'(x) = g'(f)f'(x) = (4f^3) \cdot 2 \overset{(5.34)}{=} 4(2x+1)^3 \cdot 2 = 8(2x+1)^3 \,, \quad (5.38)$$

where we used the chain rule (5.32) and substituted the definition of f in (5.34) in $g'(f)$.

5.2 Partial Differentiation and Gradients

Differentiation as discussed in Section 5.1 applies to functions f of a scalar variable $x \in \mathbb{R}$. In the following, we consider the general case where the function f depends on one or more variables $\boldsymbol{x} \in \mathbb{R}^n$, e.g., $f(\boldsymbol{x}) = f(x_1, x_2)$. The generalization of the derivative to functions of several variables is the *gradient*.

We find the gradient of the function f with respect to $\boldsymbol{x}$ by *varying one variable at a time* and keeping the others constant. The gradient is then the collection of these *partial derivatives*.

partial derivative

Definition 5.5 (Partial Derivative)**.** For a function $f : \mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{x} \mapsto f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^n$ of n variables $x_1, \ldots, x_n$ we define the *partial derivatives* as

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= \lim_{h \to 0} \frac{f(x_1 + h, x_2, \ldots, x_n) - f(\boldsymbol{x})}{h} \\ &\vdots \\ \frac{\partial f}{\partial x_n} &= \lim_{h \to 0} \frac{f(x_1, \ldots, x_{n-1}, x_n + h) - f(\boldsymbol{x})}{h} \end{aligned} \quad (5.39)$$

and collect them in the row vector

$$\nabla_{\boldsymbol{x}} f = \mathrm{grad} f = \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} & \frac{\partial f(\boldsymbol{x})}{\partial x_2} & \cdots & \frac{\partial f(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{1 \times n} \,, \quad (5.40)$$

where n is the number of variables and 1 is the dimension of the image/range/codomain of f. Here, we defined the column vector $\boldsymbol{x} = [x_1, \ldots, x_n]^\top \in \mathbb{R}^n$. The row vector in (5.40) is called the *gradient* of f or the *Jacobian* and is the generalization of the derivative from Section 5.1.

gradient

Jacobian

Remark. This definition of the Jacobian is a special case of the general definition of the Jacobian for vector-valued functions as the collection of partial derivatives. We will get back to this in Section 5.3. ◊

We can use results from scalar differentiation: Each partial derivative is a derivative with respect to a scalar.

Example 5.6 (Partial Derivatives Using the Chain Rule)

For $f(x, y) = (x + 2y^3)^2$, we obtain the partial derivatives

$$\frac{\partial f(x,y)}{\partial x} = 2(x + 2y^3)\frac{\partial}{\partial x}(x + 2y^3) = 2(x + 2y^3) \,, \quad (5.41)$$

$$\frac{\partial f(x,y)}{\partial y} = 2(x+2y^3)\frac{\partial}{\partial y}(x+2y^3) = 12(x+2y^3)y^2\,. \tag{5.42}$$

where we used the chain rule (5.32) to compute the partial derivatives.

Remark (Gradient as a Row Vector). It is not uncommon in the literature to define the gradient vector as a column vector, following the convention that vectors are generally column vectors. The reason why we define the gradient vector as a row vector is twofold: First, we can consistently generalize the gradient to vector-valued functions $f : \mathbb{R}^n \to \mathbb{R}^m$ (then the gradient becomes a matrix). Second, we can immediately apply the multi-variate chain rule without paying attention to the dimension of the gradient. We will discuss both points in Section 5.3. ♢

Example 5.7 (Gradient)
For $f(x_1, x_2) = x_1^2 x_2 + x_1 x_2^3 \in \mathbb{R}$, the partial derivatives (i.e., the derivatives of f with respect to x_1 and x_2) are

$$\frac{\partial f(x_1,x_2)}{\partial x_1} = 2x_1x_2 + x_2^3 \tag{5.43}$$

$$\frac{\partial f(x_1,x_2)}{\partial x_2} = x_1^2 + 3x_1x_2^2 \tag{5.44}$$

and the gradient is then

$$\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix}\frac{\partial f(x_1,x_2)}{\partial x_1} & \frac{\partial f(x_1,x_2)}{\partial x_2}\end{bmatrix} = \begin{bmatrix}2x_1x_2 + x_2^3 & x_1^2 + 3x_1x_2^2\end{bmatrix} \in \mathbb{R}^{1\times 2}\,. \tag{5.45}$$

5.2.1 Basic Rules of Partial Differentiation

In the multivariate case, where $\boldsymbol{x} \in \mathbb{R}^n$, the basic differentiation rules that we know from school (e.g., sum rule, product rule, chain rule; see also Section 5.1.2) still apply. However, when we compute derivatives with respect to vectors $\boldsymbol{x} \in \mathbb{R}^n$ we need to pay attention: Our gradients now involve vectors and matrices, and matrix multiplication is not commutative (Section 2.2.1), i.e., the order matters.

Product rule: $(fg)' = f'g + fg'$,
Sum rule: $(f+g)' = f' + g'$,
Chain rule: $(g(f))' = g'(f)f'$

Here are the general product rule, sum rule, and chain rule:

$$\text{Product rule:} \quad \frac{\partial}{\partial \boldsymbol{x}}\big(f(\boldsymbol{x})g(\boldsymbol{x})\big) = \frac{\partial f}{\partial \boldsymbol{x}} g(\boldsymbol{x}) + f(\boldsymbol{x})\frac{\partial g}{\partial \boldsymbol{x}} \tag{5.46}$$

$$\text{Sum rule:} \quad \frac{\partial}{\partial \boldsymbol{x}}\big(f(\boldsymbol{x}) + g(\boldsymbol{x})\big) = \frac{\partial f}{\partial \boldsymbol{x}} + \frac{\partial g}{\partial \boldsymbol{x}} \tag{5.47}$$

$$\text{Chain rule:} \qquad \frac{\partial}{\partial \boldsymbol{x}}(g \circ f)(\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{x}}\big(g(f(\boldsymbol{x}))\big) = \frac{\partial g}{\partial f}\frac{\partial f}{\partial \boldsymbol{x}} \tag{5.48}$$

This is only an intuition, but not mathematically correct since the partial derivative is not a fraction.

Let us have a closer look at the chain rule. The chain rule (5.48) resembles to some degree the rules for matrix multiplication where we said that neighboring dimensions have to match for matrix multiplication to be defined; see Section 2.2.1. If we go from left to right, the chain rule exhibits similar properties: ∂f shows up in the "denominator" of the first factor and in the "numerator" of the second factor. If we multiply the factors together, multiplication is defined, i.e., the dimensions of ∂f match, and ∂f "cancels", such that $\partial g / \partial \boldsymbol{x}$ remains.

5.2.2 Chain Rule

Consider a function $f : \mathbb{R}^2 \to \mathbb{R}$ of two variables x_1, x_2. Furthermore, $x_1(t)$ and $x_2(t)$ are themselves functions of t. To compute the gradient of f with respect to t, we need to apply the chain rule (5.48) for multivariate functions as

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1(t)}{\partial t} \\ \frac{\partial x_2(t)}{\partial t} \end{bmatrix} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial t}\,, \tag{5.49}$$

where d denotes the gradient and ∂ partial derivatives.

Example 5.8

Consider $f(x_1, x_2) = x_1^2 + 2x_2$, where $x_1 = \sin t$ and $x_2 = \cos t$, then

$$\begin{aligned} \frac{\mathrm{d}f}{\mathrm{d}t} &= \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial t} && (5.50a) \\ &= 2\sin t \frac{\partial \sin t}{\partial t} + 2\frac{\partial \cos t}{\partial t} && (5.50b) \\ &= 2\sin t \cos t - 2\sin t = 2\sin t(\cos t - 1) && (5.50c) \end{aligned}$$

is the corresponding derivative of f with respect to t.

If $f(x_1, x_2)$ is a function of x_1 and x_2, where $x_1(s,t)$ and $x_2(s,t)$ are themselves functions of two variables s and t, the chain rule yields the partial derivatives

$$\frac{\partial f}{\partial s} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial s} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial s}\,, \tag{5.51}$$

$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial t}\,, \tag{5.52}$$

and the gradient is obtained by the matrix multiplication

$$\frac{\mathrm{d}f}{\mathrm{d}(s,t)} = \frac{\partial f}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial(s,t)} = \underbrace{\begin{bmatrix}\frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2}\end{bmatrix}}_{=\frac{\partial f}{\partial \boldsymbol{x}}} \underbrace{\begin{bmatrix}\frac{\partial x_1}{\partial s} & \frac{\partial x_1}{\partial t} \\ \frac{\partial x_2}{\partial s} & \frac{\partial x_2}{\partial t}\end{bmatrix}}_{=\frac{\partial \boldsymbol{x}}{\partial(s,t)}} . \tag{5.53}$$

This compact way of writing the chain rule as a matrix multiplication only makes sense if the gradient is defined as a row vector. Otherwise, we will need to start transposing gradients for the matrix dimensions to match. This may still be straightforward as long as the gradient is a vector or a matrix; however, when the gradient becomes a tensor (we will discuss this in the following), the transpose is no longer a triviality.

The chain rule can be written as a matrix multiplication.

Remark (Verifying the Correctness of a Gradient Implementation). The definition of the partial derivatives as the limit of the corresponding difference quotient (see (5.39)) can be exploited when numerically checking the correctness of gradients in computer programs: When we compute gradients and implement them, we can use finite differences to numerically test our computation and implementation: We choose the value h to be small (e.g., $h = 10^{-4}$) and compare the finite-difference approximation from (5.39) with our (analytic) implementation of the gradient. If the error is small, our gradient implementation is probably correct. "Small" could mean that $\sqrt{\frac{\sum_i (dh_i - df_i)^2}{\sum_i (dh_i + df_i)^2}} < 10^{-6}$, where dh_i is the finite-difference approximation and df_i is the analytic gradient of f with respect to the ith variable x_i. ♢

Gradient checking

5.3 Gradients of Vector-Valued Functions

Thus far, we discussed partial derivatives and gradients of functions $f : \mathbb{R}^n \to \mathbb{R}$ mapping to the real numbers. In the following, we will generalize the concept of the gradient to vector-valued functions (vector fields) $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$, where $n \geqslant 1$ and $m > 1$.

For a function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ and a vector $\boldsymbol{x} = [x_1, \ldots, x_n]^\top \in \mathbb{R}^n$, the corresponding vector of function values is given as

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_m(\boldsymbol{x}) \end{bmatrix} \in \mathbb{R}^m . \tag{5.54}$$

Writing the vector-valued function in this way allows us to view a vector-valued function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ as a vector of functions $[f_1, \ldots, f_m]^\top$, $f_i : \mathbb{R}^n \to \mathbb{R}$ that map onto $\mathbb{R}$. The differentiation rules for every f_i are exactly the ones we discussed in Section 5.2.

Therefore, the partial derivative of a vector-valued function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ with respect to $x_i \in \mathbb{R}$, $i = 1, \dots n$, is given as the vector

$$\frac{\partial \boldsymbol{f}}{\partial x_i} = \begin{bmatrix} \frac{\partial f_1}{\partial x_i} \\ \vdots \\ \frac{\partial f_m}{\partial x_i} \end{bmatrix} = \begin{bmatrix} \lim_{h\to 0} \frac{f_1(x_1,\dots,x_{i-1},x_i+h,x_{i+1},\dots x_n)-f_1(\boldsymbol{x})}{h} \\ \vdots \\ \lim_{h\to 0} \frac{f_m(x_1,\dots,x_{i-1},x_i+h,x_{i+1},\dots x_n)-f_m(\boldsymbol{x})}{h} \end{bmatrix} \in \mathbb{R}^m . \tag{5.55}$$

From (5.40), we know that the gradient of $\boldsymbol{f}$ with respect to a vector is the row vector of the partial derivatives. In (5.55), every partial derivative $\partial \boldsymbol{f}/\partial x_i$ is itself a column vector. Therefore, we obtain the gradient of $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ with respect to $\boldsymbol{x} \in \mathbb{R}^n$ by collecting these partial derivatives:

$$\frac{\mathrm{d}\boldsymbol{f}(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}} = \left[\boxed{\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_1}} \cdots \boxed{\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_n}} \right] \tag{5.56a}$$

$$= \left[\boxed{\begin{matrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f_m(\boldsymbol{x})}{\partial x_1} \end{matrix}} \begin{matrix} \cdots \\ \\ \cdots \end{matrix} \boxed{\begin{matrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \vdots \\ \frac{\partial f_m(\boldsymbol{x})}{\partial x_n} \end{matrix}} \right] \in \mathbb{R}^{m\times n} . \tag{5.56b}$$

Jacobian

The gradient of a function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ is a matrix of size $m \times n$.

Definition 5.6 (Jacobian)**.** The collection of all first-order partial derivatives of a vector-valued function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ is called the *Jacobian*. The Jacobian $\boldsymbol{J}$ is an $m \times n$ matrix, which we define and arrange as follows:

$$\boldsymbol{J} = \nabla_{\boldsymbol{x}} \boldsymbol{f} = \frac{\mathrm{d}\boldsymbol{f}(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \tag{5.57}$$

$$= \begin{bmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\boldsymbol{x})}{\partial x_n} \end{bmatrix} , \tag{5.58}$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} , \quad J(i,j) = \frac{\partial f_i}{\partial x_j} . \tag{5.59}$$

As a special case of (5.58), a function $f : \mathbb{R}^n \to \mathbb{R}^1$, which maps a vector $\boldsymbol{x} \in \mathbb{R}^n$ onto a scalar (e.g., $f(\boldsymbol{x}) = \sum_{i=1}^n x_i$), possesses a Jacobian that is a row vector (matrix of dimension $1 \times n$); see (5.40).

numerator layout

Remark. In this book, we use the *numerator layout* of the derivative, i.e., the derivative $\mathrm{d}\boldsymbol{f}/\mathrm{d}\boldsymbol{x}$ of $\boldsymbol{f} \in \mathbb{R}^m$ with respect to $\boldsymbol{x} \in \mathbb{R}^n$ is an $m \times n$ matrix, where the elements of $\boldsymbol{f}$ define the rows and the elements of $\boldsymbol{x}$ define the columns of the corresponding Jacobian; see (5.58). There

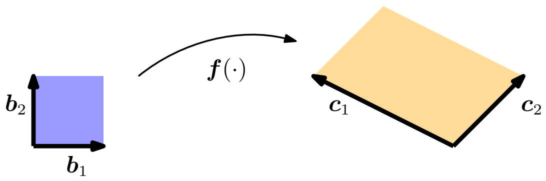


Figure 5.1 The determinant of the Jacobian of $\boldsymbol{f}$ can be used to compute the magnifier between the blue and orange area.

exists also the *denominator layout*, which is the transpose of the numerator layout. In this book, we will use the numerator layout. $\diamondsuit$

denominator layout

We will see how the Jacobian is used in the change-of-variable method for probability distributions in Section 6.7. The amount of scaling due to the transformation of a variable is provided by the determinant.

In Section 4.1, we saw that the determinant can be used to compute the area of a parallelogram. If we are given two vectors $\boldsymbol{b}_1 = [1, 0]^\top$, $\boldsymbol{b}_2 = [0, 1]^\top$ as the sides of the unit square (blue; see Figure 5.1), the area of this square is

$$\left|\det\left(\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)\right| = 1\,. \tag{5.60}$$

If we take a parallelogram with the sides $\boldsymbol{c}_1 = [-2, 1]^\top$, $\boldsymbol{c}_2 = [1, 1]^\top$ (orange in Figure 5.1), its area is given as the absolute value of the determinant (see Section 4.1)

$$\left|\det\left(\begin{bmatrix}-2 & 1\\1 & 1\end{bmatrix}\right)\right| = |-3| = 3\,, \tag{5.61}$$

i.e., the area of this is exactly three times the area of the unit square. We can find this scaling factor by finding a mapping that transforms the unit square into the other square. In linear algebra terms, we effectively perform a variable transformation from $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ to $(\boldsymbol{c}_1, \boldsymbol{c}_2)$. In our case, the mapping is linear and the absolute value of the determinant of this mapping gives us exactly the scaling factor we are looking for.

We will describe two approaches to identify this mapping. First, we exploit that the mapping is linear so that we can use the tools from Chapter 2 to identify this mapping. Second, we will find the mapping using partial derivatives using the tools we have been discussing in this chapter.

Approach 1 To get started with the linear algebra approach, we identify both $\{\boldsymbol{b}_1, \boldsymbol{b}_2\}$ and $\{\boldsymbol{c}_1, \boldsymbol{c}_2\}$ as bases of $\mathbb{R}^2$ (see Section 2.6.1 for a recap). What we effectively perform is a change of basis from $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ to $(\boldsymbol{c}_1, \boldsymbol{c}_2)$, and we are looking for the transformation matrix that implements the basis change. Using results from Section 2.7.2, we identify the desired basis change matrix as

$$\boldsymbol{J} = \begin{bmatrix}-2 & 1\\1 & 1\end{bmatrix}, \tag{5.62}$$

such that $\boldsymbol{J}\boldsymbol{b}_1 = \boldsymbol{c}_1$ and $\boldsymbol{J}\boldsymbol{b}_2 = \boldsymbol{c}_2$. The absolute value of the determinant of $\boldsymbol{J}$, which yields the scaling factor we are looking for, is given as $|\det(\boldsymbol{J})| = 3$, i.e., the area of the square spanned by $(\boldsymbol{c}_1, \boldsymbol{c}_2)$ is three times greater than the area spanned by $(\boldsymbol{b}_1, \boldsymbol{b}_2)$.

Approach 2 The linear algebra approach works for linear transformations; for nonlinear transformations (which become relevant in Section 6.7), we follow a more general approach using partial derivatives.

For this approach, we consider a function $\boldsymbol{f} : \mathbb{R}^2 \to \mathbb{R}^2$ that performs a variable transformation. In our example, $\boldsymbol{f}$ maps the coordinate representation of any vector $\boldsymbol{x} \in \mathbb{R}^2$ with respect to $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ onto the coordinate representation $\boldsymbol{y} \in \mathbb{R}^2$ with respect to $(\boldsymbol{c}_1, \boldsymbol{c}_2)$. We want to identify the mapping so that we can compute how an area (or volume) changes when it is being transformed by $\boldsymbol{f}$. For this, we need to find out how $\boldsymbol{f}(\boldsymbol{x})$ changes if we modify $\boldsymbol{x}$ a bit. This question is exactly answered by the Jacobian matrix $\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} \in \mathbb{R}^{2\times 2}$. Since we can write

$$y_1 = -2x_1 + x_2 \tag{5.63}$$

$$y_2 = x_1 + x_2 \tag{5.64}$$

we obtain the functional relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$, which allows us to get the partial derivatives

$$\frac{\partial y_1}{\partial x_1} = -2\,, \quad \frac{\partial y_1}{\partial x_2} = 1\,, \quad \frac{\partial y_2}{\partial x_1} = 1\,, \quad \frac{\partial y_2}{\partial x_2} = 1 \tag{5.65}$$

and compose the Jacobian as

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1 & 1 \end{bmatrix}. \tag{5.66}$$

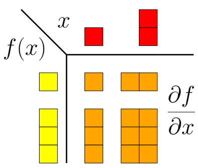
Geometrically, the Jacobian determinant gives the magnification/ scaling factor when we transform an area or volume.

Jacobian determinant

The Jacobian represents the coordinate transformation we are looking for. It is exact if the coordinate transformation is linear (as in our case), and (5.66) recovers exactly the basis change matrix in (5.62). If the coordinate transformation is nonlinear, the Jacobian approximates this nonlinear transformation locally with a linear one. The absolute value of the *Jacobian determinant* $|\det(\boldsymbol{J})|$ is the factor by which areas or volumes are scaled when coordinates are transformed. Our case yields $|\det(\boldsymbol{J})| = 3$.

The Jacobian determinant and variable transformations will become relevant in Section 6.7 when we transform random variables and probability distributions. These transformations are extremely relevant in machine learning in the context of training deep neural networks using the *reparametrization trick*, also called *infinite perturbation analysis*.

Figure 5.2 Dimensionality of (partial) derivatives.



In this chapter, we encountered derivatives of functions. Figure 5.2 summarizes the dimensions of those derivatives. If $f : \mathbb{R} \to \mathbb{R}$ the gradient is simply a scalar (top-left entry). For $f : \mathbb{R}^D \to \mathbb{R}$ the gradient is a $1 \times D$ row vector (top-right entry). For $\boldsymbol{f} : \mathbb{R} \to \mathbb{R}^E$, the gradient is an $E \times 1$ column vector, and for $\boldsymbol{f} : \mathbb{R}^D \to \mathbb{R}^E$ the gradient is an $E \times D$ matrix.

Example 5.9 (Gradient of a Vector-Valued Function)
We are given

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}\,, \qquad \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^M, \quad \boldsymbol{A} \in \mathbb{R}^{M \times N}, \quad \boldsymbol{x} \in \mathbb{R}^N\,.$$

To compute the gradient $\mathrm{d}\boldsymbol{f}/\mathrm{d}\boldsymbol{x}$ we first determine the dimension of $\mathrm{d}\boldsymbol{f}/\mathrm{d}\boldsymbol{x}$: Since $\boldsymbol{f} : \mathbb{R}^N \to \mathbb{R}^M$, it follows that $\mathrm{d}\boldsymbol{f}/\mathrm{d}\boldsymbol{x} \in \mathbb{R}^{M \times N}$. Second, to compute the gradient we determine the partial derivatives of f with respect to every x_j:

$$f_i(\boldsymbol{x}) = \sum_{j=1}^{N} A_{ij} x_j \implies \frac{\partial f_i}{\partial x_j} = A_{ij} \tag{5.67}$$

We collect the partial derivatives in the Jacobian and obtain the gradient

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \vdots & & \vdots \\ \frac{\partial f_M}{\partial x_1} & \cdots & \frac{\partial f_M}{\partial x_N} \end{bmatrix} = \begin{bmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & & \vdots \\ A_{M1} & \cdots & A_{MN} \end{bmatrix} = \boldsymbol{A} \in \mathbb{R}^{M \times N}\,. \tag{5.68}$$

Example 5.10 (Chain Rule)
Consider the function $h : \mathbb{R} \to \mathbb{R}$, $h(t) = (f \circ g)(t)$ with

$$f : \mathbb{R}^2 \to \mathbb{R} \tag{5.69}$$

$$g : \mathbb{R} \to \mathbb{R}^2 \tag{5.70}$$

$$f(\boldsymbol{x}) = \exp(x_1 x_2^2)\,, \tag{5.71}$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = g(t) = \begin{bmatrix} t\cos t \\ t\sin t \end{bmatrix} \tag{5.72}$$

and compute the gradient of h with respect to t. Since $f : \mathbb{R}^2 \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}^2$ we note that

$$\frac{\partial f}{\partial \boldsymbol{x}} \in \mathbb{R}^{1 \times 2}\,, \quad \frac{\partial g}{\partial t} \in \mathbb{R}^{2 \times 1}\,. \tag{5.73}$$

The desired gradient is computed by applying the chain rule:

$$\frac{\mathrm{d}h}{\mathrm{d}t} = \frac{\partial f}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial t} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1}{\partial t} \\ \frac{\partial x_2}{\partial t} \end{bmatrix} \tag{5.74a}$$

$$= \begin{bmatrix} \exp(x_1 x_2^2) x_2^2 & 2\exp(x_1 x_2^2) x_1 x_2 \end{bmatrix} \begin{bmatrix} \cos t - t\sin t \\ \sin t + t\cos t \end{bmatrix} \tag{5.74b}$$

$$= \exp(x_1 x_2^2)\big(x_2^2(\cos t - t\sin t) + 2x_1 x_2(\sin t + t\cos t)\big)\,, \tag{5.74c}$$

where $x_1 = t\cos t$ and $x_2 = t\sin t$; see (5.72).

We will discuss this model in much more detail in Chapter 9 in the context of linear regression, where we need derivatives of the least-squares loss L with respect to the parameters $\boldsymbol{\theta}$.

Example 5.11 (Gradient of a Least-Squares Loss in a Linear Model)

Let us consider the linear model

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\theta}\,, \tag{5.75}$$

where $\boldsymbol{\theta} \in \mathbb{R}^D$ is a parameter vector, $\boldsymbol{\Phi} \in \mathbb{R}^{N\times D}$ are input features and $\boldsymbol{y} \in \mathbb{R}^N$ are the corresponding observations. We define the functions

$$L(\boldsymbol{e}) := \|\boldsymbol{e}\|^2\,, \tag{5.76}$$

$$\boldsymbol{e}(\boldsymbol{\theta}) := \boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\,. \tag{5.77}$$

least-squares loss

We seek $\frac{\partial L}{\partial \boldsymbol{\theta}}$, and we will use the chain rule for this purpose. L is called a *least-squares loss* function.

Before we start our calculation, we determine the dimensionality of the gradient as

$$\frac{\partial L}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{1\times D}\,. \tag{5.78}$$

The chain rule allows us to compute the gradient as

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = \frac{\partial L}{\partial \boldsymbol{e}}\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{\theta}}\,, \tag{5.79}$$

```
dLdtheta =
np.einsum(
'n,nd',
dLde,dedtheta)
```

where the dth element is given by

$$\frac{\partial L}{\partial \boldsymbol{\theta}}[1,d] = \sum_{n=1}^{N} \frac{\partial L}{\partial \boldsymbol{e}}[n]\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{\theta}}[n,d]\,. \tag{5.80}$$

We know that $\|\boldsymbol{e}\|^2 = \boldsymbol{e}^\top \boldsymbol{e}$ (see Section 3.2) and determine

$$\frac{\partial L}{\partial \boldsymbol{e}} = 2\boldsymbol{e}^\top \in \mathbb{R}^{1\times N}\,. \tag{5.81}$$

Furthermore, we obtain

$$\frac{\partial \boldsymbol{e}}{\partial \boldsymbol{\theta}} = -\boldsymbol{\Phi} \in \mathbb{R}^{N\times D}\,, \tag{5.82}$$

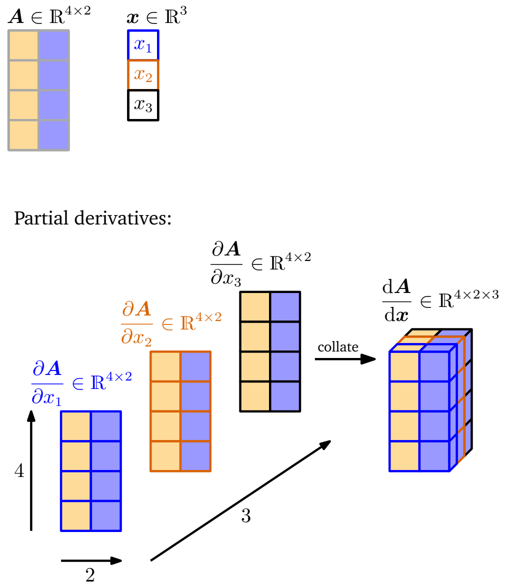
such that our desired derivative is

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = -2\boldsymbol{e}^\top\boldsymbol{\Phi} \overset{(5.77)}{=} -\underbrace{2(\boldsymbol{y}^\top - \boldsymbol{\theta}^\top\boldsymbol{\Phi}^\top)}_{1\times N}\underbrace{\boldsymbol{\Phi}}_{N\times D} \in \mathbb{R}^{1\times D}\,. \tag{5.83}$$

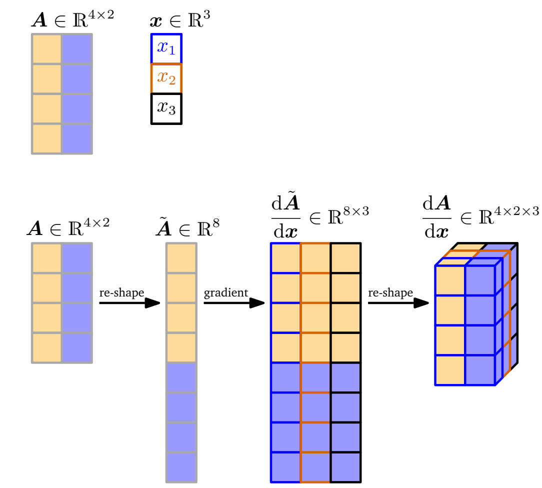
Remark. We would have obtained the same result without using the chain rule by immediately looking at the function

$$L_2(\boldsymbol{\theta}) := \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta}\|^2 = (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta})^\top(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\theta})\,. \tag{5.84}$$

This approach is still practical for simple functions like L_2 but becomes impractical for deep function compositions. ◊



(a) Approach 1: We compute the partial derivative $\frac{\partial \boldsymbol{A}}{\partial x_1}, \frac{\partial \boldsymbol{A}}{\partial x_2}, \frac{\partial \boldsymbol{A}}{\partial x_3}$, each of which is a 4×2 matrix, and collate them in a $4 \times 2 \times 3$ tensor.



(b) Approach 2: We re-shape (flatten) $\boldsymbol{A} \in \mathbb{R}^{4\times 2}$ into a vector $\tilde{\boldsymbol{A}} \in \mathbb{R}^8$. Then, we compute the gradient $\frac{d\tilde{\boldsymbol{A}}}{d\boldsymbol{x}} \in \mathbb{R}^{8\times 3}$. We obtain the gradient tensor by re-shaping this gradient as illustrated above.

Figure 5.3 Visualization of gradient computation of a matrix with respect to a vector. We are interested in computing the gradient of $\boldsymbol{A} \in \mathbb{R}^{4\times 2}$ with respect to a vector $\boldsymbol{x} \in \mathbb{R}^3$. We know that gradient $\frac{d\boldsymbol{A}}{d\boldsymbol{x}} \in \mathbb{R}^{4\times 2\times 3}$. We follow two equivalent approaches to arrive there: (a) collating partial derivatives into a Jacobian tensor; (b) flattening of the matrix into a vector, computing the Jacobian matrix, re-shaping into a Jacobian tensor.

5.4 Gradients of Matrices

We will encounter situations where we need to take gradients of matrices with respect to vectors (or other matrices), which results in a multidimensional tensor. We can think of this tensor as a multidimensional array that

We can think of a tensor as a multidimensional array.

collects partial derivatives. For example, if we compute the gradient of an $m \times n$ matrix $\boldsymbol{A}$ with respect to a $p \times q$ matrix $\boldsymbol{B}$, the resulting Jacobian would be $(m\times n)\times(p\times q)$, i.e., a four-dimensional tensor $\boldsymbol{J}$, whose entries are given as $J_{ijkl} = \partial A_{ij}/\partial B_{kl}$.

Since matrices represent linear mappings, we can exploit the fact that there is a vector-space isomorphism (linear, invertible mapping) between the space $\mathbb{R}^{m\times n}$ of $m \times n$ matrices and the space $\mathbb{R}^{mn}$ of mn vectors. Therefore, we can re-shape our matrices into vectors of lengths mn and pq, respectively. The gradient using these mn vectors results in a Jacobian of size $mn \times pq$. Figure 5.3 visualizes both approaches. In practical applications, it is often desirable to re-shape the matrix into a vector and continue working with this Jacobian matrix: The chain rule (5.48) boils down to simple matrix multiplication, whereas in the case of a Jacobian tensor, we will need to pay more attention to what dimensions we need to sum out.

Matrices can be transformed into vectors by stacking the columns of the matrix ("flattening").

Example 5.12 (Gradient of Vectors with Respect to Matrices)
Let us consider the following example, where

$$\boldsymbol{f} = \boldsymbol{A}\boldsymbol{x}\,, \quad \boldsymbol{f} \in \mathbb{R}^M, \quad \boldsymbol{A} \in \mathbb{R}^{M\times N}, \quad \boldsymbol{x} \in \mathbb{R}^N \tag{5.85}$$

and where we seek the gradient $\mathrm{d}\boldsymbol{f}/\mathrm{d}\boldsymbol{A}$. Let us start again by determining the dimension of the gradient as

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{A}} \in \mathbb{R}^{M\times(M\times N)}\,. \tag{5.86}$$

By definition, the gradient is the collection of the partial derivatives:

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{A}} = \begin{bmatrix} \frac{\partial f_1}{\partial \boldsymbol{A}} \\ \vdots \\ \frac{\partial f_M}{\partial \boldsymbol{A}} \end{bmatrix}, \quad \frac{\partial f_i}{\partial \boldsymbol{A}} \in \mathbb{R}^{1\times(M\times N)}\,. \tag{5.87}$$

To compute the partial derivatives, it will be helpful to explicitly write out the matrix vector multiplication:

$$f_i = \sum_{j=1}^{N} A_{ij}x_j, \quad i = 1, \ldots, M\,, \tag{5.88}$$

and the partial derivatives are then given as

$$\frac{\partial f_i}{\partial A_{iq}} = x_q\,. \tag{5.89}$$

This allows us to compute the partial derivatives of f_i with respect to a row of $\boldsymbol{A}$, which is given as

$$\frac{\partial f_i}{\partial A_{i,:}} = \boldsymbol{x}^\top \in \mathbb{R}^{1\times1\times N}\,, \tag{5.90}$$

$$\frac{\partial f_i}{\partial A_{k\neq i,:}} = \mathbf{0}^\top \in \mathbb{R}^{1\times 1\times N} \tag{5.91}$$

where we have to pay attention to the correct dimensionality. Since f_i maps onto $\mathbb{R}$ and each row of $\boldsymbol{A}$ is of size $1 \times N$, we obtain a $1 \times 1 \times N$-sized tensor as the partial derivative of f_i with respect to a row of $\boldsymbol{A}$.

We stack the partial derivatives (5.91) and get the desired gradient in (5.87) via

$$\frac{\partial f_i}{\partial \boldsymbol{A}} = \begin{bmatrix} \mathbf{0}^\top \\ \vdots \\ \mathbf{0}^\top \\ \boldsymbol{x}^\top \\ \mathbf{0}^\top \\ \vdots \\ \mathbf{0}^\top \end{bmatrix} \in \mathbb{R}^{1\times(M\times N)} . \tag{5.92}$$

Example 5.13 (Gradient of Matrices with Respect to Matrices)
Consider a matrix $\boldsymbol{R} \in \mathbb{R}^{M\times N}$ and $\boldsymbol{f} : \mathbb{R}^{M\times N} \to \mathbb{R}^{N\times N}$ with

$$\boldsymbol{f}(\boldsymbol{R}) = \boldsymbol{R}^\top \boldsymbol{R} =: \boldsymbol{K} \in \mathbb{R}^{N\times N} , \tag{5.93}$$

where we seek the gradient $\mathrm{d}\boldsymbol{K}/\mathrm{d}\boldsymbol{R}$.

To solve this hard problem, let us first write down what we already know: The gradient has the dimensions

$$\frac{\mathrm{d}\boldsymbol{K}}{\mathrm{d}\boldsymbol{R}} \in \mathbb{R}^{(N\times N)\times(M\times N)} , \tag{5.94}$$

which is a tensor. Moreover,

$$\frac{\mathrm{d}K_{pq}}{\mathrm{d}\boldsymbol{R}} \in \mathbb{R}^{1\times M\times N} \tag{5.95}$$

for $p, q = 1, \ldots, N$, where K_{pq} is the (p, q)th entry of $\boldsymbol{K} = \boldsymbol{f}(\boldsymbol{R})$. Denoting the ith column of $\boldsymbol{R}$ by $\boldsymbol{r}_i$, every entry of $\boldsymbol{K}$ is given by the dot product of two columns of $\boldsymbol{R}$, i.e.,

$$K_{pq} = \boldsymbol{r}_p^\top \boldsymbol{r}_q = \sum_{m=1}^{M} R_{mp} R_{mq} . \tag{5.96}$$

When we now compute the partial derivative $\frac{\partial K_{pq}}{\partial R_{ij}}$ we obtain

$$\frac{\partial K_{pq}}{\partial R_{ij}} = \sum_{m=1}^{M} \frac{\partial}{\partial R_{ij}} R_{mp} R_{mq} = \partial_{pqij} , \tag{5.97}$$

$$\partial_{pqij} = \begin{cases} R_{iq} & \text{if } j = p,\ p \neq q \\ R_{ip} & \text{if } j = q,\ p \neq q \\ 2R_{iq} & \text{if } j = p,\ p = q \\ 0 & \text{otherwise} \end{cases} . \tag{5.98}$$

From (5.94), we know that the desired gradient has the dimension $(N \times N) \times (M \times N)$, and every single entry of this tensor is given by ∂_{pqij} in (5.98), where $p, q, j = 1, \ldots, N$ and $i = 1, \ldots, M$.

5.5 Useful Identities for Computing Gradients

In the following, we list some useful gradients that are frequently required in a machine learning context (Petersen and Pedersen, 2012). Here, we use $\mathrm{tr}(\cdot)$ as the trace (see Definition 4.4), $\det(\cdot)$ as the determinant (see Section 4.1) and $\boldsymbol{f}(\boldsymbol{X})^{-1}$ as the inverse of $\boldsymbol{f}(\boldsymbol{X})$, assuming it exists.

$$\frac{\partial}{\partial \boldsymbol{X}} \boldsymbol{f}(\boldsymbol{X})^\top = \left(\frac{\partial \boldsymbol{f}(\boldsymbol{X})}{\partial \boldsymbol{X}}\right)^\top \tag{5.99}$$

$$\frac{\partial}{\partial \boldsymbol{X}} \mathrm{tr}(\boldsymbol{f}(\boldsymbol{X})) = \mathrm{tr}\left(\frac{\partial \boldsymbol{f}(\boldsymbol{X})}{\partial \boldsymbol{X}}\right) \tag{5.100}$$

$$\frac{\partial}{\partial \boldsymbol{X}} \det(\boldsymbol{f}(\boldsymbol{X})) = \det(\boldsymbol{f}(\boldsymbol{X}))\mathrm{tr}\left(\boldsymbol{f}(\boldsymbol{X})^{-1}\frac{\partial \boldsymbol{f}(\boldsymbol{X})}{\partial \boldsymbol{X}}\right) \tag{5.101}$$

$$\frac{\partial}{\partial \boldsymbol{X}} \boldsymbol{f}(\boldsymbol{X})^{-1} = -\boldsymbol{f}(\boldsymbol{X})^{-1}\frac{\partial \boldsymbol{f}(\boldsymbol{X})}{\partial \boldsymbol{X}}\boldsymbol{f}(\boldsymbol{X})^{-1} \tag{5.102}$$

$$\frac{\partial \boldsymbol{a}^\top \boldsymbol{X}^{-1} \boldsymbol{b}}{\partial \boldsymbol{X}} = -(\boldsymbol{X}^{-1})^\top \boldsymbol{a}\boldsymbol{b}^\top (\boldsymbol{X}^{-1})^\top \tag{5.103}$$

$$\frac{\partial \boldsymbol{x}^\top \boldsymbol{a}}{\partial \boldsymbol{x}} = \boldsymbol{a}^\top \tag{5.104}$$

$$\frac{\partial \boldsymbol{a}^\top \boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{a}^\top \tag{5.105}$$

$$\frac{\partial \boldsymbol{a}^\top \boldsymbol{X} \boldsymbol{b}}{\partial \boldsymbol{X}} = \boldsymbol{a}\boldsymbol{b}^\top \tag{5.106}$$

$$\frac{\partial \boldsymbol{x}^\top \boldsymbol{B} \boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{x}^\top(\boldsymbol{B} + \boldsymbol{B}^\top) \tag{5.107}$$

$$\frac{\partial}{\partial \boldsymbol{s}}(\boldsymbol{x} - \boldsymbol{A}\boldsymbol{s})^\top \boldsymbol{W}(\boldsymbol{x} - \boldsymbol{A}\boldsymbol{s}) = -2(\boldsymbol{x} - \boldsymbol{A}\boldsymbol{s})^\top \boldsymbol{W}\boldsymbol{A} \quad \text{for symmetric } \boldsymbol{W} \tag{5.108}$$

Remark. In this book, we only cover traces and transposes of matrices. However, we have seen that derivatives can be higher-dimensional tensors, in which case the usual trace and transpose are not defined. In these cases, the trace of a $D \times D \times E \times F$ tensor would be an $E \times F$-dimensional matrix. This is a special case of a tensor contraction. Similarly, when we "transpose" a tensor, we mean swapping the first two dimensions. Specifically, in (5.99) through (5.102), we require tensor-related computations when we work with multivariate functions $\boldsymbol{f}(\cdot)$ and compute derivatives with respect to matrices (and choose not to vectorize them as discussed in Section 5.4). $\diamondsuit$

5.6 Backpropagation and Automatic Differentiation

A good discussion about backpropagation and the chain rule is available at a blog by Tim Vieira at `https://tinyurl.com/ycfm2yrw`.

In many machine learning applications, we find good model parameters by performing gradient descent (Section 7.1), which relies on the fact that we can compute the gradient of a learning objective with respect to the parameters of the model. For a given objective function, we can obtain the gradient with respect to the model parameters using calculus and applying the chain rule; see Section 5.2.2. We already had a taste in Section 5.3 when we looked at the gradient of a squared loss with respect to the parameters of a linear regression model.

Consider the function

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos\left(x^2 + \exp(x^2)\right) . \tag{5.109}$$

By application of the chain rule, and noting that differentiation is linear, we compute the gradient

$$\begin{aligned} \frac{\mathrm{d}f}{\mathrm{d}x} &= \frac{2x + 2x\exp(x^2)}{2\sqrt{x^2 + \exp(x^2)}} - \sin\left(x^2 + \exp(x^2)\right)\left(2x + 2x\exp(x^2)\right) \\ &= 2x\left(\frac{1}{2\sqrt{x^2 + \exp(x^2)}} - \sin\left(x^2 + \exp(x^2)\right)\right)\left(1 + \exp(x^2)\right) . \end{aligned} \tag{5.110}$$

Writing out the gradient in this explicit way is often impractical since it often results in a very lengthy expression for a derivative. In practice, it means that, if we are not careful, the implementation of the gradient could be significantly more expensive than computing the function, which imposes unnecessary overhead. For training deep neural network models, the *backpropagation* algorithm (Kelley, 1960; Bryson, 1961; Dreyfus, 1962; Rumelhart et al., 1986) is an efficient way to compute the gradient of an error function with respect to the parameters of the model.

backpropagation

5.6.1 Gradients in a Deep Network

An area where the chain rule is used to an extreme is deep learning, where the function value $\boldsymbol{y}$ is computed as a many-level function composition

$$\boldsymbol{y} = (f_K \circ f_{K-1} \circ \cdots \circ f_1)(\boldsymbol{x}) = f_K(f_{K-1}(\cdots(f_1(\boldsymbol{x}))\cdots)) , \tag{5.111}$$

where $\boldsymbol{x}$ are the inputs (e.g., images), $\boldsymbol{y}$ are the observations (e.g., class labels), and every function f_i, $i = 1, \ldots, K$, possesses its own parameters.

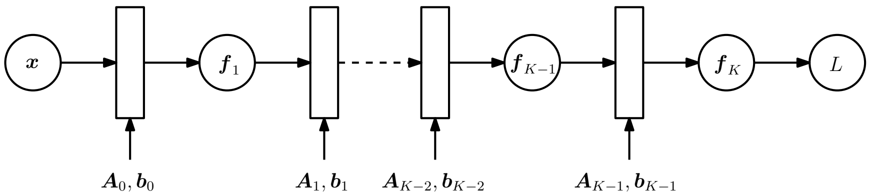


Figure 5.2 Forward pass in a multi-layer neural network to compute the loss L as a function of the inputs $\boldsymbol{x}$ and the parameters $\boldsymbol{A}_i$, $\boldsymbol{b}_i$.

We discuss the case, where the activation functions are identical in each layer to unclutter notation.

In neural networks with multiple layers, we have functions $f_i(\boldsymbol{x}_{i-1}) = \sigma(\boldsymbol{A}_{i-1}\boldsymbol{x}_{i-1} + \boldsymbol{b}_{i-1})$ in the ith layer. Here $\boldsymbol{x}_{i-1}$ is the output of layer $i-1$ and σ an activation function, such as the logistic sigmoid $\frac{1}{1+e^{-x}}$, tanh or a rectified linear unit (ReLU). In order to train these models, we require the gradient of a loss function L with respect to all model parameters $\boldsymbol{A}_j, \boldsymbol{b}_j$ for $j = 1, \ldots, K$. This also requires us to compute the gradient of L with respect to the inputs of each layer. For example, if we have inputs $\boldsymbol{x}$ and observations $\boldsymbol{y}$ and a network structure defined by

$$\boldsymbol{f}_0 := \boldsymbol{x} \tag{5.112}$$

$$\boldsymbol{f}_i := \sigma_i(\boldsymbol{A}_{i-1}\boldsymbol{f}_{i-1} + \boldsymbol{b}_{i-1}), \quad i = 1, \ldots, K, \tag{5.113}$$

see also Figure 5.2 for a visualization, we may be interested in finding $\boldsymbol{A}_j, \boldsymbol{b}_j$ for $j = 0, \ldots, K-1$, such that the squared loss

$$L(\boldsymbol{\theta}) = \|\boldsymbol{y} - \boldsymbol{f}_K(\boldsymbol{\theta}, \boldsymbol{x})\|^2 \tag{5.114}$$

is minimized, where $\boldsymbol{\theta} = \{\boldsymbol{A}_0, \boldsymbol{b}_0, \ldots, \boldsymbol{A}_{K-1}, \boldsymbol{b}_{K-1}\}$.

To obtain the gradients with respect to the parameter set $\boldsymbol{\theta}$, we require the partial derivatives of L with respect to the parameters $\boldsymbol{\theta}_j = \{\boldsymbol{A}_j, \boldsymbol{b}_j\}$ of each layer $j = 0, \ldots, K-1$. The chain rule allows us to determine the partial derivatives as

A more in-depth discussion about gradients of neural networks can be found in Justin Domke's lecture notes `https://tinyurl.com/yalcxgtv`.

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-1}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{\theta}_{K-1}} \tag{5.115}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-2}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \boxed{\frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \frac{\partial \boldsymbol{f}_{K-1}}{\partial \boldsymbol{\theta}_{K-2}}} \tag{5.116}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-3}} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \boxed{\frac{\partial \boldsymbol{f}_{K-1}}{\partial \boldsymbol{f}_{K-2}} \frac{\partial \boldsymbol{f}_{K-2}}{\partial \boldsymbol{\theta}_{K-3}}} \tag{5.117}$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_i} = \frac{\partial L}{\partial \boldsymbol{f}_K} \frac{\partial \boldsymbol{f}_K}{\partial \boldsymbol{f}_{K-1}} \cdots \boxed{\frac{\partial \boldsymbol{f}_{i+2}}{\partial \boldsymbol{f}_{i+1}} \frac{\partial \boldsymbol{f}_{i+1}}{\partial \boldsymbol{\theta}_i}} \tag{5.118}$$

The **orange** terms are partial derivatives of the output of a layer with respect to its inputs, whereas the **blue** terms are partial derivatives of the output of a layer with respect to its parameters. Assuming, we have already computed the partial derivatives $\partial L/\partial \boldsymbol{\theta}_{i+1}$, then most of the computation can be reused to compute $\partial L/\partial \boldsymbol{\theta}_i$. The additional terms that we

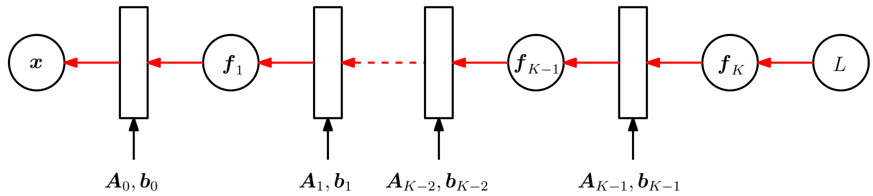


Figure 5.2 Backward pass in a multi-layer neural network to compute the gradients of the loss function.

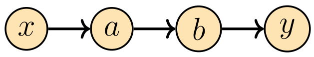


Figure 5.1 Simple graph illustrating the flow of data from x to y via some intermediate variables a, b.

need to compute are indicated by the boxes. Figure 5.2 visualizes that the gradients are passed backward through the network.

5.6.2 Automatic Differentiation

It turns out that backpropagation is a special case of a general technique in numerical analysis called *automatic differentiation*. We can think of automatic differentiation as a set of techniques to numerically (in contrast to symbolically) evaluate the exact (up to machine precision) gradient of a function by working with intermediate variables and applying the chain rule. Automatic differentiation applies a series of elementary arithmetic operations, e.g., addition and multiplication and elementary functions, e.g., $\sin, \cos, \exp, \log$. By applying the chain rule to these operations, the gradient of quite complicated functions can be computed automatically. Automatic differentiation applies to general computer programs and has forward and reverse modes. Baydin et al. (2018) give a great overview of automatic differentiation in machine learning.

automatic differentiation

Automatic differentiation is different from symbolic differentiation and numerical approximations of the gradient, e.g., by using finite differences.

Figure 5.1 shows a simple graph representing the data flow from inputs x to outputs y via some intermediate variables a, b. If we were to compute the derivative $\mathrm{d}y/\mathrm{d}x$, we would apply the chain rule and obtain

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\mathrm{d}y}{\mathrm{d}b}\frac{\mathrm{d}b}{\mathrm{d}a}\frac{\mathrm{d}a}{\mathrm{d}x}\,. \tag{5.119}$$

Intuitively, the forward and reverse mode differ in the order of multiplication. Due to the associativity of matrix multiplication, we can choose between

In the general case, we work with Jacobians, which can be vectors, matrices, or tensors.

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \left(\frac{\mathrm{d}y}{\mathrm{d}b}\frac{\mathrm{d}b}{\mathrm{d}a}\right)\frac{\mathrm{d}a}{\mathrm{d}x}\,, \tag{5.120}$$

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\mathrm{d}y}{\mathrm{d}b}\left(\frac{\mathrm{d}b}{\mathrm{d}a}\frac{\mathrm{d}a}{\mathrm{d}x}\right)\,. \tag{5.121}$$

Equation (5.120) would be the *reverse mode* because gradients are propagated backward through the graph, i.e., reverse to the data flow. Equation (5.121) would be the *forward mode*, where the gradients flow with the data from left to right through the graph.

reverse mode

forward mode

In the following, we will focus on reverse mode automatic differentiation, which is backpropagation. In the context of neural networks, where the input dimensionality is often much higher than the dimensionality of the labels, the reverse mode is computationally significantly cheaper than the forward mode. Let us start with an instructive example.

Example 5.14

Consider the function

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos\left(x^2 + \exp(x^2)\right) \tag{5.122}$$

from (5.109). If we were to implement a function f on a computer, we would be able to save some computation by using *intermediate variables*:

intermediate variables

$$a = x^2\,, \tag{5.123}$$
$$b = \exp(a)\,, \tag{5.124}$$
$$c = a + b\,, \tag{5.125}$$
$$d = \sqrt{c}\,, \tag{5.126}$$
$$e = \cos(c)\,, \tag{5.127}$$
$$f = d + e\,. \tag{5.128}$$

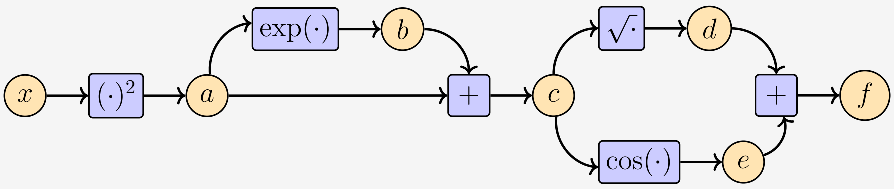


Figure 5.1 Computation graph with inputs x, function values f, and intermediate variables a, b, c, d, e.

This is the same kind of thinking process that occurs when applying the chain rule. Note that the preceding set of equations requires fewer operations than a direct implementation of the function $f(x)$ as defined in (5.109). The corresponding computation graph in Figure 5.1 shows the flow of data and computations required to obtain the function value f.

The set of equations that include intermediate variables can be thought of as a computation graph, a representation that is widely used in implementations of neural network software libraries. We can directly compute the derivatives of the intermediate variables with respect to their corresponding inputs by recalling the definition of the derivative of elementary functions. We obtain the following:

$$\frac{\partial a}{\partial x} = 2x \tag{5.129}$$
$$\frac{\partial b}{\partial a} = \exp(a) \tag{5.130}$$
$$\frac{\partial c}{\partial a} = 1 = \frac{\partial c}{\partial b} \tag{5.131}$$

$$\frac{\partial d}{\partial c} = \frac{1}{2\sqrt{c}} \tag{5.132}$$

$$\frac{\partial e}{\partial c} = -\sin(c) \tag{5.133}$$

$$\frac{\partial f}{\partial d} = 1 = \frac{\partial f}{\partial e}\,. \tag{5.134}$$

By looking at the computation graph in Figure 5.1, we can compute $\partial f/\partial x$ by working backward from the output and obtain

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial d}\frac{\partial d}{\partial c} + \frac{\partial f}{\partial e}\frac{\partial e}{\partial c} \tag{5.135}$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c}\frac{\partial c}{\partial b} \tag{5.136}$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b}\frac{\partial b}{\partial a} + \frac{\partial f}{\partial c}\frac{\partial c}{\partial a} \tag{5.137}$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a}\frac{\partial a}{\partial x}\,. \tag{5.138}$$

Note that we implicitly applied the chain rule to obtain $\partial f/\partial x$. By substituting the results of the derivatives of the elementary functions, we get

$$\frac{\partial f}{\partial c} = 1 \cdot \frac{1}{2\sqrt{c}} + 1 \cdot (-\sin(c)) \tag{5.139}$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} \cdot 1 \tag{5.140}$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b}\exp(a) + \frac{\partial f}{\partial c} \cdot 1 \tag{5.141}$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a} \cdot 2x\,. \tag{5.142}$$

By thinking of each of the derivatives above as a variable, we observe that the computation required for calculating the derivative is of similar complexity as the computation of the function itself. This is quite counterintuitive since the mathematical expression for the derivative $\frac{\partial f}{\partial x}$ (5.110) is significantly more complicated than the mathematical expression of the function $f(x)$ in (5.109).

Automatic differentiation is a formalization of Example 5.14. Let $x_1, \dots, x_d$ be the input variables to the function, $x_{d+1}, \dots, x_{D-1}$ be the intermediate variables, and x_D the output variable. Then the computation graph can be expressed as follows:

$$\text{For } i = d+1, \dots, D: \quad x_i = g_i(x_{\mathrm{Pa}(x_i)})\,, \tag{5.143}$$

where the $g_i(\cdot)$ are elementary functions and $x_{\mathrm{Pa}(x_i)}$ are the parent nodes of the variable x_i in the graph. Given a function defined in this way, we can use the chain rule to compute the derivative of the function in a step-by-step fashion. Recall that by definition $f = x_D$ and hence

$$\frac{\partial f}{\partial x_D} = 1\,. \tag{5.144}$$

For other variables x_i, we apply the chain rule

$$\frac{\partial f}{\partial x_i} = \sum_{x_j : x_i \in \mathrm{Pa}(x_j)} \frac{\partial f}{\partial x_j}\frac{\partial x_j}{\partial x_i} = \sum_{x_j : x_i \in \mathrm{Pa}(x_j)} \frac{\partial f}{\partial x_j}\frac{\partial g_j}{\partial x_i}\,, \tag{5.145}$$

where $\mathrm{Pa}(x_j)$ is the set of parent nodes of x_j in the computation graph. Equation (5.143) is the forward propagation of a function, whereas (5.145) is the backpropagation of the gradient through the computation graph. For neural network training, we backpropagate the error of the prediction with respect to the label.

Auto-differentiation in reverse mode requires a parse tree.

The automatic differentiation approach above works whenever we have a function that can be expressed as a computation graph, where the elementary functions are differentiable. In fact, the function may not even be a mathematical function but a computer program. However, not all computer programs can be automatically differentiated, e.g., if we cannot find differential elementary functions. Programming structures, such as `for` loops and `if` statements, require more care as well.

5.7 Higher-Order Derivatives

So far, we have discussed gradients, i.e., first-order derivatives. Sometimes, we are interested in derivatives of higher order, e.g., when we want to use Newton's Method for optimization, which requires second-order derivatives (Nocedal and Wright, 2006). In Section 5.1.1, we discussed the Taylor series to approximate functions using polynomials. In the multivariate case, we can do exactly the same. In the following, we will do exactly this. But let us start with some notation.

Consider a function $f : \mathbb{R}^2 \to \mathbb{R}$ of two variables x, y. We use the following notation for higher-order partial derivatives (and for gradients):

- $\frac{\partial^2 f}{\partial x^2}$ is the second partial derivative of f with respect to x.
- $\frac{\partial^n f}{\partial x^n}$ is the nth partial derivative of f with respect to x.
- $\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right)$ is the partial derivative obtained by first partial differentiating with respect to x and then with respect to y.
- $\frac{\partial^2 f}{\partial x \partial y}$ is the partial derivative obtained by first partial differentiating by y and then x.

Hessian

The *Hessian* is the collection of all second-order partial derivatives.

If $f(x, y)$ is a twice (continuously) differentiable function, then

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}\,, \tag{5.146}$$

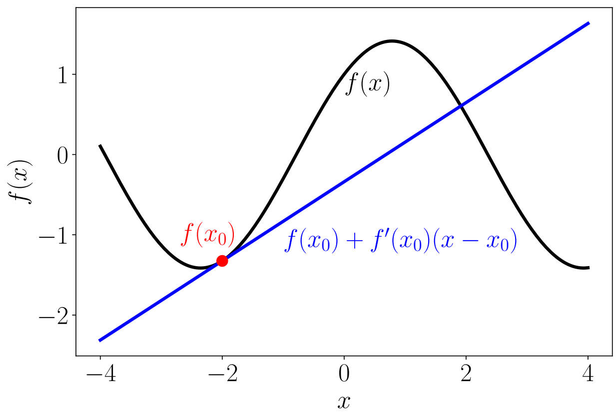


Figure 5.1 Linear approximation of a function. The original function f is linearized at $x_0 = -2$ using a first-order Taylor series expansion.

i.e., the order of differentiation does not matter, and the corresponding *Hessian matrix*

Hessian matrix

$$\boldsymbol{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \tag{5.147}$$

is symmetric. The Hessian is denoted as $\nabla^2_{x,y} f(x, y)$. Generally, for $\boldsymbol{x} \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$, the Hessian is an $n \times n$ matrix. The Hessian measures the curvature of the function locally around (x, y).

Remark (Hessian of a Vector Field). If $f : \mathbb{R}^n \to \mathbb{R}^m$ is a vector field, the Hessian is an $(m \times n \times n)$-tensor. ◊

5.8 Linearization and Multivariate Taylor Series

The gradient ∇f of a function f is often used for a locally linear approximation of f around $\boldsymbol{x}_0$:

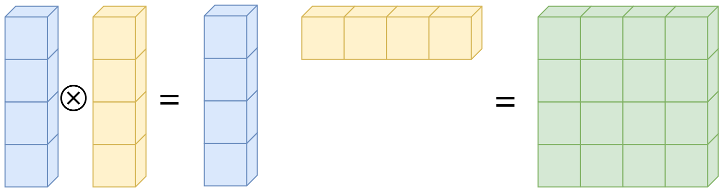
$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}_0) + (\nabla_{\boldsymbol{x}} f)(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0). \tag{5.148}$$

Here $(\nabla_{\boldsymbol{x}} f)(\boldsymbol{x}_0)$ is the gradient of f with respect to $\boldsymbol{x}$, evaluated at $\boldsymbol{x}_0$. Figure 5.1 illustrates the linear approximation of a function f at an input x_0. The original function is approximated by a straight line. This approximation is locally accurate, but the farther we move away from x_0 the worse the approximation gets. Equation (5.148) is a special case of a multivariate Taylor series expansion of f at $\boldsymbol{x}_0$, where we consider only the first two terms. We discuss the more general case in the following, which will allow for better approximations.

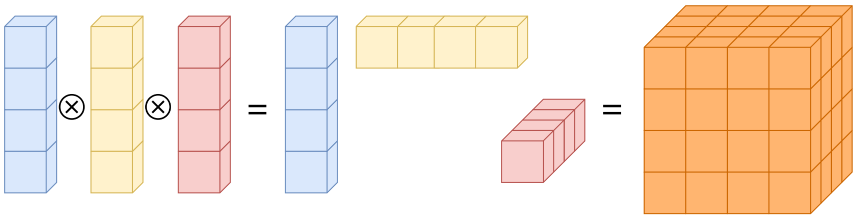
Definition 5.7 (Multivariate Taylor Series)**.** We consider a function

$$f : \mathbb{R}^D \to \mathbb{R} \tag{5.149}$$

Figure 5.1 Visualizing outer products. Outer products of vectors increase the dimensionality of the array by 1 per term. (a) The outer product of two vectors results in a matrix; (b) the outer product of three vectors yields a third-order tensor.



(a) Given a vector $\boldsymbol{\delta} \in \mathbb{R}^4$, we obtain the outer product $\boldsymbol{\delta}^2 := \boldsymbol{\delta} \otimes \boldsymbol{\delta} = \boldsymbol{\delta}\boldsymbol{\delta}^\top \in \mathbb{R}^{4\times 4}$ as a matrix.



(b) An outer product $\boldsymbol{\delta}^3 := \boldsymbol{\delta} \otimes \boldsymbol{\delta} \otimes \boldsymbol{\delta} \in \mathbb{R}^{4\times 4\times 4}$ results in a third-order tensor (“three-dimensional matrix”), i.e., an array with three indexes.

$$\boldsymbol{x} \mapsto f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^D, \tag{5.150}$$

that is smooth at $\boldsymbol{x}_0$. When we define the difference vector $\boldsymbol{\delta} := \boldsymbol{x} - \boldsymbol{x}_0$, the *multivariate Taylor series* of f at $(\boldsymbol{x}_0)$ is defined as

multivariate Taylor series

$$f(\boldsymbol{x}) = \sum_{k=0}^{\infty} \frac{D_{\boldsymbol{x}}^k f(\boldsymbol{x}_0)}{k!} \boldsymbol{\delta}^k, \tag{5.151}$$

where $D_{\boldsymbol{x}}^k f(\boldsymbol{x}_0)$ is the k-th (total) derivative of f with respect to $\boldsymbol{x}$, evaluated at $\boldsymbol{x}_0$.

Taylor polynomial

Definition 5.8 (Taylor Polynomial)**.** The *Taylor polynomial* of degree n of f at $\boldsymbol{x}_0$ contains the first $n+1$ components of the series in (5.151) and is defined as

$$T_n(\boldsymbol{x}) = \sum_{k=0}^{n} \frac{D_{\boldsymbol{x}}^k f(\boldsymbol{x}_0)}{k!} \boldsymbol{\delta}^k. \tag{5.152}$$

In (5.151) and (5.152), we used the slightly sloppy notation of $\boldsymbol{\delta}^k$, which is not defined for vectors $\boldsymbol{x} \in \mathbb{R}^D$, $D > 1$, and $k > 1$. Note that both $D_{\boldsymbol{x}}^k f$ and $\boldsymbol{\delta}^k$ are k-th order tensors, i.e., k-dimensional arrays. The kth-order tensor $\boldsymbol{\delta}^k \in \mathbb{R}^{\overbrace{D\times D\times \ldots \times D}^{k \text{ times}}}$ is obtained as a k-fold outer product, denoted by $\otimes$, of the vector $\boldsymbol{\delta} \in \mathbb{R}^D$. For example,

A vector can be implemented as a one-dimensional array, a matrix as a two-dimensional array.

$$\boldsymbol{\delta}^2 := \boldsymbol{\delta} \otimes \boldsymbol{\delta} = \boldsymbol{\delta}\boldsymbol{\delta}^\top, \quad \boldsymbol{\delta}^2[i,j] = \delta[i]\delta[j] \tag{5.153}$$

$$\boldsymbol{\delta}^3 := \boldsymbol{\delta} \otimes \boldsymbol{\delta} \otimes \boldsymbol{\delta}, \quad \boldsymbol{\delta}^3[i,j,k] = \delta[i]\delta[j]\delta[k]. \tag{5.154}$$

Figure 5.1 visualizes two such outer products. In general, we obtain the terms

$$D_{\boldsymbol{x}}^k f(\boldsymbol{x}_0)\boldsymbol{\delta}^k = \sum_{i_1=1}^{D} \cdots \sum_{i_k=1}^{D} D_{\boldsymbol{x}}^k f(\boldsymbol{x}_0)[i_1, \ldots, i_k]\delta[i_1] \cdots \delta[i_k] \tag{5.155}$$

in the Taylor series, where $D_{\boldsymbol{x}}^k f(\boldsymbol{x}_0)\boldsymbol{\delta}^k$ contains k-th order polynomials.

Now that we defined the Taylor series for vector fields, let us explicitly write down the first terms $D_{\boldsymbol{x}}^k f(\boldsymbol{x}_0)\boldsymbol{\delta}^k$ of the Taylor series expansion for $k = 0, \ldots, 3$ and $\boldsymbol{\delta} := \boldsymbol{x} - \boldsymbol{x}_0$:

```
np.einsum(
'i,i',Df1,d)
np.einsum(
'ij,i,j',
Df2,d,d)
np.einsum(
'ijk,i,j,k',
Df3,d,d,d)
```

$$k = 0 : D_{\boldsymbol{x}}^0 f(\boldsymbol{x}_0)\boldsymbol{\delta}^0 = f(\boldsymbol{x}_0) \in \mathbb{R} \tag{5.156}$$

$$k = 1 : D_{\boldsymbol{x}}^1 f(\boldsymbol{x}_0)\boldsymbol{\delta}^1 = \underbrace{\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_0)}_{1\times D} \underbrace{\boldsymbol{\delta}}_{D\times 1} = \sum_{i=1}^{D} \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_0)[i]\delta[i] \in \mathbb{R} \tag{5.157}$$

$$k = 2 : D_{\boldsymbol{x}}^2 f(\boldsymbol{x}_0)\boldsymbol{\delta}^2 = \operatorname{tr}\big(\underbrace{\boldsymbol{H}(\boldsymbol{x}_0)}_{D\times D} \underbrace{\boldsymbol{\delta}}_{D\times 1} \underbrace{\boldsymbol{\delta}^\top}_{1\times D}\big) = \boldsymbol{\delta}^\top \boldsymbol{H}(\boldsymbol{x}_0)\boldsymbol{\delta} \tag{5.158}$$

$$= \sum_{i=1}^{D} \sum_{j=1}^{D} H[i,j]\delta[i]\delta[j] \in \mathbb{R} \tag{5.159}$$

$$k = 3 : D_{\boldsymbol{x}}^3 f(\boldsymbol{x}_0)\boldsymbol{\delta}^3 = \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} D_x^3 f(\boldsymbol{x}_0)[i,j,k]\delta[i]\delta[j]\delta[k] \in \mathbb{R} \tag{5.160}$$

Here, $\boldsymbol{H}(\boldsymbol{x}_0)$ is the Hessian of f evaluated at $\boldsymbol{x}_0$.

Example 5.15 (Taylor Series Expansion of a Function with Two Variables)
Consider the function

$$f(x, y) = x^2 + 2xy + y^3 \,. \tag{5.161}$$

We want to compute the Taylor series expansion of f at $(x_0, y_0) = (1, 2)$. Before we start, let us discuss what to expect: The function in (5.161) is a polynomial of degree 3. We are looking for a Taylor series expansion, which itself is a linear combination of polynomials. Therefore, we do not expect the Taylor series expansion to contain terms of fourth or higher order to express a third-order polynomial. This means that it should be sufficient to determine the first four terms of (5.151) for an exact alternative representation of (5.161).

To determine the Taylor series expansion, we start with the constant term and the first-order derivatives, which are given by

$$f(1, 2) = 13 \tag{5.162}$$

$$\frac{\partial f}{\partial x} = 2x + 2y \implies \frac{\partial f}{\partial x}(1, 2) = 6 \tag{5.163}$$

$$\frac{\partial f}{\partial y} = 2x + 3y^2 \implies \frac{\partial f}{\partial y}(1,2) = 14\,. \tag{5.164}$$

Therefore, we obtain

$$D^1_{x,y} f(1,2) = \nabla_{x,y} f(1,2) = \begin{bmatrix} \frac{\partial f}{\partial x}(1,2) & \frac{\partial f}{\partial y}(1,2) \end{bmatrix} = \begin{bmatrix} 6 & 14 \end{bmatrix} \in \mathbb{R}^{1\times 2} \tag{5.165}$$

such that

$$\frac{D^1_{x,y} f(1,2)}{1!}\boldsymbol{\delta} = \begin{bmatrix} 6 & 14 \end{bmatrix} \begin{bmatrix} x-1 \\ y-2 \end{bmatrix} = 6(x-1) + 14(y-2)\,. \tag{5.166}$$

Note that $D^1_{x,y} f(1,2)\boldsymbol{\delta}$ contains only linear terms, i.e., first-order polynomials.

The second-order partial derivatives are given by

$$\frac{\partial^2 f}{\partial x^2} = 2 \implies \frac{\partial^2 f}{\partial x^2}(1,2) = 2 \tag{5.167}$$

$$\frac{\partial^2 f}{\partial y^2} = 6y \implies \frac{\partial^2 f}{\partial y^2}(1,2) = 12 \tag{5.168}$$

$$\frac{\partial^2 f}{\partial y \partial x} = 2 \implies \frac{\partial^2 f}{\partial y \partial x}(1,2) = 2 \tag{5.169}$$

$$\frac{\partial^2 f}{\partial x \partial y} = 2 \implies \frac{\partial^2 f}{\partial x \partial y}(1,2) = 2\,. \tag{5.170}$$

When we collect the second-order partial derivatives, we obtain the Hessian

$$\boldsymbol{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 6y \end{bmatrix}, \tag{5.171}$$

such that

$$\boldsymbol{H}(1,2) = \begin{bmatrix} 2 & 2 \\ 2 & 12 \end{bmatrix} \in \mathbb{R}^{2\times 2}\,. \tag{5.172}$$

Therefore, the next term of the Taylor-series expansion is given by

$$\frac{D^2_{x,y} f(1,2)}{2!}\boldsymbol{\delta}^2 = \frac{1}{2}\boldsymbol{\delta}^\top \boldsymbol{H}(1,2)\boldsymbol{\delta} \tag{5.173a}$$

$$= \frac{1}{2}\begin{bmatrix} x-1 & y-2 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 2 & 12 \end{bmatrix} \begin{bmatrix} x-1 \\ y-2 \end{bmatrix} \tag{5.173b}$$

$$= (x-1)^2 + 2(x-1)(y-2) + 6(y-2)^2\,. \tag{5.173c}$$

Here, $D^2_{x,y} f(1,2)\boldsymbol{\delta}^2$ contains only quadratic terms, i.e., second-order polynomials.

The third-order derivatives are obtained as

$$D^3_{x,y}f = \begin{bmatrix} \frac{\partial \boldsymbol{H}}{\partial x} & \frac{\partial \boldsymbol{H}}{\partial y} \end{bmatrix} \in \mathbb{R}^{2\times2\times2}, \tag{5.174}$$

$$D^3_{x,y}f[:,:,1] = \frac{\partial \boldsymbol{H}}{\partial x} = \begin{bmatrix} \frac{\partial^3 f}{\partial x^3} & \frac{\partial^3 f}{\partial x^2 \partial y} \\ \frac{\partial^3 f}{\partial x \partial y \partial x} & \frac{\partial^3 f}{\partial x \partial y^2} \end{bmatrix}, \tag{5.175}$$

$$D^3_{x,y}f[:,:,2] = \frac{\partial \boldsymbol{H}}{\partial y} = \begin{bmatrix} \frac{\partial^3 f}{\partial y \partial x^2} & \frac{\partial^3 f}{\partial y \partial x \partial y} \\ \frac{\partial^3 f}{\partial y^2 \partial x} & \frac{\partial^3 f}{\partial y^3} \end{bmatrix}. \tag{5.176}$$

Since most second-order partial derivatives in the Hessian in (5.171) are constant, the only nonzero third-order partial derivative is

$$\frac{\partial^3 f}{\partial y^3} = 6 \implies \frac{\partial^3 f}{\partial y^3}(1,2) = 6\,. \tag{5.177}$$

Higher-order derivatives and the mixed derivatives of degree 3 (e.g., $\frac{\partial f^3}{\partial x^2 \partial y}$) vanish, such that

$$D^3_{x,y}f[:,:,1] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad D^3_{x,y}f[:,:,2] = \begin{bmatrix} 0 & 0 \\ 0 & 6 \end{bmatrix} \tag{5.178}$$

and

$$\frac{D^3_{x,y}f(1,2)}{3!}\boldsymbol{\delta}^3 = (y-2)^3\,, \tag{5.179}$$

which collects all cubic terms of the Taylor series. Overall, the (exact) Taylor series expansion of f at $(x_0, y_0) = (1, 2)$ is

$$f(x) = f(1,2) + D^1_{x,y}f(1,2)\boldsymbol{\delta} + \frac{D^2_{x,y}f(1,2)}{2!}\boldsymbol{\delta}^2 + \frac{D^3_{x,y}f(1,2)}{3!}\boldsymbol{\delta}^3 \tag{5.180a}$$

$$\begin{aligned} &= f(1,2) + \frac{\partial f(1,2)}{\partial x}(x-1) + \frac{\partial f(1,2)}{\partial y}(y-2) \\ &\quad + \frac{1}{2!}\left(\frac{\partial^2 f(1,2)}{\partial x^2}(x-1)^2 + \frac{\partial^2 f(1,2)}{\partial y^2}(y-2)^2 \right. \\ &\quad \left. + 2\frac{\partial^2 f(1,2)}{\partial x \partial y}(x-1)(y-2)\right) + \frac{1}{6}\frac{\partial^3 f(1,2)}{\partial y^3}(y-2)^3 \end{aligned} \tag{5.180b}$$

$$\begin{aligned} &= 13 + 6(x-1) + 14(y-2) \\ &\quad + (x-1)^2 + 6(y-2)^2 + 2(x-1)(y-2) + (y-2)^3\,. \end{aligned} \tag{5.180c}$$

In this case, we obtained an exact Taylor series expansion of the polynomial in (5.161), i.e., the polynomial in (5.180c) is identical to the original polynomial in (5.161). In this particular example, this result is not surprising since the original function was a third-order polynomial, which we expressed through a linear combination of constant terms, first-order, second-order, and third-order polynomials in (5.180c).

5.9 Further Reading

Further details of matrix differentials, along with a short review of the required linear algebra, can be found in Magnus and Neudecker (2007). Automatic differentiation has had a long history, and we refer to Griewank and Walther (2003), Griewank and Walther (2008), and Elliott (2009) and the references therein.

In machine learning (and other disciplines), we often need to compute expectations, i.e., we need to solve integrals of the form

$$\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x})] = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}\,. \tag{5.181}$$

Even if $p(\boldsymbol{x})$ is in a convenient form (e.g., Gaussian), this integral generally cannot be solved analytically. The Taylor series expansion of f is one way of finding an approximate solution: Assuming $p(\boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{\mu},\, \boldsymbol{\Sigma}\big)$ is Gaussian, then the first-order Taylor series expansion around $\boldsymbol{\mu}$ locally linearizes the nonlinear function f. For linear functions, we can compute the mean (and the covariance) exactly if $p(\boldsymbol{x})$ is Gaussian distributed (see Section 6.5). This property is heavily exploited by the *extended Kalman filter* (Maybeck, 1979) for online state estimation in nonlinear dynamical systems (also called "state-space models"). Other deterministic ways to approximate the integral in (5.181) are the *unscented transform* (Julier and Uhlmann, 1997), which does not require any gradients, or the *Laplace approximation* (MacKay, 2003; Bishop, 2006; Murphy, 2012), which uses a second-order Taylor series expansion (requiring the Hessian) for a local Gaussian approximation of $p(\boldsymbol{x})$ around its mode.

extended Kalman filter

unscented transform

Laplace approximation

Exercises

5.1 Compute the derivative $f'(x)$ for

$$f(x) = \log(x^4)\sin(x^3)\,.$$

5.2 Compute the derivative $f'(x)$ of the logistic sigmoid

$$f(x) = \frac{1}{1+\exp(-x)}\,.$$

5.3 Compute the derivative $f'(x)$ of the function

$$f(x) = \exp(-\tfrac{1}{2\sigma^2}(x-\mu)^2)\,,$$

where $\mu,\ \sigma \in \mathbb{R}$ are constants.

5.4 Compute the Taylor polynomials T_n, $n = 0, \ldots, 5$ of $f(x) = \sin(x) + \cos(x)$ at $x_0 = 0$.

5.5 Consider the following functions:

$$\begin{aligned} f_1(\boldsymbol{x}) &= \sin(x_1)\cos(x_2)\,, \quad \boldsymbol{x} \in \mathbb{R}^2 \\ f_2(\boldsymbol{x}, \boldsymbol{y}) &= \boldsymbol{x}^\top \boldsymbol{y}\,, \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n \\ f_3(\boldsymbol{x}) &= \boldsymbol{x}\boldsymbol{x}^\top\,, \qquad \boldsymbol{x} \in \mathbb{R}^n \end{aligned}$$

a. What are the dimensions of $\frac{\partial f_i}{\partial \boldsymbol{x}}$?
b. Compute the Jacobians.

5.6 Differentiate f with respect to $\boldsymbol{t}$ and g with respect to $\boldsymbol{X}$, where

$$f(\boldsymbol{t}) = \sin(\log(\boldsymbol{t}^\top \boldsymbol{t}))\,, \qquad t \in \mathbb{R}^D$$
$$g(\boldsymbol{X}) = \mathrm{tr}(\boldsymbol{AXB})\,, \qquad \boldsymbol{A} \in \mathbb{R}^{D\times E}, \boldsymbol{X} \in \mathbb{R}^{E\times F}, \boldsymbol{B} \in \mathbb{R}^{F\times D}\,,$$

where tr$(\cdot)$ denotes the trace.

5.7 Compute the derivatives $\mathrm{d}f/\mathrm{d}\boldsymbol{x}$ of the following functions by using the chain rule. Provide the dimensions of every single partial derivative. Describe your steps in detail.

a.

$$f(z) = \log(1+z)\,, \quad z = \boldsymbol{x}^\top \boldsymbol{x}\,, \quad \boldsymbol{x} \in \mathbb{R}^D$$

b.

$$f(\boldsymbol{z}) = \sin(\boldsymbol{z})\,, \quad \boldsymbol{z} = \boldsymbol{Ax} + \boldsymbol{b}\,, \quad \boldsymbol{A} \in \mathbb{R}^{E\times D}, \boldsymbol{x} \in \mathbb{R}^D, \boldsymbol{b} \in \mathbb{R}^E$$

where $\sin(\cdot)$ is applied to every element of $\boldsymbol{z}$.

5.8 Compute the derivatives $\mathrm{d}f/\mathrm{d}\boldsymbol{x}$ of the following functions. Describe your steps in detail.

a. Use the chain rule. Provide the dimensions of every single partial derivative.

$$f(z) = \exp(-\tfrac{1}{2}z)$$
$$z = g(\boldsymbol{y}) = \boldsymbol{y}^\top \boldsymbol{S}^{-1} \boldsymbol{y}$$
$$\boldsymbol{y} = h(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{\mu}$$

where $\boldsymbol{x}, \boldsymbol{\mu} \in \mathbb{R}^D$, $\boldsymbol{S} \in \mathbb{R}^{D\times D}$.

b.

$$f(\boldsymbol{x}) = \mathrm{tr}(\boldsymbol{x}\boldsymbol{x}^\top + \sigma^2 \boldsymbol{I})\,, \quad \boldsymbol{x} \in \mathbb{R}^D$$

Here tr$(\boldsymbol{A})$ is the trace of $\boldsymbol{A}$, i.e., the sum of the diagonal elements A_{ii}. *Hint: Explicitly write out the outer product.*

c. Use the chain rule. Provide the dimensions of every single partial derivative. You do not need to compute the product of the partial derivatives explicitly.

$$\boldsymbol{f} = \tanh(\boldsymbol{z}) \in \mathbb{R}^M$$
$$\boldsymbol{z} = \boldsymbol{Ax} + \boldsymbol{b}, \quad \boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{A} \in \mathbb{R}^{M\times N}, \boldsymbol{b} \in \mathbb{R}^M.$$

Here, $\tanh$ is applied to every component of $\boldsymbol{z}$.

5.9 We define

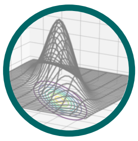
$$g(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\nu}) := \log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}, \boldsymbol{\nu})$$
$$\boldsymbol{z} := t(\boldsymbol{\epsilon}, \boldsymbol{\nu})$$

for differentiable functions p, q, t and $\boldsymbol{x} \in \mathbb{R}^D, \boldsymbol{z} \in \mathbb{R}^E, \boldsymbol{\nu} \in \mathbb{R}^F, \boldsymbol{\epsilon} \in \mathbb{R}^G$. By using the chain rule, compute the gradient

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\nu}} g(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\nu})\,.$$

6

Probability and Distributions



Probability, loosely speaking, concerns the study of uncertainty. Probability can be thought of as the fraction of times an event occurs, or as a degree of belief about an event. We then would like to use this probability to measure the chance of something occurring in an experiment. As mentioned in Chapter 1, we often quantify uncertainty in the data, uncertainty in the machine learning model, and uncertainty in the predictions produced by the model. Quantifying uncertainty requires the idea of a *random variable*, which is a function that maps outcomes of random experiments to a set of properties that we are interested in. Associated with the random variable is a function that measures the probability that a particular outcome (or set of outcomes) will occur; this is called the *probability distribution*.

random variable

probability distribution

Probability distributions are used as a building block for other concepts, such as probabilistic modeling (Section 8.4), graphical models (Section 8.5), and model selection (Section 8.6). In the next section, we present the three concepts that define a probability space (the sample space, the events, and the probability of an event) and how they are related to a fourth concept called the random variable. The presentation is deliberately slightly hand wavy since a rigorous presentation may occlude the intuition behind the concepts. An outline of the concepts presented in this chapter are shown in Figure 6.2.

6.1 Construction of a Probability Space

The theory of probability aims at defining a mathematical structure to describe random outcomes of experiments. For example, when tossing a single coin, we cannot determine the outcome, but by doing a large number of coin tosses, we can observe a regularity in the average outcome. Using this mathematical structure of probability, the goal is to perform automated reasoning, and in this sense, probability generalizes logical reasoning (Jaynes, 2003).

6.1.1 Philosophical Issues

When constructing automated reasoning systems, classical Boolean logic does not allow us to express certain forms of plausible reasoning. Consider

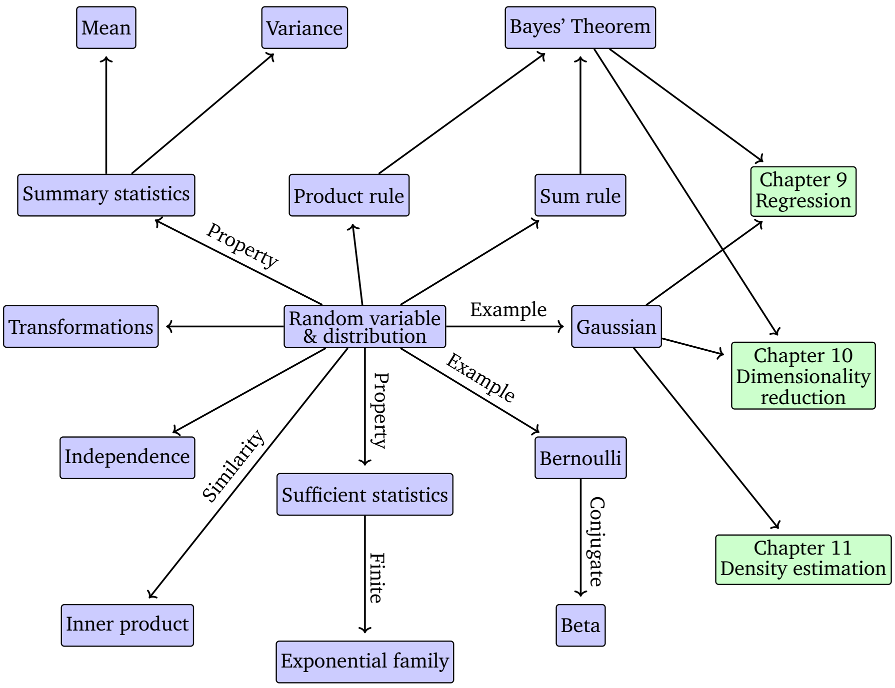


Figure 6.2 A mind map of the concepts related to random variables and probability distributions, as described in this chapter.

the following scenario: We observe that A is false. We find B becomes less plausible, although no conclusion can be drawn from classical logic. We observe that B is true. It seems A becomes more plausible. We use this form of reasoning daily. We are waiting for a friend, and consider three possibilities: H1, she is on time; H2, she has been delayed by traffic; and H3, she has been abducted by aliens. When we observe our friend is late, we must logically rule out H1. We also tend to consider H2 to be more likely, though we are not logically required to do so. Finally, we may consider H3 to be possible, but we continue to consider it quite unlikely. How do we conclude H2 is the most plausible answer? Seen in this way, probability theory can be considered a generalization of Boolean logic. In the context of machine learning, it is often applied in this way to formalize the design of automated reasoning systems. Further arguments about how probability theory is the foundation of reasoning systems can be found in Pearl (1988).

"For plausible reasoning it is necessary to extend the discrete true and false values of truth to continuous plausibilities" (Jaynes, 2003).

The philosophical basis of probability and how it should be somehow related to what we think should be true (in the logical sense) was studied by Cox (Jaynes, 2003). Another way to think about it is that if we are precise about our common sense we end up constructing probabilities. E. T. Jaynes (1922–1998) identified three mathematical criteria, which must apply to all plausibilities:

1. The degrees of plausibility are represented by real numbers.
2. These numbers must be based on the rules of common sense.

3. The resulting reasoning must be consistent, with the three following meanings of the word "consistent":
   (a) Consistency or non-contradiction: When the same result can be reached through different means, the same plausibility value must be found in all cases.
   (b) Honesty: All available data must be taken into account.
   (c) Reproducibility: If our state of knowledge about two problems are the same, then we must assign the same degree of plausibility to both of them.

The Cox–Jaynes theorem proves these plausibilities to be sufficient to define the universal mathematical rules that apply to plausibility p, up to transformation by an arbitrary monotonic function. Crucially, these rules *are* the rules of probability.

Remark. In machine learning and statistics, there are two major interpretations of probability: the Bayesian and frequentist interpretations (Bishop, 2006; Efron and Hastie, 2016). The Bayesian interpretation uses probability to specify the degree of uncertainty that the user has about an event. It is sometimes referred to as "subjective probability" or "degree of belief". The frequentist interpretation considers the relative frequencies of events of interest to the total number of events that occurred. The probability of an event is defined as the relative frequency of the event in the limit when one has infinite data. $\diamondsuit$

Some machine learning texts on probabilistic models use lazy notation and jargon, which is confusing. This text is no exception. Multiple distinct concepts are all referred to as "probability distribution", and the reader has to often disentangle the meaning from the context. One trick to help make sense of probability distributions is to check whether we are trying to model something categorical (a discrete random variable) or something continuous (a continuous random variable). The kinds of questions we tackle in machine learning are closely related to whether we are considering categorical or continuous models.

6.1.2 Probability and Random Variables

There are three distinct ideas that are often confused when discussing probabilities. First is the idea of a probability space, which allows us to quantify the idea of a probability. However, we mostly do not work directly with this basic probability space. Instead, we work with random variables (the second idea), which transfers the probability to a more convenient (often numerical) space. The third idea is the idea of a distribution or law associated with a random variable. We will introduce the first two ideas in this section and expand on the third idea in Section 6.2.

Modern probability is based on a set of axioms proposed by Kolmogorov

(Grinstead and Snell, 1997; Jaynes, 2003) that introduce the three concepts of sample space, event space, and probability measure. The probability space models a real-world process (referred to as an experiment) with random outcomes.

The sample space Ω

The *sample space* is the set of all possible outcomes of the experiment, usually denoted by Ω. For example, two successive coin tosses have a sample space of $\{$hh, tt, ht, th$\}$, where "h" denotes "heads" and "t" denotes "tails".

sample space

The event space $\mathcal{A}$

The *event space* is the space of potential results of the experiment. A subset A of the sample space Ω is in the event space $\mathcal{A}$ if at the end of the experiment we can observe whether a particular outcome $\omega \in \Omega$ is in A. The event space $\mathcal{A}$ is obtained by considering the collection of subsets of Ω, and for discrete probability distributions (Section 6.2.1) $\mathcal{A}$ is often the power set of Ω.

event space

The probability P

With each event $A \in \mathcal{A}$, we associate a number $P(A)$ that measures the probability or degree of belief that the event will occur. $P(A)$ is called the *probability* of A.

probability

The probability of a single event must lie in the interval $[0, 1]$, and the total probability over all outcomes in the sample space Ω must be 1, i.e., $P(\Omega) = 1$. Given a probability space $(\Omega, \mathcal{A}, P)$, we want to use it to model some real-world phenomenon. In machine learning, we often avoid explicitly referring to the probability space, but instead refer to probabilities on quantities of interest, which we denote by $\mathcal{T}$. In this book, we refer to $\mathcal{T}$ as the *target space* and refer to elements of $\mathcal{T}$ as states. We introduce a function $X : \Omega \to \mathcal{T}$ that takes an element of Ω (an outcome) and returns a particular quantity of interest x, a value in $\mathcal{T}$. This association/mapping from Ω to $\mathcal{T}$ is called a *random variable*. For example, in the case of tossing two coins and counting the number of heads, a random variable X maps to the three possible outcomes: $X(\mathrm{hh}) = 2$, $X(\mathrm{ht}) = 1$, $X(\mathrm{th}) = 1$, and $X(\mathrm{tt}) = 0$. In this particular case, $\mathcal{T} = \{0, 1, 2\}$, and it is the probabilities on elements of $\mathcal{T}$ that we are interested in. For a finite sample space Ω and finite $\mathcal{T}$, the function corresponding to a random variable is essentially a lookup table. For any subset $S \subseteq \mathcal{T}$, we associate $P_X(S) \in [0, 1]$ (the probability) to a particular event occurring corresponding to the random variable X. Example 6.1 provides a concrete illustration of the terminology.

target space

random variable

The name "random variable" is a great source of misunderstanding as it is neither random nor is it a variable. It is a function.

Remark. The aforementioned sample space Ω unfortunately is referred to by different names in different books. Another common name for Ω is "state space" (Jacod and Protter, 2004), but state space is sometimes reserved for referring to states in a dynamical system (Hasselblatt and

Katok, 2003). Other names sometimes used to describe Ω are: "sample description space", "possibility space," and "event space". $\diamondsuit$

This toy example is essentially a biased coin flip example.

Example 6.1

We assume that the reader is already familiar with computing probabilities of intersections and unions of sets of events. A gentler introduction to probability with many examples can be found in chapter 2 of Walpole et al. (2011).

Consider a statistical experiment where we model a funfair game consisting of drawing two coins from a bag (with replacement). There are coins from USA (denoted as \$) and UK (denoted as £) in the bag, and since we draw two coins from the bag, there are four outcomes in total. The state space or sample space Ω of this experiment is then (\$, \$), (\$, £), (£, \$), (£, £). Let us assume that the composition of the bag of coins is such that a draw returns at random a \$ with probability 0.3.

The event we are interested in is the total number of times the repeated draw returns \$. Let us define a random variable X that maps the sample space Ω to $\mathcal{T}$, which denotes the number of times we draw \$ out of the bag. We can see from the preceding sample space we can get zero \$, one \$, or two \$s, and therefore $\mathcal{T} = \{0, 1, 2\}$. The random variable X (a function or lookup table) can be represented as a table like the following:

$$X((\$, \$)) = 2 \tag{6.1}$$

$$X((\$, \pounds)) = 1 \tag{6.2}$$

$$X((\pounds, \$)) = 1 \tag{6.3}$$

$$X((\pounds, \pounds)) = 0\,. \tag{6.4}$$

Since we return the first coin we draw before drawing the second, this implies that the two draws are independent of each other, which we will discuss in Section 6.4.5. Note that there are two experimental outcomes, which map to the same event, where only one of the draws returns \$. Therefore, the probability mass function (Section 6.2.1) of X is given by

$$\begin{aligned} P(X=2) &= P((\$, \$)) \\ &= P(\$) \cdot P(\$) \\ &= 0.3 \cdot 0.3 = 0.09 \end{aligned} \tag{6.5}$$

$$\begin{aligned} P(X=1) &= P((\$, \pounds) \cup (\pounds, \$)) \\ &= P((\$, \pounds)) + P((\pounds, \$)) \\ &= 0.3 \cdot (1-0.3) + (1-0.3) \cdot 0.3 = 0.42 \end{aligned} \tag{6.6}$$

$$\begin{aligned} P(X=0) &= P((\pounds, \pounds)) \\ &= P(\pounds) \cdot P(\pounds) \\ &= (1-0.3) \cdot (1-0.3) = 0.49\,. \end{aligned} \tag{6.7}$$

In the calculation, we equated two different concepts, the probability of the output of X and the probability of the samples in Ω. For example, in (6.7) we say $P(X = 0) = P((\pounds, \pounds))$. Consider the random variable $X : \Omega \to \mathcal{T}$ and a subset $S \subseteq \mathcal{T}$ (for example, a single element of $\mathcal{T}$, such as the outcome that one head is obtained when tossing two coins). Let $X^{-1}(S)$ be the pre-image of S by X, i.e., the set of elements of Ω that map to S under X; $\{\omega \in \Omega : X(\omega) \in S\}$. One way to understand the transformation of probability from events in Ω via the random variable X is to associate it with the probability of the pre-image of S (Jacod and Protter, 2004). For $S \subseteq \mathcal{T}$, we have the notation

$$P_X(S) = P(X \in S) = P(X^{-1}(S)) = P(\{\omega \in \Omega : X(\omega) \in S\}) . \quad (6.8)$$

The left-hand side of (6.8) is the probability of the set of possible outcomes (e.g., number of $\$ = 1$) that we are interested in. Via the random variable X, which maps states to outcomes, we see in the right-hand side of (6.8) that this is the probability of the set of states (in Ω) that have the property (e.g., $\$\pounds$, $\pounds\$$). We say that a random variable X is distributed according to a particular probability distribution P_X, which defines the probability mapping between the event and the probability of the outcome of the random variable. In other words, the function P_X or equivalently $P \circ X^{-1}$ is the *law* or *distribution* of random variable X.

law
distribution

Remark. The target space, that is, the range $\mathcal{T}$ of the random variable X, is used to indicate the kind of probability space, i.e., a $\mathcal{T}$ random variable. When $\mathcal{T}$ is finite or countably infinite, this is called a discrete random variable (Section 6.2.1). For continuous random variables (Section 6.2.2), we only consider $\mathcal{T} = \mathbb{R}$ or $\mathcal{T} = \mathbb{R}^D$. ♢

6.1.3 Statistics

Probability theory and statistics are often presented together, but they concern different aspects of uncertainty. One way of contrasting them is by the kinds of problems that are considered. Using probability, we can consider a model of some process, where the underlying uncertainty is captured by random variables, and we use the rules of probability to derive what happens. In statistics, we observe that something has happened and try to figure out the underlying process that explains the observations. In this sense, machine learning is close to statistics in its goals to construct a model that adequately represents the process that generated the data. We can use the rules of probability to obtain a "best-fitting" model for some data.

Another aspect of machine learning systems is that we are interested in generalization error (see Chapter 8). This means that we are actually interested in the performance of our system on instances that we will observe in future, which are not identical to the instances that we have seen so far. This analysis of future performance relies on probability and statistics, most of which is beyond what will be presented in this chapter. The interested reader is encouraged to look at the books by Boucheron et al. (2013) and Shalev-Shwartz and Ben-David (2014). We will see more about statistics in Chapter 8.

6.2 Discrete and Continuous Probabilities

Let us focus our attention on ways to describe the probability of an event as introduced in Section 6.1. Depending on whether the target space is discrete or continuous, the natural way to refer to distributions is different. When the target space $\mathcal{T}$ is discrete, we can specify the probability that a random variable X takes a particular value $x \in \mathcal{T}$, denoted as $P(X = x)$. The expression $P(X = x)$ for a discrete random variable X is known as the *probability mass function*. When the target space $\mathcal{T}$ is continuous, e.g., the real line $\mathbb{R}$, it is more natural to specify the probability that a random variable X is in an interval, denoted by $P(a \leqslant X \leqslant b)$ for $a < b$. By convention, we specify the probability that a random variable X is less than a particular value x, denoted by $P(X \leqslant x)$. The expression $P(X \leqslant x)$ for a continuous random variable X is known as the *cumulative distribution function*. We will discuss continuous random variables in Section 6.2.2. We will revisit the nomenclature and contrast discrete and continuous random variables in Section 6.2.3.

probability mass function

cumulative distribution function

Remark. We will use the phrase *univariate* distribution to refer to distributions of a single random variable (whose states are denoted by non-bold x). We will refer to distributions of more than one random variable as *multivariate* distributions, and will usually consider a vector of random variables (whose states are denoted by bold $\boldsymbol{x}$). $\diamondsuit$

univariate

multivariate

6.2.1 Discrete Probabilities

When the target space is discrete, we can imagine the probability distribution of multiple random variables as filling out a (multidimensional) array of numbers. Figure 6.1 shows an example. The target space of the joint probability is the Cartesian product of the target spaces of each of the random variables. We define the *joint probability* as the entry of both values jointly

joint probability

$$P(X = x_i, Y = y_j) = \frac{n_{ij}}{N}\,, \tag{6.9}$$

where n_{ij} is the number of events with state x_i and y_j and N the total number of events. The joint probability is the probability of the intersection of both events, that is, $P(X = x_i, Y = y_j) = P(X = x_i \cap Y = y_j)$. Figure 6.1 illustrates the *probability mass function* (pmf) of a discrete probability distribution. For two random variables X and Y, the probability

probability mass function

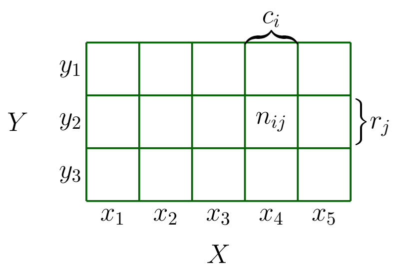


Figure 6.1 Visualization of a discrete bivariate probability mass function, with random variables X and Y. This diagram is adapted from Bishop (2006).

that $X = x$ and $Y = y$ is (lazily) written as $p(x, y)$ and is called the joint probability. One can think of a probability as a function that takes state x and y and returns a real number, which is the reason we write $p(x, y)$. The *marginal probability* that X takes the value x irrespective of the value of random variable Y is (lazily) written as $p(x)$. We write $X \sim p(x)$ to denote that the random variable X is distributed according to $p(x)$. If we consider only the instances where $X = x$, then the fraction of instances (the *conditional probability*) for which $Y = y$ is written (lazily) as $p(y \mid x)$.

marginal probability

conditional probability

Example 6.2

Consider two random variables X and Y, where X has five possible states and Y has three possible states, as shown in Figure 6.1. We denote by n_{ij} the number of events with state $X = x_i$ and $Y = y_j$, and denote by N the total number of events. The value c_i is the sum of the individual frequencies for the ith column, that is, $c_i = \sum_{j=1}^{3} n_{ij}$. Similarly, the value r_j is the row sum, that is, $r_j = \sum_{i=1}^{5} n_{ij}$. Using these definitions, we can compactly express the distribution of X and Y.

The probability distribution of each random variable, the marginal probability, can be seen as the sum over a row or column

$$P(X = x_i) = \frac{c_i}{N} = \frac{\sum_{j=1}^{3} n_{ij}}{N} \tag{6.10}$$

and

$$P(Y = y_j) = \frac{r_j}{N} = \frac{\sum_{i=1}^{5} n_{ij}}{N}, \tag{6.11}$$

where c_i and r_j are the ith column and jth row of the probability table, respectively. By convention, for discrete random variables with a finite number of events, we assume that probabilties sum up to one, that is,

$$\sum_{i=1}^{5} P(X = x_i) = 1 \quad \text{and} \quad \sum_{j=1}^{3} P(Y = y_j) = 1\,. \tag{6.12}$$

The conditional probability is the fraction of a row or column in a particular cell. For example, the conditional probability of Y given X is

$$P(Y = y_j \,|\, X = x_i) = \frac{n_{ij}}{c_i}\,, \tag{6.13}$$

and the conditional probability of X given Y is

$$P(X = x_i \,|\, Y = y_j) = \frac{n_{ij}}{r_j}\,. \tag{6.14}$$

categorical variable

In machine learning, we use discrete probability distributions to model *categorical variables*, i.e., variables that take a finite set of unordered values. They could be categorical features, such as the degree taken at university when used for predicting the salary of a person, or categorical labels, such as letters of the alphabet when doing handwriting recognition. Discrete distributions are also often used to construct probabilistic models that combine a finite number of continuous distributions (Chapter 11).

6.2.2 Continuous Probabilities

We consider real-valued random variables in this section, i.e., we consider target spaces that are intervals of the real line $\mathbb{R}$. In this book, we pretend that we can perform operations on real random variables as if we have discrete probability spaces with finite states. However, this simplification is not precise for two situations: when we repeat something infinitely often, and when we want to draw a point from an interval. The first situation arises when we discuss generalization errors in machine learning (Chapter 8). The second situation arises when we want to discuss continuous distributions, such as the Gaussian (Section 6.5). For our purposes, the lack of precision allows for a briefer introduction to probability.

measure

Borel σ-algebra

Remark. In continuous spaces, there are two additional technicalities, which are counterintuitive. First, the set of all subsets (used to define the event space $\mathcal{A}$ in Section 6.1) is not well behaved enough. $\mathcal{A}$ needs to be restricted to behave well under set complements, set intersections, and set unions. Second, the size of a set (which in discrete spaces can be obtained by counting the elements) turns out to be tricky. The size of a set is called its *measure*. For example, the cardinality of discrete sets, the length of an interval in $\mathbb{R}$, and the volume of a region in $\mathbb{R}^d$ are all measures. Sets that behave well under set operations and additionally have a topology are called a *Borel σ-algebra*. Betancourt details a careful construction of probability spaces from set theory without being bogged down in technicalities; see `https://tinyurl.com/yb3t6mfd`. For a more precise construction, we refer to Billingsley (1995) and Jacod and Protter (2004).

In this book, we consider real-valued random variables with their corresponding Borel σ-algebra. We consider random variables with values in $\mathbb{R}^D$ to be a vector of real-valued random variables. $\diamondsuit$

Definition 6.1 (Probability Density Function)**.** A function $f : \mathbb{R}^D \to \mathbb{R}$ is called a *probability density function* (*pdf*) if

probability density function
pdf

1. $\forall \boldsymbol{x} \in \mathbb{R}^D : f(\boldsymbol{x}) \geqslant 0$
2. Its integral exists and

$$\int_{\mathbb{R}^D} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1 \,. \tag{6.15}$$

For probability mass functions (pmf) of discrete random variables, the integral in (6.15) is replaced with a sum (6.12).

Observe that the probability density function is any function f that is non-negative and integrates to one. We associate a random variable X with this function f by

$$P(a \leqslant X \leqslant b) = \int_a^b f(x) \mathrm{d}x \,, \tag{6.16}$$

where $a, b \in \mathbb{R}$ and $x \in \mathbb{R}$ are outcomes of the continuous random variable X. States $\boldsymbol{x} \in \mathbb{R}^D$ are defined analogously by considering a vector of $x \in \mathbb{R}$. This association (6.16) is called the *law* or *distribution* of the random variable X.

law

$P(X = x)$ is a set of measure zero.

Remark. In contrast to discrete random variables, the probability of a continuous random variable X taking a particular value $P(X = x)$ is zero. This is like trying to specify an interval in (6.16) where $a = b$. $\diamondsuit$

Definition 6.2 (Cumulative Distribution Function)**.** A *cumulative distribution function* (cdf) of a multivariate real-valued random variable X with states $\boldsymbol{x} \in \mathbb{R}^D$ is given by

cumulative distribution function

$$F_X(\boldsymbol{x}) = P(X_1 \leqslant x_1, \ldots, X_D \leqslant x_D) \,, \tag{6.17}$$

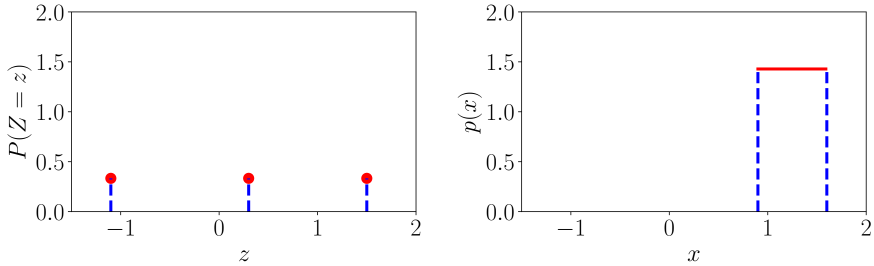
where $X = [X_1, \ldots, X_D]^\top$, $\boldsymbol{x} = [x_1, \ldots, x_D]^\top$, and the right-hand side represents the probability that random variable X_i takes the value smaller than or equal to x_i.

There are cdfs, which do not have corresponding pdfs.

The cdf can be expressed also as the integral of the probability density function $f(\boldsymbol{x})$ so that

$$F_X(\boldsymbol{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_D} f(z_1, \ldots, z_D) \mathrm{d}z_1 \cdots \mathrm{d}z_D \,. \tag{6.18}$$

Remark. We reiterate that there are in fact two distinct concepts when talking about distributions. First is the idea of a pdf (denoted by $f(x)$), which is a nonnegative function that sums to one. Second is the law of a random variable X, that is, the association of a random variable X with the pdf $f(x)$. $\diamondsuit$



(a) Discrete distribution (b) Continuous distribution

Figure 6.2 Examples of (a) discrete and (b) continuous uniform distributions. See Example 6.3 for details of the distributions.

For most of this book, we will not use the notation $f(x)$ and $F_X(x)$ as we mostly do not need to distinguish between the pdf and cdf. However, we will need to be careful about pdfs and cdfs in Section 6.7.

6.2.3 *Contrasting Discrete and Continuous Distributions*

Recall from Section 6.1.2 that probabilities are positive and the total probability sums up to one. For discrete random variables (see (6.12)), this implies that the probability of each state must lie in the interval $[0, 1]$. However, for continuous random variables the normalization (see (6.15)) does not imply that the value of the density is less than or equal to 1 for all values. We illustrate this in Figure 6.2 using the *uniform distribution* for both discrete and continuous random variables.

uniform distribution

Example 6.3

We consider two examples of the uniform distribution, where each state is equally likely to occur. This example illustrates some differences between discrete and continuous probability distributions.

Let Z be a discrete uniform random variable with three states $\{z = -1.1, z = 0.3, z = 1.5\}$. The probability mass function can be represented as a table of probability values:

The actual values of these states are not meaningful here, and we deliberately chose numbers to drive home the point that we do not want to use (and should ignore) the ordering of the states.

z	-1.1	0.3	1.5
$P(Z=z)$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

Alternatively, we can think of this as a graph (Figure 6.2(a)), where we use the fact that the states can be located on the x-axis, and the y-axis represents the probability of a particular state. The y-axis in Figure 6.2(a) is deliberately extended so that is it the same as in Figure 6.2(b).

Let X be a continuous random variable taking values in the range $0.9 \leqslant X \leqslant 1.6$, as represented by Figure 6.2(b). Observe that the height of the

Type	"Point probability"	"Interval probability"
Discrete	$P(X = x)$ Probability mass function	Not applicable
Continuous	$p(x)$ Probability density function	$P(X \leqslant x)$ Cumulative distribution function

Table 6.1 Nomenclature for probability distributions.

density can be greater than 1. However, it needs to hold that

$$\int_{0.9}^{1.6} p(x)\mathrm{d}x = 1\,. \tag{6.19}$$

Remark. There is an additional subtlety with regards to discrete probability distributions. The states $z_1, \ldots, z_d$ do not in principle have any structure, i.e., there is usually no way to compare them, for example $z_1 = \text{red}, z_2 = \text{green}, z_3 = \text{blue}$. However, in many machine learning applications discrete states take numerical values, e.g., $z_1 = -1.1, z_2 = 0.3, z_3 = 1.5$, where we could say $z_1 < z_2 < z_3$. Discrete states that assume numerical values are particularly useful because we often consider expected values (Section 6.4.1) of random variables. $\diamondsuit$

Unfortunately, machine learning literature uses notation and nomenclature that hides the distinction between the sample space Ω, the target space $\mathcal{T}$, and the random variable X. For a value x of the set of possible outcomes of the random variable X, i.e., $x \in \mathcal{T}$, $p(x)$ denotes the probability that random variable X has the outcome x. For discrete random variables, this is written as $P(X = x)$, which is known as the probability mass function. The pmf is often referred to as the "distribution". For continuous variables, $p(x)$ is called the probability density function (often referred to as a density). To muddy things even further, the cumulative distribution function $P(X \leqslant x)$ is often also referred to as the "distribution". In this chapter, we will use the notation X to refer to both univariate and multivariate random variables, and denote the states by x and $\boldsymbol{x}$ respectively. We summarize the nomenclature in Table 6.1.

We think of the outcome x as the argument that results in the probability $p(x)$.

Remark. We will be using the expression "probability distribution" not only for discrete probability mass functions but also for continuous probability density functions, although this is technically incorrect. In line with most machine learning literature, we also rely on context to distinguish the different uses of the phrase probability distribution. $\diamondsuit$

6.3 Sum Rule, Product Rule, and Bayes' Theorem

We think of probability theory as an extension to logical reasoning. As we discussed in Section 6.1.1, the rules of probability presented here follow naturally from fulfilling the desiderata (Jaynes, 2003, chapter 2). Probabilistic modeling (Section 8.4) provides a principled foundation for designing machine learning methods. Once we have defined probability distributions (Section 6.2) corresponding to the uncertainties of the data and our problem, it turns out that there are only two fundamental rules, the sum rule and the product rule.

Recall from (6.9) that $p(\boldsymbol{x}, \boldsymbol{y})$ is the joint distribution of the two random variables $\boldsymbol{x}, \boldsymbol{y}$. The distributions $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are the corresponding marginal distributions, and $p(\boldsymbol{y} \mid \boldsymbol{x})$ is the conditional distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$. Given the definitions of the marginal and conditional probability for discrete and continuous random variables in Section 6.2, we can now present the two fundamental rules in probability theory.

These two rules arise naturally (Jaynes, 2003) from the requirements we discussed in Section 6.1.1.

sum rule

The first rule, the *sum rule*, states that

$$p(\boldsymbol{x}) = \begin{cases} \displaystyle\sum_{\boldsymbol{y} \in \mathcal{Y}} p(\boldsymbol{x}, \boldsymbol{y}) & \text{if } \boldsymbol{y} \text{ is discrete} \\ \displaystyle\int_{\mathcal{Y}} p(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{y} & \text{if } \boldsymbol{y} \text{ is continuous} \end{cases}, \tag{6.20}$$

where $\mathcal{Y}$ are the states of the target space of random variable Y. This means that we sum out (or integrate out) the set of states $\boldsymbol{y}$ of the random variable Y. The sum rule is also known as the *marginalization property*. The sum rule relates the joint distribution to a marginal distribution. In general, when the joint distribution contains more than two random variables, the sum rule can be applied to any subset of the random variables, resulting in a marginal distribution of potentially more than one random variable. More concretely, if $\boldsymbol{x} = [x_1, \ldots, x_D]^\top$, we obtain the marginal

marginalization property

$$p(x_i) = \int p(x_1, \ldots, x_D) \mathrm{d}\boldsymbol{x}_{\setminus i} \tag{6.21}$$

by repeated application of the sum rule where we integrate/sum out all random variables except x_i, which is indicated by $\setminus i$, which reads "all except i."

Remark. Many of the computational challenges of probabilistic modeling are due to the application of the sum rule. When there are many variables or discrete variables with many states, the sum rule boils down to performing a high-dimensional sum or integral. Performing high-dimensional sums or integrals is generally computationally hard, in the sense that there is no known polynomial-time algorithm to calculate them exactly. ♢

product rule

The second rule, known as the *product rule*, relates the joint distribution to the conditional distribution via

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x}) \,. \tag{6.22}$$

The product rule can be interpreted as the fact that every joint distribution of two random variables can be factorized (written as a product) of two other distributions. The two factors are the marginal distribution of the first random variable $p(\boldsymbol{x})$, and the conditional distribution of the second random variable given the first $p(\boldsymbol{y} \mid \boldsymbol{x})$. Since the ordering of random variables is arbitrary in $p(\boldsymbol{x}, \boldsymbol{y})$, the product rule also implies $p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x} \mid \boldsymbol{y})p(\boldsymbol{y})$. To be precise, (6.22) is expressed in terms of the probability mass functions for discrete random variables. For continuous random variables, the product rule is expressed in terms of the probability density functions (Section 6.2.3).

In machine learning and Bayesian statistics, we are often interested in making inferences of unobserved (latent) random variables given that we have observed other random variables. Let us assume we have some prior knowledge $p(\boldsymbol{x})$ about an unobserved random variable $\boldsymbol{x}$ and some relationship $p(\boldsymbol{y} \mid \boldsymbol{x})$ between $\boldsymbol{x}$ and a second random variable $\boldsymbol{y}$, which we can observe. If we observe $\boldsymbol{y}$, we can use Bayes' theorem to draw some conclusions about $\boldsymbol{x}$ given the observed values of $\boldsymbol{y}$. *Bayes' theorem* (also *Bayes' rule* or *Bayes' law*)

Bayes' theorem
Bayes' rule
Bayes' law

$$\underbrace{p(\boldsymbol{x} \mid \boldsymbol{y})}_{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{y} \mid \boldsymbol{x})}^{\text{likelihood}}\overbrace{p(\boldsymbol{x})}^{\text{prior}}}{\underbrace{p(\boldsymbol{y})}_{\text{evidence}}} \tag{6.23}$$

is a direct consequence of the product rule in (6.22) since

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x} \mid \boldsymbol{y})p(\boldsymbol{y}) \tag{6.24}$$

and

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{x})p(\boldsymbol{x}) \tag{6.25}$$

so that

$$p(\boldsymbol{x} \mid \boldsymbol{y})p(\boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{x})p(\boldsymbol{x}) \iff p(\boldsymbol{x} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})} . \tag{6.26}$$

In (6.23), $p(\boldsymbol{x})$ is the *prior*, which encapsulates our subjective prior knowledge of the unobserved (latent) variable $\boldsymbol{x}$ before observing any data. We can choose any prior that makes sense to us, but it is critical to ensure that the prior has a nonzero pdf (or pmf) on all plausible $\boldsymbol{x}$, even if they are very rare.

prior

The *likelihood* $p(\boldsymbol{y} \mid \boldsymbol{x})$ describes how $\boldsymbol{x}$ and $\boldsymbol{y}$ are related, and in the case of discrete probability distributions, it is the probability of the data $\boldsymbol{y}$ if we were to know the latent variable $\boldsymbol{x}$. Note that the likelihood is not a distribution in $\boldsymbol{x}$, but only in $\boldsymbol{y}$. We call $p(\boldsymbol{y} \mid \boldsymbol{x})$ either the "likelihood of $\boldsymbol{x}$ (given $\boldsymbol{y}$)" or the "probability of $\boldsymbol{y}$ given $\boldsymbol{x}$" but never the likelihood of $\boldsymbol{y}$ (MacKay, 2003).

likelihood

The likelihood is sometimes also called the "measurement model".

The *posterior* $p(\boldsymbol{x} \mid \boldsymbol{y})$ is the quantity of interest in Bayesian statistics because it expresses exactly what we are interested in, i.e., what we know about $\boldsymbol{x}$ after having observed $\boldsymbol{y}$.

posterior

The quantity

$$p(\boldsymbol{y}) := \int p(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \mathbb{E}_X[p(\boldsymbol{y} \mid \boldsymbol{x})] \tag{6.27}$$

marginal likelihood
evidence

is the *marginal likelihood/evidence*. The right-hand side of (6.27) uses the expectation operator which we define in Section 6.4.1. By definition, the marginal likelihood integrates the numerator of (6.23) with respect to the latent variable $\boldsymbol{x}$. Therefore, the marginal likelihood is independent of $\boldsymbol{x}$, and it ensures that the posterior $p(\boldsymbol{x} \mid \boldsymbol{y})$ is normalized. The marginal likelihood can also be interpreted as the expected likelihood where we take the expectation with respect to the prior $p(\boldsymbol{x})$. Beyond normalization of the posterior, the marginal likelihood also plays an important role in Bayesian model selection, as we will discuss in Section 8.6. Due to the integration in (8.44), the evidence is often hard to compute.

Bayes' theorem is also called the "probabilistic inverse."
probabilistic inverse

Bayes' theorem (6.23) allows us to invert the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ given by the likelihood. Therefore, Bayes' theorem is sometimes called the *probabilistic inverse*. We will discuss Bayes' theorem further in Section 8.4.

Remark. In Bayesian statistics, the posterior distribution is the quantity of interest as it encapsulates all available information from the prior and the data. Instead of carrying the posterior around, it is possible to focus on some statistic of the posterior, such as the maximum of the posterior, which we will discuss in Section 8.3. However, focusing on some statistic of the posterior leads to loss of information. If we think in a bigger context, then the posterior can be used within a decision-making system, and having the full posterior can be extremely useful and lead to decisions that are robust to disturbances. For example, in the context of model-based reinforcement learning, Deisenroth et al. (2015) show that using the full posterior distribution of plausible transition functions leads to very fast (data/sample efficient) learning, whereas focusing on the maximum of the posterior leads to consistent failures. Therefore, having the full posterior can be very useful for a downstream task. In Chapter 9, we will continue this discussion in the context of linear regression. $\diamondsuit$

6.4 Summary Statistics and Independence

We are often interested in summarizing sets of random variables and comparing pairs of random variables. A statistic of a random variable is a deterministic function of that random variable. The summary statistics of a distribution provide one useful view of how a random variable behaves, and as the name suggests, provide numbers that summarize and characterize the distribution. We describe the mean and the variance, two well-known summary statistics. Then we discuss two ways to compare a pair of random variables: first, how to say that two random variables are independent; and second, how to compute an inner product between them.

6.4.1 Means and Covariances

Mean and (co)variance are often useful to describe properties of probability distributions (expected values and spread). We will see in Section 6.6 that there is a useful family of distributions (called the exponential family), where the statistics of the random variable capture all possible information.

The concept of the expected value is central to machine learning, and the foundational concepts of probability itself can be derived from the expected value (Whittle, 2000).

Definition 6.3 (Expected Value)**.** The *expected value* of a function $g : \mathbb{R} \to \mathbb{R}$ of a univariate continuous random variable $X \sim p(x)$ is given by

expected value

$$\mathbb{E}_X[g(x)] = \int_{\mathcal{X}} g(x)p(x)\mathrm{d}x\,. \tag{6.28}$$

Correspondingly, the expected value of a function g of a discrete random variable $X \sim p(x)$ is given by

$$\mathbb{E}_X[g(x)] = \sum_{x \in \mathcal{X}} g(x)p(x)\,, \tag{6.29}$$

where $\mathcal{X}$ is the set of possible outcomes (the target space) of the random variable X.

In this section, we consider discrete random variables to have numerical outcomes. This can be seen by observing that the function g takes real numbers as inputs.

The expected value of a function of a random variable is sometimes referred to as the law of the unconscious statistician (Casella and Berger, 2002, Section 2.2).

Remark. We consider multivariate random variables X as a finite vector of univariate random variables $[X_1, \ldots, X_D]^\top$. For multivariate random variables, we define the expected value element wise

$$\mathbb{E}_X[g(\boldsymbol{x})] = \begin{bmatrix} \mathbb{E}_{X_1}[g(x_1)] \\ \vdots \\ \mathbb{E}_{X_D}[g(x_D)] \end{bmatrix} \in \mathbb{R}^D\,, \tag{6.30}$$

where the subscript $\mathbb{E}_{X_d}$ indicates that we are taking the expected value with respect to the dth element of the vector $\boldsymbol{x}$. ♢

Definition 6.3 defines the meaning of the notation $\mathbb{E}_X$ as the operator indicating that we should take the integral with respect to the probability density (for continuous distributions) or the sum over all states (for discrete distributions). The definition of the mean (Definition 6.4), is a special case of the expected value, obtained by choosing g to be the identity function.

Definition 6.4 (Mean)**.** The *mean* of a random variable X with states

mean

$\boldsymbol{x} \in \mathbb{R}^D$ is an average and is defined as

$$\mathbb{E}_X[\boldsymbol{x}] = \begin{bmatrix} \mathbb{E}_{X_1}[x_1] \\ \vdots \\ \mathbb{E}_{X_D}[x_D] \end{bmatrix} \in \mathbb{R}^D, \tag{6.31}$$

where

$$\mathbb{E}_{X_d}[x_d] := \begin{cases} \displaystyle\int_{\mathcal{X}} x_d p(x_d) \mathrm{d}x_d & \text{if } X \text{ is a continuous random variable} \\ \displaystyle\sum_{x_i \in \mathcal{X}} x_i p(x_d = x_i) & \text{if } X \text{ is a discrete random variable} \end{cases} \tag{6.32}$$

for $d = 1, \ldots, D$, where the subscript d indicates the corresponding dimension of $\boldsymbol{x}$. The integral and sum are over the states $\mathcal{X}$ of the target space of the random variable X.

median

mode

In one dimension, there are two other intuitive notions of "average", which are the *median* and the *mode*. The *median* is the "middle" value if we sort the values, i.e., 50% of the values are greater than the median and 50% are smaller than the median. This idea can be generalized to continuous values by considering the value where the cdf (Definition 6.2) is 0.5. For distributions, which are asymmetric or have long tails, the median provides an estimate of a typical value that is closer to human intuition than the mean value. Furthermore, the median is more robust to outliers than the mean. The generalization of the median to higher dimensions is non-trivial as there is no obvious way to "sort" in more than one dimension (Hallin et al., 2010; Kong and Mizera, 2012). The *mode* is the most frequently occurring value. For a discrete random variable, the mode is defined as the value of x having the highest frequency of occurrence. For a continuous random variable, the mode is defined as a peak in the density $p(\boldsymbol{x})$. A particular density $p(\boldsymbol{x})$ may have more than one mode, and furthermore there may be a very large number of modes in high-dimensional distributions. Therefore, finding all the modes of a distribution can be computationally challenging.

Example 6.4

Consider the two-dimensional distribution illustrated in Figure 6.2:

$$p(x) = 0.4\,\mathcal{N}\left(\boldsymbol{x} \,\middle|\, \begin{bmatrix} 10 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) + 0.6\,\mathcal{N}\left(\boldsymbol{x} \,\middle|\, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 8.4 & 2.0 \\ 2.0 & 1.7 \end{bmatrix}\right). \tag{6.33}$$

We will define the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ in Section 6.5. Also shown is its corresponding marginal distribution in each dimension. Observe that the distribution is bimodal (has two modes), but one of the marginal distributions is unimodal (has one mode). The horizontal bimodal univariate distribution illustrates that the mean and median can be different from each other. While it is tempting to define the two-dimensional median to be the concatenation of the medians in each dimension, the fact that we cannot define an ordering of two-dimensional points makes it difficult. When we say "cannot define an ordering", we mean that there is more than one way to define the relation $<$ so that $\begin{bmatrix}3\\0\end{bmatrix} < \begin{bmatrix}2\\3\end{bmatrix}$.

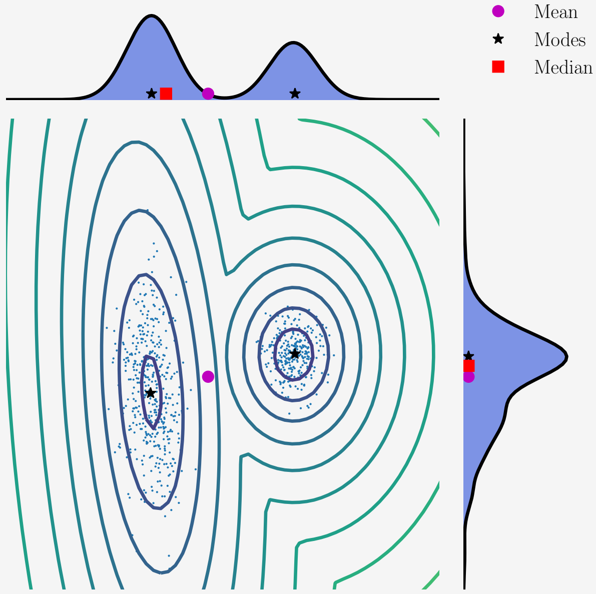


Figure 6.2 Illustration of the mean, mode, and median for a two-dimensional dataset, as well as its marginal densities.

Remark. The expected value (Definition 6.3) is a linear operator. For example, given a real-valued function $f(\boldsymbol{x}) = ag(\boldsymbol{x}) + bh(\boldsymbol{x})$ where $a, b \in \mathbb{R}$ and $\boldsymbol{x} \in \mathbb{R}^D$, we obtain

$$\mathbb{E}_X[f(\boldsymbol{x})] = \int f(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{6.34a}$$

$$= \int [ag(\boldsymbol{x}) + bh(\boldsymbol{x})]p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{6.34b}$$

$$= a\int g(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}x + b\int h(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{6.34c}$$

$$= a\mathbb{E}_X[g(\boldsymbol{x})] + b\mathbb{E}_X[h(\boldsymbol{x})]\,. \tag{6.34d}$$

$\diamondsuit$

For two random variables, we may wish to characterize their correspondence to each other. The covariance intuitively represents the notion of how dependent random variables are to one another.

covariance

Definition 6.5 (Covariance (Univariate))**.** The *covariance* between two univariate random variables $X, Y \in \mathbb{R}$ is given by the expected product of their deviations from their respective means, i.e.,

$$\mathrm{Cov}_{X,Y}[x, y] := \mathbb{E}_{X,Y}\big[(x - \mathbb{E}_X[x])(y - \mathbb{E}_Y[y])\big]\,. \tag{6.35}$$

Terminology: The covariance of multivariate random variables $\mathrm{Cov}[x, y]$ is sometimes referred to as cross-covariance, with covariance referring to $\mathrm{Cov}[x, x]$.

Remark. When the random variable associated with the expectation or covariance is clear by its arguments, the subscript is often suppressed (for example, $\mathbb{E}_X[x]$ is often written as $\mathbb{E}[x]$). ♢

By using the linearity of expectations, the expression in Definition 6.5 can be rewritten as the expected value of the product minus the product of the expected values, i.e.,

$$\mathrm{Cov}[x, y] = \mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y]\,. \tag{6.36}$$

variance

standard deviation

The covariance of a variable with itself $\mathrm{Cov}[x, x]$ is called the *variance* and is denoted by $\mathbb{V}_X[x]$. The square root of the variance is called the *standard deviation* and is often denoted by $\sigma(x)$. The notion of covariance can be generalized to multivariate random variables.

covariance

Definition 6.6 (Covariance (Multivariate))**.** If we consider two multivariate random variables X and Y with states $\boldsymbol{x} \in \mathbb{R}^D$ and $\boldsymbol{y} \in \mathbb{R}^E$ respectively, the *covariance* between X and Y is defined as

$$\mathrm{Cov}[\boldsymbol{x}, \boldsymbol{y}] = \mathbb{E}[\boldsymbol{x}\boldsymbol{y}^\top] - \mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{y}]^\top = \mathrm{Cov}[\boldsymbol{y}, \boldsymbol{x}]^\top \in \mathbb{R}^{D\times E}\,. \tag{6.37}$$

Definition 6.6 can be applied with the same multivariate random variable in both arguments, which results in a useful concept that intuitively captures the "spread" of a random variable. For a multivariate random variable, the variance describes the relation between individual dimensions of the random variable.

variance

Definition 6.7 (Variance)**.** The *variance* of a random variable X with states $\boldsymbol{x} \in \mathbb{R}^D$ and a mean vector $\boldsymbol{\mu} \in \mathbb{R}^D$ is defined as

$$\mathbb{V}_X[\boldsymbol{x}] = \mathrm{Cov}_X[\boldsymbol{x}, \boldsymbol{x}] \tag{6.38a}$$

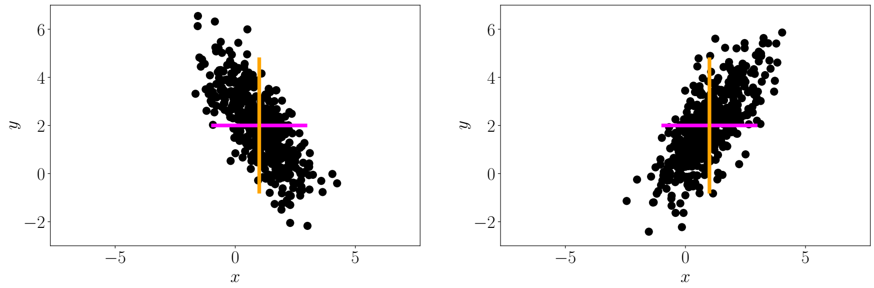
$$= \mathbb{E}_X[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top] = \mathbb{E}_X[\boldsymbol{x}\boldsymbol{x}^\top] - \mathbb{E}_X[\boldsymbol{x}]\mathbb{E}_X[\boldsymbol{x}]^\top \tag{6.38b}$$

$$= \begin{bmatrix} \mathrm{Cov}[x_1, x_1] & \mathrm{Cov}[x_1, x_2] & \dots & \mathrm{Cov}[x_1, x_D] \\ \mathrm{Cov}[x_2, x_1] & \mathrm{Cov}[x_2, x_2] & \dots & \mathrm{Cov}[x_2, x_D] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[x_D, x_1] & \dots & \dots & \mathrm{Cov}[x_D, x_D] \end{bmatrix}. \tag{6.38c}$$

covariance matrix

marginal

The $D \times D$ matrix in (6.38c) is called the *covariance matrix* of the multivariate random variable X. The covariance matrix is symmetric and positive semidefinite and tells us something about the spread of the data. On its diagonal, the covariance matrix contains the variances of the *marginals*



(a) x and y are negatively correlated. (b) x and y are positively correlated.

Figure 6.3 Two-dimensional datasets with identical means and variances along each axis (colored lines) but with different covariances.

$$p(x_i) = \int p(x_1, \ldots, x_D) \mathrm{d}x_{\backslash i} \,, \tag{6.39}$$

where "$\backslash i$" denotes "all variables but i". The off-diagonal entries are the *cross-covariance* terms $\mathrm{Cov}[x_i, x_j]$ for $i, j = 1, \ldots, D,\ i \neq j$.

cross-covariance

Remark. In this book, we generally assume that covariance matrices are positive definite to enable better intuition. We therefore do not discuss corner cases that result in positive semidefinite (low-rank) covariance matrices. ♢

When we want to compare the covariances between different pairs of random variables, it turns out that the variance of each random variable affects the value of the covariance. The normalized version of covariance is called the correlation.

Definition 6.8 (Correlation)**.** The *correlation* between two random variables X, Y is given by

correlation

$$\mathrm{corr}[x, y] = \frac{\mathrm{Cov}[x, y]}{\sqrt{\mathbb{V}[x]\mathbb{V}[y]}} \in [-1, 1] \,. \tag{6.40}$$

The correlation matrix is the covariance matrix of standardized random variables, $x/\sigma(x)$. In other words, each random variable is divided by its standard deviation (the square root of the variance) in the correlation matrix.

The covariance (and correlation) indicate how two random variables are related; see Figure 6.3. Positive correlation $\mathrm{corr}[x, y]$ means that when x grows, then y is also expected to grow. Negative correlation means that as x increases, then y decreases.

6.4.2 Empirical Means and Covariances

The definitions in Section 6.4.1 are often also called the *population mean and covariance*, as it refers to the true statistics for the population. In machine learning, we need to learn from empirical observations of data. Consider a random variable X. There are two conceptual steps to go from

population mean and covariance

population statistics to the realization of empirical statistics. First, we use the fact that we have a finite dataset (of size N) to construct an empirical statistic that is a function of a finite number of identical random variables, $X_1, \ldots, X_N$. Second, we observe the data, that is, we look at the realization $x_1, \ldots, x_N$ of each of the random variables and apply the empirical statistic.

empirical mean
sample mean

Specifically, for the mean (Definition 6.4), given a particular dataset we can obtain an estimate of the mean, which is called the *empirical mean* or *sample mean*. The same holds for the empirical covariance.

empirical mean

Definition 6.9 (Empirical Mean and Covariance)**.** The *empirical mean* vector is the arithmetic average of the observations for each variable, and it is defined as

$$\bar{\boldsymbol{x}} := \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n\,, \tag{6.41}$$

where $\boldsymbol{x}_n \in \mathbb{R}^D$.

empirical covariance

Similar to the empirical mean, the *empirical covariance* matrix is a $D \times D$ matrix

$$\boldsymbol{\Sigma} := \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \bar{\boldsymbol{x}})(\boldsymbol{x}_n - \bar{\boldsymbol{x}})^\top . \tag{6.42}$$

Throughout the book, we use the empirical covariance, which is a biased estimate. The unbiased (sometimes called corrected) covariance has the factor $N-1$ in the denominator instead of N.

To compute the statistics for a particular dataset, we would use the realizations (observations) $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ and use (6.41) and (6.42). Empirical covariance matrices are symmetric, positive semidefinite (see Section 3.2.3).

6.4.3 Three Expressions for the Variance

The derivations are exercises at the end of this chapter.

We now focus on a single random variable X and use the preceding empirical formulas to derive three possible expressions for the variance. The following derivation is the same for the population variance, except that we need to take care of integrals. The standard definition of variance, corresponding to the definition of covariance (Definition 6.5), is the expectation of the squared deviation of a random variable X from its expected value μ, i.e.,

$$\mathbb{V}_X[x] := \mathbb{E}_X[(x-\mu)^2]\,. \tag{6.43}$$

The expectation in (6.43) and the mean $\mu = \mathbb{E}_X(x)$ are computed using (6.32), depending on whether X is a discrete or continuous random variable. The variance as expressed in (6.43) is the mean of a new random variable $Z := (X-\mu)^2$.

When estimating the variance in (6.43) empirically, we need to resort to a two-pass algorithm: one pass through the data to calculate the mean μ using (6.41), and then a second pass using this estimate $\hat{\mu}$ calculate the variance. It turns out that we can avoid two passes by rearranging the terms. The formula in (6.43) can be converted to the so-called *raw-score formula for variance*:

raw-score formula for variance

$$\mathbb{V}_X[x] = \mathbb{E}_X[x^2] - (\mathbb{E}_X[x])^2 \,. \tag{6.44}$$

The expression in (6.44) can be remembered as "the mean of the square minus the square of the mean". It can be calculated empirically in one pass through data since we can accumulate x_i (to calculate the mean) and x_i^2 simultaneously, where x_i is the ith observation. Unfortunately, if implemented in this way, it can be numerically unstable. The raw-score version of the variance can be useful in machine learning, e.g., when deriving the bias–variance decomposition (Bishop, 2006).

If the two terms in (6.44) are huge and approximately equal, we may suffer from an unnecessary loss of numerical precision in floating-point arithmetic.

A third way to understand the variance is that it is a sum of pairwise differences between all pairs of observations. Consider a sample $x_1, \ldots, x_N$ of realizations of random variable X, and we compute the squared difference between pairs of x_i and x_j. By expanding the square, we can show that the sum of N^2 pairwise differences is the empirical variance of the observations:

$$\frac{1}{N^2}\sum_{i,j=1}^{N}(x_i - x_j)^2 = 2\left[\frac{1}{N}\sum_{i=1}^{N}x_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N}x_i\right)^2\right] . \tag{6.45}$$

We see that (6.45) is twice the raw-score expression (6.44). This means that we can express the sum of pairwise distances (of which there are N^2 of them) as a sum of deviations from the mean (of which there are N). Geometrically, this means that there is an equivalence between the pairwise distances and the distances from the center of the set of points. From a computational perspective, this means that by computing the mean (N terms in the summation), and then computing the variance (again N terms in the summation), we can obtain an expression (left-hand side of (6.45)) that has N^2 terms.

6.4.4 Sums and Transformations of Random Variables

We may want to model a phenomenon that cannot be well explained by textbook distributions (we introduce some in Sections 6.5 and 6.6), and hence may perform simple manipulations of random variables (such as adding two random variables).

Consider two random variables X, Y with states $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^D$. Then:

$$\mathbb{E}[\boldsymbol{x} + \boldsymbol{y}] = \mathbb{E}[\boldsymbol{x}] + \mathbb{E}[\boldsymbol{y}] \tag{6.46}$$

$$\mathbb{E}[\boldsymbol{x} - \boldsymbol{y}] = \mathbb{E}[\boldsymbol{x}] - \mathbb{E}[\boldsymbol{y}] \tag{6.47}$$

$$\mathbb{V}[\boldsymbol{x} + \boldsymbol{y}] = \mathbb{V}[\boldsymbol{x}] + \mathbb{V}[\boldsymbol{y}] + \mathrm{Cov}[\boldsymbol{x}, \boldsymbol{y}] + \mathrm{Cov}[\boldsymbol{y}, \boldsymbol{x}] \tag{6.48}$$

$$\mathbb{V}[\boldsymbol{x} - \boldsymbol{y}] = \mathbb{V}[\boldsymbol{x}] + \mathbb{V}[\boldsymbol{y}] - \mathrm{Cov}[\boldsymbol{x}, \boldsymbol{y}] - \mathrm{Cov}[\boldsymbol{y}, \boldsymbol{x}] \,. \tag{6.49}$$

Mean and (co)variance exhibit some useful properties when it comes to affine transformation of random variables. Consider a random variable X with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ and a (deterministic) affine transformation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ of $\boldsymbol{x}$. Then $\boldsymbol{y}$ is itself a random variable whose mean vector and covariance matrix are given by

$$\mathbb{E}_Y[\boldsymbol{y}] = \mathbb{E}_X[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}] = \boldsymbol{A}\mathbb{E}_X[\boldsymbol{x}] + \boldsymbol{b} = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}\,, \tag{6.50}$$

$$\mathbb{V}_Y[\boldsymbol{y}] = \mathbb{V}_X[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}] = \mathbb{V}_X[\boldsymbol{A}\boldsymbol{x}] = \boldsymbol{A}\mathbb{V}_X[\boldsymbol{x}]\boldsymbol{A}^\top = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top\,, \tag{6.51}$$

This can be shown directly by using the definition of the mean and covariance.

respectively. Furthermore,

$$\begin{aligned}
\mathrm{Cov}[\boldsymbol{x}, \boldsymbol{y}] &= \mathbb{E}[\boldsymbol{x}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})^\top] - \mathbb{E}[\boldsymbol{x}]\mathbb{E}[\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}]^\top && (6.52a)\\
&= \mathbb{E}[\boldsymbol{x}]\boldsymbol{b}^\top + \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]\boldsymbol{A}^\top - \boldsymbol{\mu}\boldsymbol{b}^\top - \boldsymbol{\mu}\boldsymbol{\mu}^\top\boldsymbol{A}^\top && (6.52b)\\
&= \boldsymbol{\mu}\boldsymbol{b}^\top - \boldsymbol{\mu}\boldsymbol{b}^\top + \left(\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] - \boldsymbol{\mu}\boldsymbol{\mu}^\top\right)\boldsymbol{A}^\top && (6.52c)\\
&\overset{(6.38b)}{=} \boldsymbol{\Sigma}\boldsymbol{A}^\top\,, && (6.52d)
\end{aligned}$$

where $\boldsymbol{\Sigma} = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] - \boldsymbol{\mu}\boldsymbol{\mu}^\top$ is the covariance of X.

6.4.5 Statistical Independence

statistical independence

Definition 6.10 (Independence)**.** Two random variables X, Y are *statistically independent* if and only if

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y})\,. \tag{6.53}$$

Intuitively, two random variables X and Y are independent if the value of $\boldsymbol{y}$ (once known) does not add any additional information about $\boldsymbol{x}$ (and vice versa). If X, Y are (statistically) independent, then

- $p(\boldsymbol{y}\,|\,\boldsymbol{x}) = p(\boldsymbol{y})$
- $p(\boldsymbol{x}\,|\,\boldsymbol{y}) = p(\boldsymbol{x})$
- $\mathbb{V}_{X,Y}[\boldsymbol{x} + \boldsymbol{y}] = \mathbb{V}_X[\boldsymbol{x}] + \mathbb{V}_Y[\boldsymbol{y}]$
- $\mathrm{Cov}_{X,Y}[\boldsymbol{x}, \boldsymbol{y}] = \boldsymbol{0}$

The last point may not hold in converse, i.e., two random variables can have covariance zero but are not statistically independent. To understand why, recall that covariance measures only linear dependence. Therefore, random variables that are nonlinearly dependent could have covariance zero.

Example 6.5

Consider a random variable X with zero mean ($\mathbb{E}_X[x] = 0$) and also $\mathbb{E}_X[x^3] = 0$. Let $y = x^2$ (hence, Y is dependent on X) and consider the covariance (6.36) between X and Y. But this gives

$$\mathrm{Cov}[x, y] = \mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y] = \mathbb{E}[x^3] = 0\,. \tag{6.54}$$

In machine learning, we often consider problems that can be modeled as *independent and identically distributed* (*i.i.d.*) random variables, $X_1, \ldots, X_N$. For more than two random variables, the word "independent" (Definition 6.10) usually refers to mutually independent random variables, where all subsets are independent (see Pollard (2002, chapter 4) and Jacod and Protter (2004, chapter 3)). The phrase "identically distributed" means that all the random variables are from the same distribution.

independent and identically distributed i.i.d.

Another concept that is important in machine learning is conditional independence.

Definition 6.11 (Conditional Independence)**.** Two random variables X and Y are *conditionally independent* given Z if and only if

conditionally independent

$$p(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{z}) = p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{y} \mid \boldsymbol{z}) \quad \text{for all} \quad \boldsymbol{z} \in \mathcal{Z}\,, \tag{6.55}$$

where $\mathcal{Z}$ is the set of states of random variable Z. We write $X \perp\!\!\!\perp Y \mid Z$ to denote that X is conditionally independent of Y given Z.

Definition 6.11 requires that the relation in (6.55) must hold true for every value of $\boldsymbol{z}$. The interpretation of (6.55) can be understood as "given knowledge about $\boldsymbol{z}$, the distribution of $\boldsymbol{x}$ and $\boldsymbol{y}$ factorizes". Independence can be cast as a special case of conditional independence if we write $X \perp\!\!\!\perp Y \mid \emptyset$. By using the product rule of probability (6.22), we can expand the left-hand side of (6.55) to obtain

$$p(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{z}) = p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{z})p(\boldsymbol{y} \mid \boldsymbol{z})\,. \tag{6.56}$$

By comparing the right-hand side of (6.55) with (6.56), we see that $p(\boldsymbol{y} \mid \boldsymbol{z})$ appears in both of them so that

$$p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{z}) = p(\boldsymbol{x} \mid \boldsymbol{z})\,. \tag{6.57}$$

Equation (6.57) provides an alternative definition of conditional independence, i.e., $X \perp\!\!\!\perp Y \mid Z$. This alternative presentation provides the interpretation "given that we know $\boldsymbol{z}$, knowledge about $\boldsymbol{y}$ does not change our knowledge of $\boldsymbol{x}$".

6.4.6 Inner Products of Random Variables

Recall the definition of inner products from Section 3.2. We can define an inner product between random variables, which we briefly describe in this section. If we have two uncorrelated random variables X, Y, then

Inner products between multivariate random variables can be treated in a similar fashion

$$\mathbb{V}[x + y] = \mathbb{V}[x] + \mathbb{V}[y]\,. \tag{6.58}$$

Since variances are measured in squared units, this looks very much like the Pythagorean theorem for right triangles $c^2 = a^2 + b^2$.

In the following, we see whether we can find a geometric interpretation of the variance relation of uncorrelated random variables in (6.58).

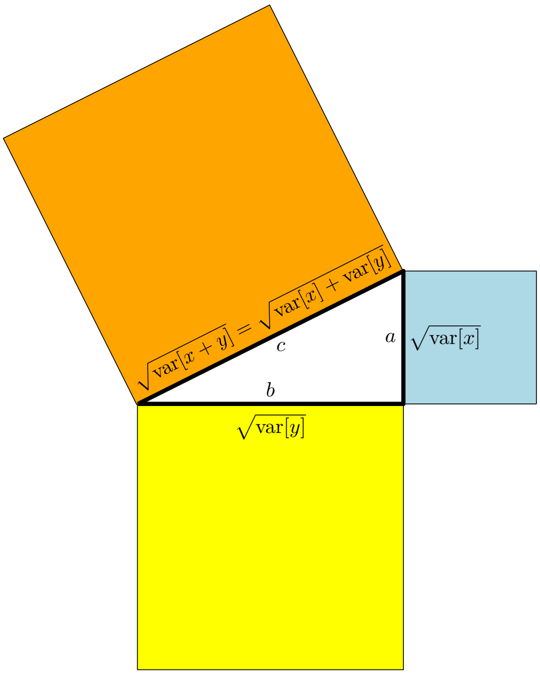


Figure 6.1 Geometry of random variables. If random variables X and Y are uncorrelated, they are orthogonal vectors in a corresponding vector space, and the Pythagorean theorem applies.

Random variables can be considered vectors in a vector space, and we can define inner products to obtain geometric properties of random variables (Eaton, 2007). If we define

$$\langle X, Y \rangle := \mathrm{Cov}[x, y] \tag{6.59}$$

for zero mean random variables X and Y, we obtain an inner product. We see that the covariance is symmetric, positive definite, and linear in either argument. The length of a random variable is

$\mathrm{Cov}[x, x] = 0 \iff x = 0$
$\mathrm{Cov}[\alpha x + z, y] = \alpha\,\mathrm{Cov}[x, y] + \mathrm{Cov}[z, y]$ for $\alpha \in \mathbb{R}$.

$$\|X\| = \sqrt{\mathrm{Cov}[x, x]} = \sqrt{\mathbb{V}[x]} = \sigma[x]\,, \tag{6.60}$$

i.e., its standard deviation. The "longer" the random variable, the more uncertain it is; and a random variable with length 0 is deterministic.

If we look at the angle θ between two random variables X, Y, we get

$$\cos\theta = \frac{\langle X, Y \rangle}{\|X\|\,\|Y\|} = \frac{\mathrm{Cov}[x, y]}{\sqrt{\mathbb{V}[x]\mathbb{V}[y]}}\,, \tag{6.61}$$

which is the correlation (Definition 6.8) between the two random variables. This means that we can think of correlation as the cosine of the angle between two random variables when we consider them geometrically. We know from Definition 3.7 that $X \perp Y \iff \langle X, Y \rangle = 0$. In our case, this means that X and Y are orthogonal if and only if $\mathrm{Cov}[x, y] = 0$, i.e., they are uncorrelated. Figure 6.1 illustrates this relationship.

Remark. While it is tempting to use the Euclidean distance (constructed

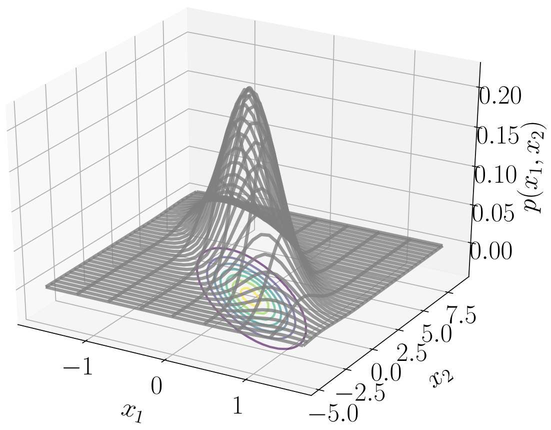


Figure 6.1 Gaussian distribution of two random variables x_1 and x_2.

from the preceding definition of inner products) to compare probability distributions, it is unfortunately not the best way to obtain distances between distributions. Recall that the probability mass (or density) is positive and needs to add up to 1. These constraints mean that distributions live on something called a statistical manifold. The study of this space of probability distributions is called information geometry. Computing distances between distributions are often done using Kullback-Leibler divergence, which is a generalization of distances that account for properties of the statistical manifold. Just like the Euclidean distance is a special case of a metric (Section 3.3), the Kullback-Leibler divergence is a special case of two more general classes of divergences called Bregman divergences and f-divergences. The study of divergences is beyond the scope of this book, and we refer for more details to the recent book by Amari (2016), one of the founders of the field of information geometry. $\diamondsuit$

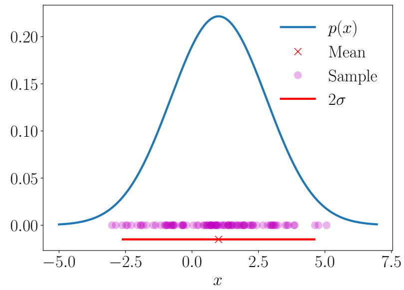
6.5 Gaussian Distribution

The Gaussian distribution is the most well-studied probability distribution for continuous-valued random variables. It is also referred to as the *normal distribution*. Its importance originates from the fact that it has many computationally convenient properties, which we will be discussing in the following. In particular, we will use it to define the likelihood and prior for linear regression (Chapter 9), and consider a mixture of Gaussians for density estimation (Chapter 11).

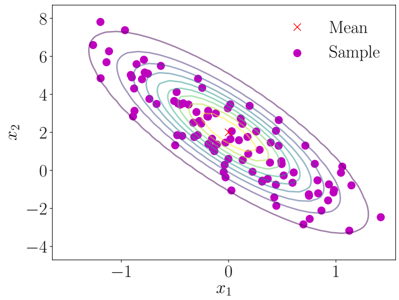
normal distribution

The Gaussian distribution arises naturally when we consider sums of independent and identically distributed random variables. This is known as the central limit theorem (Grinstead and Snell, 1997).

There are many other areas of machine learning that also benefit from using a Gaussian distribution, for example Gaussian processes, variational inference, and reinforcement learning. It is also widely used in other application areas such as signal processing (e.g., Kalman filter), control (e.g., linear quadratic regulator), and statistics (e.g., hypothesis testing).



(a) Univariate (one-dimensional) Gaussian; The red cross shows the mean and the red line shows the extent of the variance.



(b) Multivariate (two-dimensional) Gaussian, viewed from top. The red cross shows the mean and the colored lines show the contour lines of the density.

Figure 6.2 Gaussian distributions overlaid with 100 samples. (a) One-dimensional case; (b) two-dimensional case.

For a univariate random variable, the Gaussian distribution has a density that is given by

$$p(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \quad (6.62)$$

multivariate Gaussian distribution

mean vector

covariance matrix

The *multivariate Gaussian distribution* is fully characterized by a *mean vector* $\boldsymbol{\mu}$ and a *covariance matrix* $\boldsymbol{\Sigma}$ and defined as

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{D}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \quad (6.63)$$

Also known as a multivariate normal distribution.

where $\boldsymbol{x} \in \mathbb{R}^D$. We write $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ or $X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Figure 6.1 shows a bivariate Gaussian (mesh), with the corresponding contour plot. Figure 6.2 shows a univariate Gaussian and a bivariate Gaussian with corresponding samples. The special case of the Gaussian with zero mean and identity covariance, that is, $\boldsymbol{\mu} = \boldsymbol{0}$ and $\boldsymbol{\Sigma} = \boldsymbol{I}$, is referred to as the *standard normal distribution*.

standard normal distribution

Gaussians are widely used in statistical estimation and machine learning as they have closed-form expressions for marginal and conditional distributions. In Chapter 9, we use these closed-form expressions extensively for linear regression. A major advantage of modeling with Gaussian random variables is that variable transformations (Section 6.7) are often not needed. Since the Gaussian distribution is fully specified by its mean and covariance, we often can obtain the transformed distribution by applying the transformation to the mean and covariance of the random variable.

6.5.1 Marginals and Conditionals of Gaussians are Gaussians

In the following, we present marginalization and conditioning in the general case of multivariate random variables. If this is confusing at first reading, the reader is advised to consider two univariate random variables instead. Let X and Y be two multivariate random variables, that may have different dimensions. To consider the effect of applying the sum rule of probability and the effect of conditioning, we explicitly write the Gaussian distribution in terms of the concatenated states $[\boldsymbol{x}^\top, \boldsymbol{y}^\top]$,

$$p(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy}\end{bmatrix}\right). \tag{6.64}$$

where $\boldsymbol{\Sigma}_{xx} = \mathrm{Cov}[\boldsymbol{x}, \boldsymbol{x}]$ and $\boldsymbol{\Sigma}_{yy} = \mathrm{Cov}[\boldsymbol{y}, \boldsymbol{y}]$ are the marginal covariance matrices of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and $\boldsymbol{\Sigma}_{xy} = \mathrm{Cov}[\boldsymbol{x}, \boldsymbol{y}]$ is the cross-covariance matrix between $\boldsymbol{x}$ and $\boldsymbol{y}$.

The conditional distribution $p(\boldsymbol{x} \mid \boldsymbol{y})$ is also Gaussian (illustrated in Figure 6.3(c)) and given by (derived in Section 2.3 of Bishop, 2006)

$$p(\boldsymbol{x} \mid \boldsymbol{y}) = \mathcal{N}\big(\boldsymbol{\mu}_{x \mid y}, \boldsymbol{\Sigma}_{x \mid y}\big) \tag{6.65}$$

$$\boldsymbol{\mu}_{x \mid y} = \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y) \tag{6.66}$$

$$\boldsymbol{\Sigma}_{x \mid y} = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}\,. \tag{6.67}$$

Note that in the computation of the mean in (6.66), the $\boldsymbol{y}$-value is an observation and no longer random.

Remark. The conditional Gaussian distribution shows up in many places, where we are interested in posterior distributions:

- The Kalman filter (Kalman, 1960), one of the most central algorithms for state estimation in signal processing, does nothing but computing Gaussian conditionals of joint distributions (Deisenroth and Ohlsson, 2011; Särkkä, 2013).
- Gaussian processes (Rasmussen and Williams, 2006), which are a practical implementation of a distribution over functions. In a Gaussian process, we make assumptions of joint Gaussianity of random variables. By (Gaussian) conditioning on observed data, we can determine a posterior distribution over functions.
- Latent linear Gaussian models (Roweis and Ghahramani, 1999; Murphy, 2012), which include probabilistic principal component analysis (PPCA) (Tipping and Bishop, 1999). We will look at PPCA in more detail in Section 10.7.

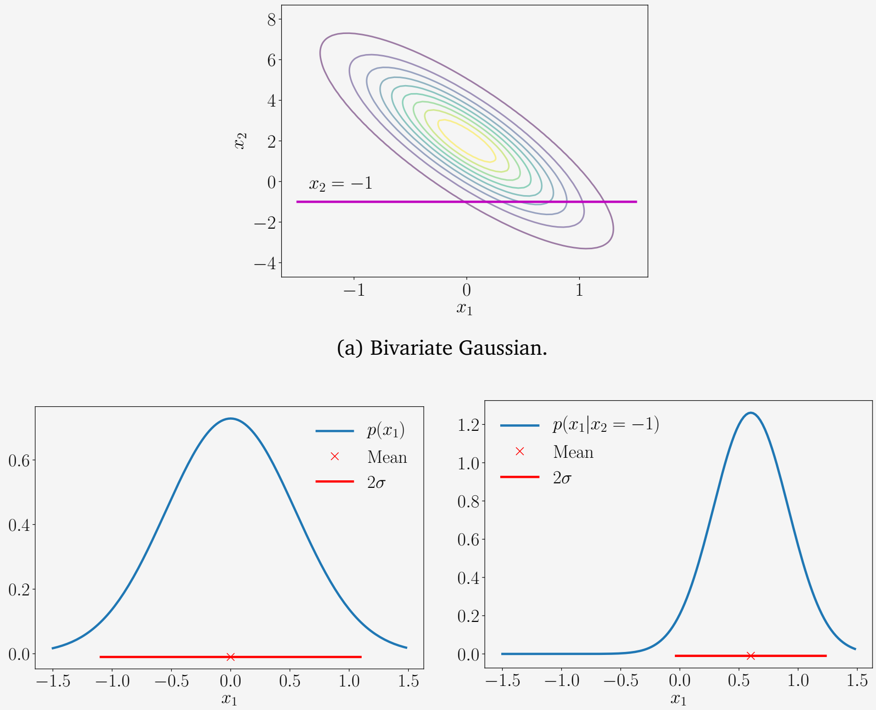
$\diamondsuit$

The marginal distribution $p(\boldsymbol{x})$ of a joint Gaussian distribution $p(\boldsymbol{x}, \boldsymbol{y})$ (see (6.64)) is itself Gaussian and computed by applying the sum rule (6.20) and given by

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{y})\mathrm{d}\boldsymbol{y} = \mathcal{N}\big(\boldsymbol{x} \mid \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx}\big)\,. \tag{6.68}$$

The corresponding result holds for $p(\boldsymbol{y})$, which is obtained by marginalizing with respect to $\boldsymbol{x}$. Intuitively, looking at the joint distribution in (6.64), we ignore (i.e., integrate out) everything we are not interested in. This is illustrated in Figure 6.3(b).

Example 6.6



(a) Bivariate Gaussian.

(b) Marginal distribution.

(c) Conditional distribution.

Figure 6.3 (a) Bivariate Gaussian; (b) marginal of a joint Gaussian distribution is Gaussian; (c) the conditional distribution of a Gaussian is also Gaussian.

Consider the bivariate Gaussian distribution (illustrated in Figure 6.3):

$$p(x_1, x_2) = \mathcal{N}\left(\begin{bmatrix}0\\2\end{bmatrix}, \begin{bmatrix}0.3 & -1\\-1 & 5\end{bmatrix}\right). \tag{6.69}$$

We can compute the parameters of the univariate Gaussian, conditioned on $x_2 = -1$, by applying (6.66) and (6.67) to obtain the mean and variance respectively. Numerically, this is

$$\mu_{x_1 \mid x_2=-1} = 0 + (-1) \cdot 0.2 \cdot (-1 - 2) = 0.6 \tag{6.70}$$

and

$$\sigma^2_{x_1 \mid x_2=-1} = 0.3 - (-1) \cdot 0.2 \cdot (-1) = 0.1\,. \tag{6.71}$$

Therefore, the conditional Gaussian is given by

$$p(x_1 \mid x_2 = -1) = \mathcal{N}\big(0.6,\, 0.1\big)\,. \tag{6.72}$$

The marginal distribution $p(x_1)$, in contrast, can be obtained by applying (6.68), which is essentially using the mean and variance of the random variable x_1, giving us

$$p(x_1) = \mathcal{N}\big(0,\, 0.3\big)\,. \tag{6.73}$$

6.5.2 Product of Gaussian Densities

For linear regression (Chapter 9), we need to compute a Gaussian likelihood. Furthermore, we may wish to assume a Gaussian prior (Section 9.3). We apply Bayes' Theorem to compute the posterior, which results in a multiplication of the likelihood and the prior, that is, the multiplication of two Gaussian densities. The *product* of two Gaussians $\mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{a},\, \boldsymbol{A}\big)\mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{b},\, \boldsymbol{B}\big)$ is a Gaussian distribution scaled by a $c \in \mathbb{R}$, given by $c\mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{c},\, \boldsymbol{C}\big)$ with

The derivation is an exercise at the end of this chapter.

$$\boldsymbol{C} = (\boldsymbol{A}^{-1} + \boldsymbol{B}^{-1})^{-1} \tag{6.74}$$

$$\boldsymbol{c} = \boldsymbol{C}(\boldsymbol{A}^{-1}\boldsymbol{a} + \boldsymbol{B}^{-1}\boldsymbol{b}) \tag{6.75}$$

$$c = (2\pi)^{-\frac{D}{2}} |\boldsymbol{A} + \boldsymbol{B}|^{-\frac{1}{2}} \exp\big(-\tfrac{1}{2}(\boldsymbol{a} - \boldsymbol{b})^\top (\boldsymbol{A} + \boldsymbol{B})^{-1}(\boldsymbol{a} - \boldsymbol{b})\big)\,. \tag{6.76}$$

The scaling constant c itself can be written in the form of a Gaussian density either in $\boldsymbol{a}$ or in $\boldsymbol{b}$ with an "inflated" covariance matrix $\boldsymbol{A} + \boldsymbol{B}$, i.e., $c = \mathcal{N}\big(\boldsymbol{a} \,|\, \boldsymbol{b},\, \boldsymbol{A} + \boldsymbol{B}\big) = \mathcal{N}\big(\boldsymbol{b} \,|\, \boldsymbol{a},\, \boldsymbol{A} + \boldsymbol{B}\big)$.

Remark. For notation convenience, we will sometimes use $\mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{m},\, \boldsymbol{S}\big)$ to describe the functional form of a Gaussian density even if $\boldsymbol{x}$ is not a random variable. We have just done this in the preceding demonstration when we wrote

$$c = \mathcal{N}\big(\boldsymbol{a} \,|\, \boldsymbol{b},\, \boldsymbol{A} + \boldsymbol{B}\big) = \mathcal{N}\big(\boldsymbol{b} \,|\, \boldsymbol{a},\, \boldsymbol{A} + \boldsymbol{B}\big)\,. \tag{6.77}$$

Here, neither $\boldsymbol{a}$ nor $\boldsymbol{b}$ are random variables. However, writing c in this way is more compact than (6.76). ◊

6.5.3 Sums and Linear Transformations

If X, Y are independent Gaussian random variables (i.e., the joint distribution is given as $p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y})$) with $p(\boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{x} \,|\, \boldsymbol{\mu}_x,\, \boldsymbol{\Sigma}_x\big)$ and $p(\boldsymbol{y}) = \mathcal{N}\big(\boldsymbol{y} \,|\, \boldsymbol{\mu}_y,\, \boldsymbol{\Sigma}_y\big)$, then $\boldsymbol{x} + \boldsymbol{y}$ is also Gaussian distributed and given by

$$p(\boldsymbol{x} + \boldsymbol{y}) = \mathcal{N}\big(\boldsymbol{\mu}_x + \boldsymbol{\mu}_y,\, \boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_y\big)\,. \tag{6.78}$$

Knowing that $p(\boldsymbol{x} + \boldsymbol{y})$ is Gaussian, the mean and covariance matrix can be determined immediately using the results from (6.46) through (6.49). This property will be important when we consider i.i.d. Gaussian noise acting on random variables, as is the case for linear regression (Chapter 9).

Example 6.7
Since expectations are linear operations, we can obtain the weighted sum of independent Gaussian random variables

$$p(a\boldsymbol{x} + b\boldsymbol{y}) = \mathcal{N}\big(a\boldsymbol{\mu}_x + b\boldsymbol{\mu}_y,\, a^2\boldsymbol{\Sigma}_x + b^2\boldsymbol{\Sigma}_y\big)\,. \tag{6.79}$$

Remark. A case that will be useful in Chapter 11 is the weighted sum of Gaussian densities. This is different from the weighted sum of Gaussian random variables. ◊

In Theorem 6.12, the random variable x is from a density that is a mixture of two densities $p_1(x)$ and $p_2(x)$, weighted by α. The theorem can be generalized to the multivariate random variable case, since linearity of expectations holds also for multivariate random variables. However, the idea of a squared random variable needs to be replaced by $\boldsymbol{x}\boldsymbol{x}^\top$.

Theorem 6.12. *Consider a mixture of two univariate Gaussian densities*

$$p(x) = \alpha p_1(x) + (1-\alpha)p_2(x)\,, \tag{6.80}$$

where the scalar $0 < \alpha < 1$ *is the mixture weight, and* $p_1(x)$ *and* $p_2(x)$ *are univariate Gaussian densities (Equation* (6.62)*) with different parameters, i.e.,* $(\mu_1, \sigma_1^2) \neq (\mu_2, \sigma_2^2)$.

Then the mean of the mixture density $p(x)$ *is given by the weighted sum of the means of each random variable:*

$$\mathbb{E}[x] = \alpha\mu_1 + (1-\alpha)\mu_2\,. \tag{6.81}$$

The variance of the mixture density $p(x)$ *is given by*

$$\mathbb{V}[x] = \left[\alpha\sigma_1^2 + (1-\alpha)\sigma_2^2\right] + \left(\left[\alpha\mu_1^2 + (1-\alpha)\mu_2^2\right] - \left[\alpha\mu_1 + (1-\alpha)\mu_2\right]^2\right). \tag{6.82}$$

Proof The mean of the mixture density $p(x)$ is given by the weighted sum of the means of each random variable. We apply the definition of the mean (Definition 6.4), and plug in our mixture (6.80), which yields

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} x p(x)\mathrm{d}x \tag{6.83a}$$

$$= \int_{-\infty}^{\infty} \left(\alpha x p_1(x) + (1-\alpha) x p_2(x)\right)\mathrm{d}x \tag{6.83b}$$

$$= \alpha\int_{-\infty}^{\infty} x p_1(x)\mathrm{d}x + (1-\alpha)\int_{-\infty}^{\infty} x p_2(x)\mathrm{d}x \tag{6.83c}$$

$$= \alpha\mu_1 + (1-\alpha)\mu_2\,. \tag{6.83d}$$

To compute the variance, we can use the raw-score version of the variance from (6.44), which requires an expression of the expectation of the squared random variable. Here we use the definition of an expectation of a function (the square) of a random variable (Definition 6.3),

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} x^2 p(x)\mathrm{d}x \tag{6.84a}$$

$$= \int_{-\infty}^{\infty} \left(\alpha x^2 p_1(x) + (1-\alpha) x^2 p_2(x)\right)\mathrm{d}x \tag{6.84b}$$

$$= \alpha \int_{-\infty}^{\infty} x^2 p_1(x) \mathrm{d}x + (1-\alpha) \int_{-\infty}^{\infty} x^2 p_2(x) \mathrm{d}x \tag{6.84c}$$
$$= \alpha(\mu_1^2 + \sigma_1^2) + (1-\alpha)(\mu_2^2 + \sigma_2^2)\,, \tag{6.84d}$$

where in the last equality, we again used the raw-score version of the variance (6.44) giving $\sigma^2 = \mathbb{E}[x^2] - \mu^2$. This is rearranged such that the expectation of a squared random variable is the sum of the squared mean and the variance.

Therefore, the variance is given by subtracting (6.83d) from (6.84d),

$$\mathbb{V}[x] = \mathbb{E}[x^2] - (\mathbb{E}[x])^2 \tag{6.85a}$$
$$= \alpha(\mu_1^2 + \sigma_1^2) + (1-\alpha)(\mu_2^2 + \sigma_2^2) - (\alpha\mu_1 + (1-\alpha)\mu_2)^2 \tag{6.85b}$$
$$= \left[\alpha\sigma_1^2 + (1-\alpha)\sigma_2^2\right] + \left(\left[\alpha\mu_1^2 + (1-\alpha)\mu_2^2\right] - \left[\alpha\mu_1 + (1-\alpha)\mu_2\right]^2\right) . \tag{6.85c}$$

□

Remark. The preceding derivation holds for any density, but since the Gaussian is fully determined by the mean and variance, the mixture density can be determined in closed form. ♢

For a mixture density, the individual components can be considered to be conditional distributions (conditioned on the component identity). Equation (6.85c) is an example of the conditional variance formula, also known as the *law of total variance*, which generally states that for two random variables X and Y it holds that $\mathbb{V}_X[x] = \mathbb{E}_Y[\mathbb{V}_X[x|y]] + \mathbb{V}_Y[\mathbb{E}_X[x|y]]$, i.e., the (total) variance of X is the expected conditional variance plus the variance of a conditional mean.

law of total variance

We consider in Example 6.17 a bivariate standard Gaussian random variable X and performed a linear transformation $\boldsymbol{Ax}$ on it. The outcome is a Gaussian random variable with mean zero and covariance $\boldsymbol{AA}^\top$. Observe that adding a constant vector will change the mean of the distribution, without affecting its variance, that is, the random variable $\boldsymbol{x} + \boldsymbol{\mu}$ is Gaussian with mean $\boldsymbol{\mu}$ and identity covariance. Hence, any linear/affine transformation of a Gaussian random variable is Gaussian distributed.

Any linear/affine transformation of a Gaussian random variable is also Gaussian distributed.

Consider a Gaussian distributed random variable $X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For a given matrix $\boldsymbol{A}$ of appropriate shape, let Y be a random variable such that $\boldsymbol{y} = \boldsymbol{Ax}$ is a transformed version of $\boldsymbol{x}$. We can compute the mean of $\boldsymbol{y}$ by exploiting that the expectation is a linear operator (6.50) as follows:

$$\mathbb{E}[\boldsymbol{y}] = \mathbb{E}[\boldsymbol{Ax}] = \boldsymbol{A}\mathbb{E}[\boldsymbol{x}] = \boldsymbol{A\mu}\,. \tag{6.86}$$

Similarly the variance of $\boldsymbol{y}$ can be found by using (6.51):

$$\mathbb{V}[\boldsymbol{y}] = \mathbb{V}[\boldsymbol{Ax}] = \boldsymbol{A}\mathbb{V}[\boldsymbol{x}]\boldsymbol{A}^\top = \boldsymbol{A\Sigma A}^\top\,. \tag{6.87}$$

This means that the random variable $\boldsymbol{y}$ is distributed according to

$$p(\boldsymbol{y}) = \mathcal{N}\big(\boldsymbol{y} \,|\, \boldsymbol{A\mu},\, \boldsymbol{A\Sigma A}^\top\big). \tag{6.88}$$

Let us now consider the reverse transformation: when we know that a random variable has a mean that is a linear transformation of another random variable. For a given full rank matrix $\boldsymbol{A} \in \mathbb{R}^{M \times N}$, where $M \geqslant N$, let $\boldsymbol{y} \in \mathbb{R}^M$ be a Gaussian random variable with mean $\boldsymbol{Ax}$, i.e.,

$$p(\boldsymbol{y}) = \mathcal{N}\big(\boldsymbol{y} \,|\, \boldsymbol{Ax},\, \boldsymbol{\Sigma}\big)\,. \tag{6.89}$$

What is the corresponding probability distribution $p(\boldsymbol{x})$? If $\boldsymbol{A}$ is invertible, then we can write $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y}$ and apply the transformation in the previous paragraph. However, in general $\boldsymbol{A}$ is not invertible, and we use an approach similar to that of the pseudo-inverse (3.57). That is, we premultiply both sides with $\boldsymbol{A}^\top$ and then invert $\boldsymbol{A}^\top\boldsymbol{A}$, which is symmetric and positive definite, giving us the relation

$$\boldsymbol{y} = \boldsymbol{Ax} \iff (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{y} = \boldsymbol{x}\,. \tag{6.90}$$

Hence, $\boldsymbol{x}$ is a linear transformation of $\boldsymbol{y}$, and we obtain

$$p(\boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{x} \,|\, (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{y},\, (\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{\Sigma}\boldsymbol{A}(\boldsymbol{A}^\top\boldsymbol{A})^{-1}\big)\,. \tag{6.91}$$

6.5.4 Sampling from Multivariate Gaussian Distributions

We will not explain the subtleties of random sampling on a computer, and the interested reader is referred to Gentle (2004). In the case of a multivariate Gaussian, this process consists of three stages: first, we need a source of pseudo-random numbers that provide a uniform sample in the interval [0,1]; second, we use a non-linear transformation such as the Box-Müller transform (Devroye, 1986) to obtain a sample from a univariate Gaussian; and third, we collate a vector of these samples to obtain a sample from a multivariate standard normal $\mathcal{N}\big(\boldsymbol{0},\, \boldsymbol{I}\big)$.

For a general multivariate Gaussian, that is, where the mean is non zero and the covariance is not the identity matrix, we use the properties of linear transformations of a Gaussian random variable. Assume we are interested in generating samples $\boldsymbol{x}_i, i = 1, \ldots, n$, from a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. We would like to construct the sample from a sampler that provides samples from the multivariate standard normal $\mathcal{N}\big(\boldsymbol{0},\, \boldsymbol{I}\big)$.

To compute the Cholesky factorization of a matrix, it is required that the matrix is symmetric and positive definite (Section 3.2.3). Covariance matrices possess this property.

To obtain samples from a multivariate normal $\mathcal{N}\big(\boldsymbol{\mu},\, \boldsymbol{\Sigma}\big)$, we can use the properties of a linear transformation of a Gaussian random variable: If $\boldsymbol{x} \sim \mathcal{N}\big(\boldsymbol{0},\, \boldsymbol{I}\big)$, then $\boldsymbol{y} = \boldsymbol{Ax} + \boldsymbol{\mu}$, where $\boldsymbol{AA}^\top = \boldsymbol{\Sigma}$ is Gaussian distributed with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. One convenient choice of $\boldsymbol{A}$ is to use the Cholesky decomposition (Section 4.3) of the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{AA}^\top$. The Cholesky decomposition has the benefit that $\boldsymbol{A}$ is triangular, leading to efficient computation.

6.6 Conjugacy and the Exponential Family

Many of the probability distributions "with names" that we find in statistics textbooks were discovered to model particular types of phenomena. For example, we have seen the Gaussian distribution in Section 6.5. The distributions are also related to each other in complex ways (Leemis and McQueston, 2008). For a beginner in the field, it can be overwhelming to figure out which distribution to use. In addition, many of these distributions were discovered at a time that statistics and computation were done by pencil and paper. It is natural to ask what are meaningful concepts in the computing age (Efron and Hastie, 2016). In the previous section, we saw that many of the operations required for inference can be conveniently calculated when the distribution is Gaussian. It is worth recalling at this point the desiderata for manipulating probability distributions in the machine learning context:

"Computers" used to be a job description.

1. There is some "closure property" when applying the rules of probability, e.g., Bayes' theorem. By closure, we mean that applying a particular operation returns an object of the same type.
2. As we collect more data, we do not need more parameters to describe the distribution.
3. Since we are interested in learning from data, we want parameter estimation to behave nicely.

It turns out that the class of distributions called the *exponential family* provides the right balance of generality while retaining favorable computation and inference properties. Before we introduce the exponential family, let us see three more members of "named" probability distributions, the Bernoulli (Example 6.8), Binomial (Example 6.9), and Beta (Example 6.10) distributions.

exponential family

Example 6.8

The *Bernoulli distribution* is a distribution for a single binary random variable X with state $x \in \{0, 1\}$. It is governed by a single continuous parameter $\mu \in [0, 1]$ that represents the probability of $X = 1$. The Bernoulli distribution $\mathrm{Ber}(\mu)$ is defined as

Bernoulli distribution

$$p(x \mid \mu) = \mu^x (1-\mu)^{1-x}\,, \quad x \in \{0, 1\}\,, \tag{6.92}$$

$$\mathbb{E}[x] = \mu\,, \tag{6.93}$$

$$\mathbb{V}[x] = \mu(1-\mu)\,, \tag{6.94}$$

where $\mathbb{E}[x]$ and $\mathbb{V}[x]$ are the mean and variance of the binary random variable X.

An example where the Bernoulli distribution can be used is when we are interested in modeling the probability of "heads" when flipping a coin.

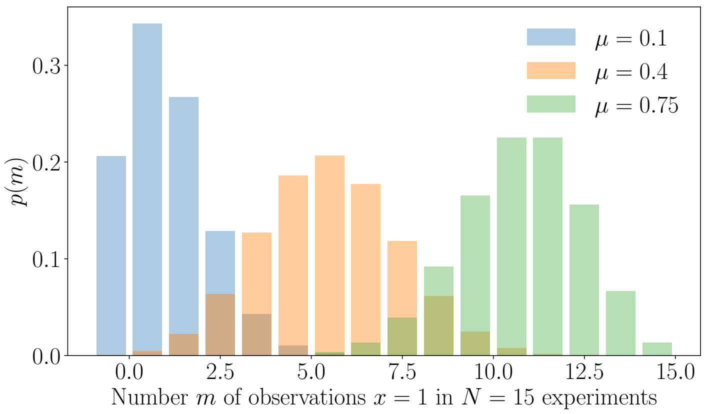


Figure 6.1 Examples of the Binomial distribution for $\mu \in \{0.1, 0.4, 0.75\}$ and $N = 15$.

Remark. The rewriting above of the Bernoulli distribution, where we use Boolean variables as numerical 0 or 1 and express them in the exponents, is a trick that is often used in machine learning textbooks. Another occurence of this is when expressing the Multinomial distribution. ♢

Example 6.9 (Binomial Distribution)

Binomial distribution

The *Binomial distribution* is a generalization of the Bernoulli distribution to a distribution over integers (illustrated in Figure 6.1). In particular, the Binomial can be used to describe the probability of observing m occurrences of $X = 1$ in a set of N samples from a Bernoulli distribution where $p(X = 1) = \mu \in [0, 1]$. The Binomial distribution $\text{Bin}(N, \mu)$ is defined as

$$p(m \mid N, \mu) = \binom{N}{m} \mu^m (1 - \mu)^{N-m}, \tag{6.95}$$

$$\mathbb{E}[m] = N\mu, \tag{6.96}$$

$$\mathbb{V}[m] = N\mu(1 - \mu), \tag{6.97}$$

where $\mathbb{E}[m]$ and $\mathbb{V}[m]$ are the mean and variance of m, respectively.

An example where the Binomial could be used is if we want to describe the probability of observing m “heads” in N coin-flip experiments if the probability for observing head in a single experiment is μ.

Example 6.10 (Beta Distribution)

Beta distribution

We may wish to model a continuous random variable on a finite interval. The *Beta distribution* is a distribution over a continuous random variable $\mu \in [0, 1]$, which is often used to represent the probability for some binary event (e.g., the parameter governing the Bernoulli distribution). The Beta distribution $\text{Beta}(\alpha, \beta)$ (illustrated in Figure 6.2) itself is governed by two parameters $\alpha > 0,\ \beta > 0$ and is defined as

$$p(\mu \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\mu^{\alpha-1}(1-\mu)^{\beta-1} \tag{6.98}$$

$$\mathbb{E}[\mu] = \frac{\alpha}{\alpha+\beta}\,, \qquad \mathbb{V}[\mu] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{6.99}$$

where $\Gamma(\cdot)$ is the Gamma function defined as

$$\Gamma(t) := \int_0^\infty x^{t-1}\exp(-x)dx, \qquad t > 0\,. \tag{6.100}$$

$$\Gamma(t+1) = t\Gamma(t)\,. \tag{6.101}$$

Note that the fraction of Gamma functions in (6.98) normalizes the Beta distribution.

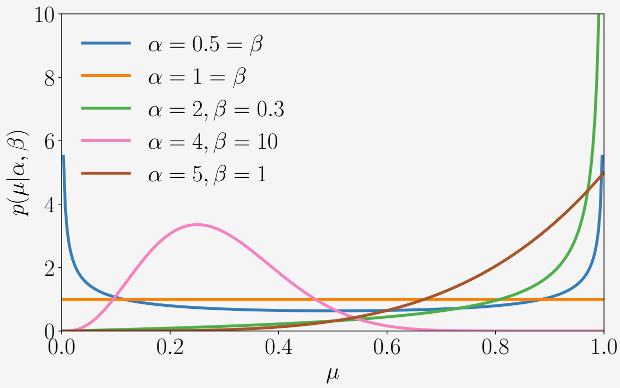


Figure 6.2 Examples of the Beta distribution for different values of α and β.

Intuitively, α moves probability mass toward 1, whereas β moves probability mass toward 0. There are some special cases (Murphy, 2012):

- For $\alpha = 1 = \beta$, we obtain the uniform distribution $\mathcal{U}[0, 1]$.
- For $\alpha, \beta < 1$, we get a bimodal distribution with spikes at 0 and 1.
- For $\alpha, \beta > 1$, the distribution is unimodal.
- For $\alpha, \beta > 1$ and $\alpha = \beta$, the distribution is unimodal, symmetric, and centered in the interval $[0, 1]$, i.e., the mode/mean is at $\frac{1}{2}$.

Remark. There is a whole zoo of distributions with names, and they are related in different ways to each other (Leemis and McQueston, 2008). It is worth keeping in mind that each named distribution is created for a particular reason, but may have other applications. Knowing the reason behind the creation of a particular distribution often allows insight into how to best use it. We introduced the preceding three distributions to be able to illustrate the concepts of conjugacy (Section 6.6.1) and exponential families (Section 6.6.3). ♢

6.6.1 Conjugacy

According to Bayes' theorem (6.23), the posterior is proportional to the product of the prior and the likelihood. The specification of the prior can be tricky for two reasons: First, the prior should encapsulate our knowledge about the problem before we see any data. This is often difficult to describe. Second, it is often not possible to compute the posterior distribution analytically. However, there are some priors that are computationally convenient: *conjugate priors*.

conjugate prior

conjugate

Definition 6.13 (Conjugate Prior)**.** A prior is *conjugate* for the likelihood function if the posterior is of the same form/type as the prior.

Conjugacy is particularly convenient because we can algebraically calculate our posterior distribution by updating the parameters of the prior distribution.

Remark. When considering the geometry of probability distributions, conjugate priors retain the same distance structure as the likelihood (Agarwal and Daumé III, 2010). $\diamondsuit$

To introduce a concrete example of conjugate priors, we describe in Example 6.11 the Binomial distribution (defined on discrete random variables) and the Beta distribution (defined on continuous random variables).

Example 6.11 (Beta-Binomial Conjugacy)
Consider a Binomial random variable $x \sim \text{Bin}(N, \mu)$ where

$$p(x \mid N, \mu) = \binom{N}{x} \mu^x (1-\mu)^{N-x}, \quad x = 0, 1, \ldots, N, \tag{6.102}$$

is the probability of finding x times the outcome "heads" in N coin flips, where μ is the probability of a "head". We place a Beta prior on the parameter μ, that is, $\mu \sim \text{Beta}(\alpha, \beta)$, where

$$p(\mu \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \mu^{\alpha-1}(1-\mu)^{\beta-1}. \tag{6.103}$$

If we now observe some outcome $x = h$, that is, we see h heads in N coin flips, we compute the posterior distribution on μ as

$$\begin{aligned}
p(\mu \mid x = h, N, \alpha, \beta) &\propto p(x \mid N, \mu) p(\mu \mid \alpha, \beta) && \text{(6.104a)}\\
&\propto \mu^h (1-\mu)^{(N-h)} \mu^{\alpha-1}(1-\mu)^{\beta-1} && \text{(6.104b)}\\
&= \mu^{h+\alpha-1}(1-\mu)^{(N-h)+\beta-1} && \text{(6.104c)}\\
&\propto \text{Beta}(h+\alpha, N-h+\beta), && \text{(6.104d)}
\end{aligned}$$

i.e., the posterior distribution is a Beta distribution as the prior, i.e., the

Likelihood	Conjugate prior	Posterior
Bernoulli	Beta	Beta
Binomial	Beta	Beta
Gaussian	Gaussian/inverse Gamma	Gaussian/inverse Gamma
Gaussian	Gaussian/inverse Wishart	Gaussian/inverse Wishart
Multinomial	Dirichlet	Dirichlet

Table 6.2 Examples of conjugate priors for common likelihood functions.

Beta prior is conjugate for the parameter μ in the Binomial likelihood function.

In the following example, we will derive a result that is similar to the Beta-Binomial conjugacy result. Here we will show that the Beta distribution is a conjugate prior for the Bernoulli distribution.

Example 6.12 (Beta-Bernoulli Conjugacy)
Let $x \in \{0, 1\}$ be distributed according to the Bernoulli distribution with parameter $\theta \in [0, 1]$, that is, $p(x = 1 \mid \theta) = \theta$. This can also be expressed as $p(x \mid \theta) = \theta^x(1-\theta)^{1-x}$. Let θ be distributed according to a Beta distribution with parameters α, β, that is, $p(\theta \mid \alpha, \beta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$.

Multiplying the Beta and the Bernoulli distributions, we get

$$
\begin{aligned}
p(\theta \mid x, \alpha, \beta) &= p(x \mid \theta)p(\theta \mid \alpha, \beta) && \text{(6.105a)} \\
&\propto \theta^x(1-\theta)^{1-x}\theta^{\alpha-1}(1-\theta)^{\beta-1} && \text{(6.105b)} \\
&= \theta^{\alpha+x-1}(1-\theta)^{\beta+(1-x)-1} && \text{(6.105c)} \\
&\propto p(\theta \mid \alpha + x, \beta + (1-x)) \,. && \text{(6.105d)}
\end{aligned}
$$

The last line is the Beta distribution with parameters $(\alpha + x, \beta + (1-x))$.

Table 6.2 lists examples for conjugate priors for the parameters of some standard likelihoods used in probabilistic modeling. Distributions such as Multinomial, inverse Gamma, inverse Wishart, and Dirichlet can be found in any statistical text, and are described in Bishop (2006), for example.

The Beta distribution is the conjugate prior for the parameter μ in both the Binomial and the Bernoulli likelihood. For a Gaussian likelihood function, we can place a conjugate Gaussian prior on the mean. The reason why the Gaussian likelihood appears twice in the table is that we need to distinguish the univariate from the multivariate case. In the univariate (scalar) case, the inverse Gamma is the conjugate prior for the variance. In the multivariate case, we use a conjugate inverse Wishart distribution as a prior on the covariance matrix. The Dirichlet distribution is the conjugate prior for the multinomial likelihood function. For further details, we refer to Bishop (2006).

The Gamma prior is conjugate for the precision (inverse variance) in the univariate Gaussian likelihood, and the Wishart prior is conjugate for the precision matrix (inverse covariance matrix) in the multivariate Gaussian likelihood.

6.6.2 Sufficient Statistics

Recall that a statistic of a random variable is a deterministic function of that random variable. For example, if $\boldsymbol{x} = [x_1, \dots, x_N]^\top$ is a vector of univariate Gaussian random variables, that is, $x_n \sim \mathcal{N}(\mu, \sigma^2)$, then the sample mean $\hat{\mu} = \frac{1}{N}(x_1 + \cdots + x_N)$ is a statistic. Sir Ronald Fisher discovered the notion of *sufficient statistics*: the idea that there are statistics that will contain all available information that can be inferred from data corresponding to the distribution under consideration. In other words, sufficient statistics carry all the information needed to make inference about the population, that is, they are the statistics that are sufficient to represent the distribution.

sufficient statistics

For a set of distributions parametrized by θ, let X be a random variable with distribution $p(x \mid \theta_0)$ given an unknown θ_0. A vector $\phi(x)$ of statistics is called sufficient statistics for θ_0 if they contain all possible information about θ_0. To be more formal about "contain all possible information", this means that the probability of x given θ can be factored into a part that does not depend on θ, and a part that depends on θ only via $\phi(x)$. The Fisher-Neyman factorization theorem formalizes this notion, which we state in Theorem 6.14 without proof.

Theorem 6.14 (Fisher-Neyman)**.** *[Theorem 6.5 in Lehmann and Casella (1998)] Let X have probability density function $p(x \mid \theta)$. Then the statistics $\phi(x)$ are sufficient for θ if and only if $p(x \mid \theta)$ can be written in the form*

Fisher-Neyman theorem

$$p(x \mid \theta) = h(x) g_\theta(\phi(x)) , \tag{6.106}$$

where $h(x)$ is a distribution independent of θ and g_θ captures all the dependence on θ via sufficient statistics $\phi(x)$.

If $p(x \mid \theta)$ does not depend on θ, then $\phi(x)$ is trivially a sufficient statistic for any function ϕ. The more interesting case is that $p(x \mid \theta)$ is dependent only on $\phi(x)$ and not x itself. In this case, $\phi(x)$ is a sufficient statistic for θ.

In machine learning, we consider a finite number of samples from a distribution. One could imagine that for simple distributions (such as the Bernoulli in Example 6.8) we only need a small number of samples to estimate the parameters of the distributions. We could also consider the opposite problem: If we have a set of data (a sample from an unknown distribution), which distribution gives the best fit? A natural question to ask is, as we observe more data, do we need more parameters θ to describe the distribution? It turns out that the answer is yes in general, and this is studied in non-parametric statistics (Wasserman, 2007). A converse question is to consider which class of distributions have finite-dimensional sufficient statistics, that is the number of parameters needed to describe them does not increase arbitrarily. The answer is exponential family distributions, described in the following section.

6.6.3 Exponential Family

There are three possible levels of abstraction we can have when considering distributions (of discrete or continuous random variables). At level one (the most concrete end of the spectrum), we have a particular named distribution with fixed parameters, for example a univariate Gaussian $\mathcal{N}(0,\,1)$ with zero mean and unit variance. In machine learning, we often use the second level of abstraction, that is, we fix the parametric form (the univariate Gaussian) and infer the parameters from data. For example, we assume a univariate Gaussian $\mathcal{N}(\mu,\,\sigma^2)$ with unknown mean μ and unknown variance σ^2, and use a maximum likelihood fit to determine the best parameters (μ, σ^2). We will see an example of this when considering linear regression in Chapter 9. A third level of abstraction is to consider families of distributions, and in this book, we consider the exponential family. The univariate Gaussian is an example of a member of the exponential family. Many of the widely used statistical models, including all the "named" models in Table 6.2, are members of the exponential family. They can all be unified into one concept (Brown, 1986).

Remark. A brief historical anecdote: Like many concepts in mathematics and science, exponential families were independently discovered at the same time by different researchers. In the years 1935–1936, Edwin Pitman in Tasmania, Georges Darmois in Paris, and Bernard Koopman in New York independently showed that the exponential families are the only families that enjoy finite-dimensional sufficient statistics under repeated independent sampling (Lehmann and Casella, 1998). $\diamondsuit$

An *exponential family* is a family of probability distributions, parameterized by $\boldsymbol{\theta} \in \mathbb{R}^D$, of the form

exponential family

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = h(\boldsymbol{x}) \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{\phi}(\boldsymbol{x})\rangle - A(\boldsymbol{\theta})\right) , \tag{6.107}$$

where $\boldsymbol{\phi}(\boldsymbol{x})$ is the vector of sufficient statistics. In general, any inner product (Section 3.2) can be used in (6.107), and for concreteness we will use the standard dot product here ($\langle \boldsymbol{\theta}, \boldsymbol{\phi}(\boldsymbol{x})\rangle = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x})$). Note that the form of the exponential family is essentially a particular expression of $g_\theta(\phi(x))$ in the Fisher-Neyman theorem (Theorem 6.14).

The factor $h(\boldsymbol{x})$ can be absorbed into the dot product term by adding another entry ($\log h(\boldsymbol{x})$) to the vector of sufficient statistics $\boldsymbol{\phi}(\boldsymbol{x})$, and constraining the corresponding parameter $\theta_0 = 1$. The term $A(\boldsymbol{\theta})$ is the normalization constant that ensures that the distribution sums up or integrates to one and is called the *log-partition function*. A good intuitive notion of exponential families can be obtained by ignoring these two terms and considering exponential families as distributions of the form

log-partition function

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) \propto \exp\left(\boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x})\right) . \tag{6.108}$$

For this form of parametrization, the parameters $\boldsymbol{\theta}$ are called the *natural*

natural parameters

parameters. At first glance, it seems that exponential families are a mundane transformation by adding the exponential function to the result of a dot product. However, there are many implications that allow for convenient modeling and efficient computation based on the fact that we can capture information about data in $\boldsymbol{\phi}(\boldsymbol{x})$.

Example 6.13 (Gaussian as Exponential Family)

Consider the univariate Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Let $\boldsymbol{\phi}(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$. Then by using the definition of the exponential family,

$$p(x \mid \boldsymbol{\theta}) \propto \exp(\theta_1 x + \theta_2 x^2)\,. \tag{6.109}$$

Setting

$$\boldsymbol{\theta} = \left[\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}\right]^\top \tag{6.110}$$

and substituting into (6.109), we obtain

$$p(x \mid \boldsymbol{\theta}) \propto \exp\left(\frac{\mu x}{\sigma^2} - \frac{x^2}{2\sigma^2}\right) \propto \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)\,. \tag{6.111}$$

Therefore, the univariate Gaussian distribution is a member of the exponential family with sufficient statistic $\boldsymbol{\phi}(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$, and natural parameters given by $\boldsymbol{\theta}$ in (6.110).

Example 6.14 (Bernoulli as Exponential Family)

Recall the Bernoulli distribution from Example 6.8

$$p(x \mid \mu) = \mu^x (1-\mu)^{1-x}\,, \quad x \in \{0, 1\}. \tag{6.112}$$

This can be written in exponential family form

$$p(x \mid \mu) = \exp\left[\log\left(\mu^x (1-\mu)^{1-x}\right)\right] \tag{6.113a}$$
$$= \exp\left[x \log \mu + (1-x)\log(1-\mu)\right] \tag{6.113b}$$
$$= \exp\left[x \log \mu - x\log(1-\mu) + \log(1-\mu)\right] \tag{6.113c}$$
$$= \exp\left[x \log \tfrac{\mu}{1-\mu} + \log(1-\mu)\right]. \tag{6.113d}$$

The last line (6.113d) can be identified as being in exponential family form (6.107) by observing that

$$h(x) = 1 \tag{6.114}$$
$$\theta = \log \tfrac{\mu}{1-\mu} \tag{6.115}$$
$$\phi(x) = x \tag{6.116}$$

$$A(\theta) = -\log(1-\mu) = \log(1+\exp(\theta)). \tag{6.117}$$

The relationship between θ and μ is invertible so that

$$\mu = \frac{1}{1+\exp(-\theta)}. \tag{6.118}$$

The relation (6.118) is used to obtain the right equality of (6.117).

Remark. The relationship between the original Bernoulli parameter μ and the natural parameter θ is known as the *sigmoid* or logistic function. Observe that $\mu \in (0,1)$ but $\theta \in \mathbb{R}$, and therefore the sigmoid function squeezes a real value into the range $(0,1)$. This property is useful in machine learning, for example it is used in logistic regression (Bishop, 2006, section 4.3.2), as well as as a nonlinear activation functions in neural networks (Goodfellow et al., 2016, chapter 6). ♢

sigmoid

It is often not obvious how to find the parametric form of the conjugate distribution of a particular distribution (for example, those in Table 6.2). Exponential families provide a convenient way to find conjugate pairs of distributions. Consider the random variable X is a member of the exponential family (6.107):

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = h(\boldsymbol{x}) \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{\phi}(\boldsymbol{x})\rangle - A(\boldsymbol{\theta})\right). \tag{6.119}$$

Every member of the exponential family has a conjugate prior (Brown, 1986)

$$p(\boldsymbol{\theta} \mid \boldsymbol{\gamma}) = h_c(\boldsymbol{\theta}) \exp\left(\left\langle \begin{bmatrix}\gamma_1\\ \gamma_2\end{bmatrix}, \begin{bmatrix}\boldsymbol{\theta}\\ -A(\boldsymbol{\theta})\end{bmatrix}\right\rangle - A_c(\boldsymbol{\gamma})\right), \tag{6.120}$$

where $\boldsymbol{\gamma} = \begin{bmatrix}\gamma_1\\ \gamma_2\end{bmatrix}$ has dimension $\dim(\boldsymbol{\theta})+1$. The sufficient statistics of the conjugate prior are $\begin{bmatrix}\boldsymbol{\theta}\\ -A(\boldsymbol{\theta})\end{bmatrix}$. By using the knowledge of the general form of conjugate priors for exponential families, we can derive functional forms of conjugate priors corresponding to particular distributions.

Example 6.15

Recall the exponential family form of the Bernoulli distribution (6.113d)

$$p(x \mid \mu) = \exp\left[x \log \frac{\mu}{1-\mu} + \log(1-\mu)\right]. \tag{6.121}$$

The canonical conjugate prior has the form

$$p(\mu \mid \alpha, \beta) = \frac{\mu}{1-\mu} \exp\left[\alpha \log \frac{\mu}{1-\mu} + (\beta+\alpha)\log(1-\mu) - A_c(\boldsymbol{\gamma})\right], \tag{6.122}$$

where we defined $\boldsymbol{\gamma} := [\alpha, \beta + \alpha]^\top$ and $h_c(\mu) := \mu/(1-\mu)$. Equation (6.122) then simplifies to

$$p(\mu \mid \alpha, \beta) = \exp\left[(\alpha - 1)\log\mu + (\beta - 1)\log(1-\mu) - A_c(\alpha, \beta)\right] . \tag{6.123}$$

Putting this in non-exponential family form yields

$$p(\mu \mid \alpha, \beta) \propto \mu^{\alpha-1}(1-\mu)^{\beta-1} , \tag{6.124}$$

which we identify as the Beta distribution (6.98). In example 6.12, we assumed that the Beta distribution is the conjugate prior of the Bernoulli distribution and showed that it was indeed the conjugate prior. In this example, we derived the form of the Beta distribution by looking at the canonical conjugate prior of the Bernoulli distribution in exponential family form.

As mentioned in the previous section, the main motivation for exponential families is that they have finite-dimensional sufficient statistics. Additionally, conjugate distributions are easy to write down, and the conjugate distributions also come from an exponential family. From an inference perspective, maximum likelihood estimation behaves nicely because empirical estimates of sufficient statistics are optimal estimates of the population values of sufficient statistics (recall the mean and covariance of a Gaussian). From an optimization perspective, the log-likelihood function is concave, allowing for efficient optimization approaches to be applied (Chapter 7).

6.7 Change of Variables/Inverse Transform

It may seem that there are very many known distributions, but in reality the set of distributions for which we have names is quite limited. Therefore, it is often useful to understand how transformed random variables are distributed. For example, assuming that X is a random variable distributed according to the univariate normal distribution $\mathcal{N}(0, 1)$, what is the distribution of X^2? Another example, which is quite common in machine learning, is, given that X_1 and X_2 are univariate standard normal, what is the distribution of $\frac{1}{2}(X_1 + X_2)$?

One option to work out the distribution of $\frac{1}{2}(X_1 + X_2)$ is to calculate the mean and variance of X_1 and X_2 and then combine them. As we saw in Section 6.4.4, we can calculate the mean and variance of resulting random variables when we consider affine transformations of random variables. However, we may not be able to obtain the functional form of the distribution under transformations. Furthermore, we may be interested in nonlinear transformations of random variables for which closed-form expressions are not readily available.

Remark (Notation). In this section, we will be explicit about random variables and the values they take. Hence, recall that we use capital letters X, Y to denote random variables and small letters x, y to denote the values in the target space $\mathcal{T}$ that the random variables take. We will explicitly write pmfs of discrete random variables X as $P(X = x)$. For continuous random variables X (Section 6.2.2), the pdf is written as $f(x)$ and the cdf is written as $F_X(x)$. $\diamondsuit$

We will look at two approaches for obtaining distributions of transformations of random variables: a direct approach using the definition of a cumulative distribution function and a change-of-variable approach that uses the chain rule of calculus (Section 5.2.2). The change-of-variable approach is widely used because it provides a "recipe" for attempting to compute the resulting distribution due to a transformation. We will explain the techniques for univariate random variables, and will only briefly provide the results for the general case of multivariate random variables.

Moment generating functions can also be used to study transformations of random variables (Casella and Berger, 2002, chapter 2).

Transformations of discrete random variables can be understood directly. Suppose that there is a discrete random variable X with pmf $P(X = x)$ (Section 6.2.1), and an invertible function $U(x)$. Consider the transformed random variable $Y := U(X)$, with pmf $P(Y = y)$. Then

$$\begin{aligned} P(Y = y) &= P(U(X) = y) && \text{transformation of interest} && (6.125a) \\ &= P(X = U^{-1}(y)) && \text{inverse} && (6.125b) \end{aligned}$$

where we can observe that $x = U^{-1}(y)$. Therefore, for discrete random variables, transformations directly change the individual events (with the probabilities appropriately transformed).

6.7.1 Distribution Function Technique

The distribution function technique goes back to first principles, and uses the definition of a cdf $F_X(x) = P(X \leqslant x)$ and the fact that its differential is the pdf $f(x)$ (Wasserman, 2004, chapter 2). For a random variable X and a function U, we find the pdf of the random variable $Y := U(X)$ by

1. Finding the cdf:

$$F_Y(y) = P(Y \leqslant y) \tag{6.126}$$

2. Differentiating the cdf $F_Y(y)$ to get the pdf $f(y)$.

$$f(y) = \frac{\mathrm{d}}{\mathrm{d}y} F_Y(y)\,. \tag{6.127}$$

We also need to keep in mind that the domain of the random variable may have changed due to the transformation by U.

Example 6.16
Let X be a continuous random variable with probability density function on $0 \leqslant x \leqslant 1$

$$f(x) = 3x^2 \,. \tag{6.128}$$

We are interested in finding the pdf of $Y = X^2$.

The function f is an increasing function of x, and therefore the resulting value of y lies in the interval $[0, 1]$. We obtain

$$\begin{aligned}
F_Y(y) &= P(Y \leqslant y) && \text{definition of cdf} && (6.129a)\\
&= P(X^2 \leqslant y) && \text{transformation of interest} && (6.129b)\\
&= P(X \leqslant y^{\frac{1}{2}}) && \text{inverse} && (6.129c)\\
&= F_X(y^{\frac{1}{2}}) && \text{definition of cdf} && (6.129d)\\
&= \int_0^{y^{\frac{1}{2}}} 3t^2 \mathrm{d}t && \text{cdf as a definite integral} && (6.129e)\\
&= \left[t^3\right]_{t=0}^{t=y^{\frac{1}{2}}} && \text{result of integration} && (6.129f)\\
&= y^{\frac{3}{2}} \,, \quad 0 \leqslant y \leqslant 1 \,. && && (6.129g)
\end{aligned}$$

Therefore, the cdf of Y is

$$F_Y(y) = y^{\frac{3}{2}} \tag{6.130}$$

for $0 \leqslant y \leqslant 1$. To obtain the pdf, we differentiate the cdf

$$f(y) = \frac{\mathrm{d}}{\mathrm{d}y} F_Y(y) = \frac{3}{2} y^{\frac{1}{2}} \tag{6.131}$$

for $0 \leqslant y \leqslant 1$.

Functions that have inverses are called bijective functions (Section 2.7).

In Example 6.16, we considered a strictly monotonically increasing function $f(x) = 3x^2$. This means that we could compute an inverse function. In general, we require that the function of interest $y = U(x)$ has an inverse $x = U^{-1}(y)$. A useful result can be obtained by considering the cumulative distribution function $F_X(x)$ of a random variable X, and using it as the transformation $U(x)$. This leads to the following theorem.

Theorem 6.15. *[Theorem 2.1.10 in Casella and Berger (2002)] Let X be a continuous random variable with a strictly monotonic cumulative distribution function $F_X(x)$. Then the random variable Y defined as*

$$Y := F_X(X) \tag{6.132}$$

has a uniform distribution.

probability integral transform

Theorem 6.15 is known as the *probability integral transform*, and it is used to derive algorithms for sampling from distributions by transforming the result of sampling from a uniform random variable (Bishop, 2006). The algorithm works by first generating a sample from a uniform distribution, then transforming it by the inverse cdf (assuming this is available) to obtain a sample from the desired distribution. The probability integral transform is also used for hypothesis testing whether a sample comes from a particular distribution (Lehmann and Romano, 2005). The idea that the output of a cdf gives a uniform distribution also forms the basis of copulas (Nelsen, 2006).

6.7.2 Change of Variables

The distribution function technique in Section 6.7.1 is derived from first principles, based on the definitions of cdfs and using properties of inverses, differentiation, and integration. This argument from first principles relies on two facts:

1. We can transform the cdf of Y into an expression that is a cdf of X.
2. We can differentiate the cdf to obtain the pdf.

Let us break down the reasoning step by step, with the goal of understanding the more general change-of-variables approach in Theorem 6.16.

Change of variables in probability relies on the change-of-variables method in calculus (Tandra, 2014).

Remark. The name "change of variables" comes from the idea of changing the variable of integration when faced with a difficult integral. For univariate functions, we use the substitution rule of integration,

$$\int f(g(x))g'(x)\mathrm{d}x = \int f(u)\mathrm{d}u\,, \quad \text{where} \quad u = g(x)\,. \tag{6.133}$$

The derivation of this rule is based on the chain rule of calculus (5.32) and by applying twice the fundamental theorem of calculus. The fundamental theorem of calculus formalizes the fact that integration and differentiation are somehow "inverses" of each other. An intuitive understanding of the rule can be obtained by thinking (loosely) about small changes (differentials) to the equation $u = g(x)$, that is by considering $\Delta u = g'(x)\Delta x$ as a differential of $u = g(x)$. By substituting $u = g(x)$, the argument inside the integral on the right-hand side of (6.133) becomes $f(g(x))$. By pretending that the term $\mathrm{d}u$ can be approximated by $\mathrm{d}u \approx \Delta u = g'(x)\Delta x$, and that $\mathrm{d}x \approx \Delta x$, we obtain (6.133). $\diamondsuit$

Consider a univariate random variable X, and an *invertible* function U, which gives us another random variable $Y = U(X)$. We assume that random variable X has states $x \in [a, b]$. By the definition of the cdf, we have

$$F_Y(y) = P(Y \leqslant y)\,. \tag{6.134}$$

We are interested in a function U of the random variable

$$P(Y \leqslant y) = P(U(X) \leqslant y)\,, \tag{6.135}$$

where we assume that the function U is invertible. An invertible function on an interval is either strictly increasing or strictly decreasing. In the case that U is strictly increasing, then its inverse U^{-1} is also strictly increasing. By applying the inverse U^{-1} to the arguments of $P(U(X) \leqslant y)$, we obtain

$$P(U(X) \leqslant y) = P(U^{-1}(U(X)) \leqslant U^{-1}(y)) = P(X \leqslant U^{-1}(y)) . \tag{6.136}$$

The right-most term in (6.136) is an expression of the cdf of X. Recall the definition of the cdf in terms of the pdf

$$P(X \leqslant U^{-1}(y)) = \int_a^{U^{-1}(y)} f(x)\mathrm{d}x . \tag{6.137}$$

Now we have an expression of the cdf of Y in terms of x:

$$F_Y(y) = \int_a^{U^{-1}(y)} f(x)\mathrm{d}x . \tag{6.138}$$

To obtain the pdf, we differentiate (6.138) with respect to y:

$$f(y) = \frac{\mathrm{d}}{\mathrm{d}y} F_y(y) = \frac{\mathrm{d}}{\mathrm{d}y} \int_a^{U^{-1}(y)} f(x)\mathrm{d}x . \tag{6.139}$$

Note that the integral on the right-hand side is with respect to x, but we need an integral with respect to y because we are differentiating with respect to y. In particular, we use (6.133) to get the substitution

$$\int f(U^{-1}(y)) {U^{-1}}'(y)\mathrm{d}y = \int f(x)\mathrm{d}x \quad \text{where} \quad x = U^{-1}(y) . \tag{6.140}$$

Using (6.140) on the right-hand side of (6.139) gives us

$$f(y) = \frac{\mathrm{d}}{\mathrm{d}y} \int_a^{U^{-1}(y)} f_x(U^{-1}(y)) {U^{-1}}'(y)\mathrm{d}y . \tag{6.141}$$

We then recall that differentiation is a linear operator and we use the subscript x to remind ourselves that $f_x(U^{-1}(y))$ is a function of x and not y. Invoking the fundamental theorem of calculus again gives us

$$f(y) = f_x(U^{-1}(y)) \cdot \left(\frac{\mathrm{d}}{\mathrm{d}y} U^{-1}(y) \right) . \tag{6.142}$$

Recall that we assumed that U is a strictly increasing function. For decreasing functions, it turns out that we have a negative sign when we follow the same derivation. We introduce the absolute value of the differential to have the same expression for both increasing and decreasing U:

$$f(y) = f_x(U^{-1}(y)) \cdot \left| \frac{\mathrm{d}}{\mathrm{d}y} U^{-1}(y) \right| . \tag{6.143}$$

change-of-variable technique

This is called the *change-of-variable technique*. The term $\left| \frac{\mathrm{d}}{\mathrm{d}y} U^{-1}(y) \right|$ in (6.143) measures how much a unit volume changes when applying U (see also the definition of the Jacobian in Section 5.3).

Remark. In comparison to the discrete case in (6.125b), we have an additional factor $\left|\frac{\mathrm{d}}{\mathrm{d}y}U^{-1}(y)\right|$. The continuous case requires more care because $P(Y = y) = 0$ for all y. The probability density function $f(y)$ does not have a description as a probability of an event involving y. $\diamondsuit$

So far in this section, we have been studying univariate change of variables. The case for multivariate random variables is analogous, but complicated by fact that the absolute value cannot be used for multivariate functions. Instead, we use the determinant of the Jacobian matrix. Recall from (5.58) that the Jacobian is a matrix of partial derivatives, and that the existence of a nonzero determinant shows that we can invert the Jacobian. Recall the discussion in Section 4.1 that the determinant arises because our differentials (cubes of volume) are transformed into parallelepipeds by the Jacobian. Let us summarize preceding the discussion in the following theorem, which gives us a recipe for multivariate change of variables.

Theorem 6.16. *[Theorem 17.2 in Billingsley (1995)] Let $f(\boldsymbol{x})$ be the value of the probability density of the multivariate continuous random variable X. If the vector-valued function $\boldsymbol{y} = U(\boldsymbol{x})$ is differentiable and invertible for all values within the domain of $\boldsymbol{x}$, then for corresponding values of $\boldsymbol{y}$, the probability density of $Y = U(X)$ is given by*

$$f(\boldsymbol{y}) = f_{\boldsymbol{x}}(U^{-1}(\boldsymbol{y})) \cdot \left|\det\left(\frac{\partial}{\partial \boldsymbol{y}}U^{-1}(\boldsymbol{y})\right)\right|. \tag{6.144}$$

The theorem looks intimidating at first glance, but the key point is that a change of variable of a multivariate random variable follows the procedure of the univariate change of variable. First we need to work out the inverse transform, and substitute that into the density of $\boldsymbol{x}$. Then we calculate the determinant of the Jacobian and multiply the result. The following example illustrates the case of a bivariate random variable.

Example 6.17

Consider a bivariate random variable X with states $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ and probability density function

$$f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \frac{1}{2\pi}\exp\left(-\frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right). \tag{6.145}$$

We use the change-of-variable technique from Theorem 6.16 to derive the effect of a linear transformation (Section 2.7) of the random variable. Consider a matrix $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ defined as

$$\boldsymbol{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}. \tag{6.146}$$

We are interested in finding the probability density function of the transformed bivariate random variable Y with states $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$.

Recall that for change of variables we require the inverse transformation of $\boldsymbol{x}$ as a function of $\boldsymbol{y}$. Since we consider linear transformations, the inverse transformation is given by the matrix inverse (see Section 2.2.2). For 2×2 matrices, we can explicitly write out the formula, given by

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \boldsymbol{A}^{-1} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \tag{6.147}$$

Observe that $ad - bc$ is the determinant (Section 4.1) of $\boldsymbol{A}$. The corresponding probability density function is given by

$$f(\boldsymbol{x}) = f(\boldsymbol{A}^{-1}\boldsymbol{y}) = \frac{1}{2\pi} \exp\left(-\tfrac{1}{2}\boldsymbol{y}^\top \boldsymbol{A}^{-\top}\boldsymbol{A}^{-1}\boldsymbol{y}\right). \tag{6.148}$$

The partial derivative of a matrix times a vector with respect to the vector is the matrix itself (Section 5.5), and therefore

$$\frac{\partial}{\partial \boldsymbol{y}} \boldsymbol{A}^{-1}\boldsymbol{y} = \boldsymbol{A}^{-1}. \tag{6.149}$$

Recall from Section 4.1 that the determinant of the inverse is the inverse of the determinant so that the determinant of the Jacobian matrix is

$$\det\left(\frac{\partial}{\partial \boldsymbol{y}} \boldsymbol{A}^{-1}\boldsymbol{y}\right) = \frac{1}{ad - bc}. \tag{6.150}$$

We are now able to apply the change-of-variable formula from Theorem 6.16 by multiplying (6.148) with (6.150), which yields

$$f(\boldsymbol{y}) = f(\boldsymbol{x}) \left|\det\left(\frac{\partial}{\partial \boldsymbol{y}} \boldsymbol{A}^{-1}\boldsymbol{y}\right)\right| \tag{6.151a}$$

$$= \frac{1}{2\pi} \exp\left(-\tfrac{1}{2}\boldsymbol{y}^\top \boldsymbol{A}^{-\top}\boldsymbol{A}^{-1}\boldsymbol{y}\right) |ad - bc|^{-1}. \tag{6.151b}$$

While Example 6.17 is based on a bivariate random variable, which allows us to easily compute the matrix inverse, the preceding relation holds for higher dimensions.

Remark. We saw in Section 6.5 that the density $f(\boldsymbol{x})$ in (6.148) is actually the standard Gaussian distribution, and the transformed density $f(\boldsymbol{y})$ is a bivariate Gaussian with covariance $\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^\top$. ♢

We will use the ideas in this chapter to describe probabilistic modeling in Section 8.4, as well as introduce a graphical language in Section 8.5. We will see direct machine learning applications of these ideas in Chapters 9 and 11.

6.8 Further Reading

This chapter is rather terse at times. Grinstead and Snell (1997) and Walpole et al. (2011) provide more relaxed presentations that are suitable for self-study. Readers interested in more philosophical aspects of probability should consider Hacking (2001), whereas an approach that is more related to software engineering is presented by Downey (2014). An overview of exponential families can be found in Barndorff-Nielsen (2014). We will see more about how to use probability distributions to model machine learning tasks in Chapter 8. Ironically, the recent surge in interest in neural networks has resulted in a broader appreciation of probabilistic models. For example, the idea of normalizing flows (Jimenez Rezende and Mohamed, 2015) relies on change of variables for transforming random variables. An overview of methods for variational inference as applied to neural networks is described in chapters 16 to 20 of the book by Goodfellow et al. (2016).

We side stepped a large part of the difficulty in continuous random variables by avoiding measure theoretic questions (Billingsley, 1995; Pollard, 2002), and by assuming without construction that we have real numbers, and ways of defining sets on real numbers as well as their appropriate frequency of occurrence. These details do matter, for example, in the specification of conditional probability $p(y \mid x)$ for continuous random variables x, y (Proschan and Presnell, 1998). The lazy notation hides the fact that we want to specify that $X = x$ (which is a set of measure zero). Furthermore, we are interested in the probability density function of y. A more precise notation would have to say $\mathbb{E}_y[f(y) \mid \sigma(x)]$, where we take the expectation over y of a test function f conditioned on the σ-algebra of x. A more technical audience interested in the details of probability theory have many options (Jaynes, 2003; MacKay, 2003; Jacod and Protter, 2004; Grimmett and Welsh, 2014), including some very technical discussions (Shiryayev, 1984; Lehmann and Casella, 1998; Dudley, 2002; Bickel and Doksum, 2006; Çinlar, 2011). An alternative way to approach probability is to start with the concept of expectation, and "work backward" to derive the necessary properties of a probability space (Whittle, 2000). As machine learning allows us to model more intricate distributions on ever more complex types of data, a developer of probabilistic machine learning models would have to understand these more technical aspects. Machine learning texts with a probabilistic modeling focus include the books by MacKay (2003); Bishop (2006); Rasmussen and Williams (2006); Barber (2012); Murphy (2012).

Exercises

6.1 Consider the following bivariate distribution $p(x, y)$ of two discrete random variables X and Y.

Y	y_1	0.01	0.02	0.03	0.1	0.1
	y_2	0.05	0.1	0.05	0.07	0.2
	y_3	0.1	0.05	0.03	0.05	0.04
		x_1	x_2	x_3	x_4	x_5
				X		

Compute:

a. The marginal distributions $p(x)$ and $p(y)$.
b. The conditional distributions $p(x|Y = y_1)$ and $p(y|X = x_3)$.

6.2 Consider a mixture of two Gaussian distributions (illustrated in Figure 6.2),

$$0.4\,\mathcal{N}\left(\begin{bmatrix}10\\2\end{bmatrix}, \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right) + 0.6\,\mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}8.4 & 2.0\\2.0 & 1.7\end{bmatrix}\right).$$

a. Compute the marginal distributions for each dimension.
b. Compute the mean, mode and median for each marginal distribution.
c. Compute the mean and mode for the two-dimensional distribution.

6.3 You have written a computer program that sometimes compiles and sometimes not (code does not change). You decide to model the apparent stochasticity (success vs. no success) x of the compiler using a Bernoulli distribution with parameter μ:

$$p(x \mid \mu) = \mu^x (1-\mu)^{1-x}, \quad x \in \{0, 1\}.$$

Choose a conjugate prior for the Bernoulli likelihood and compute the posterior distribution $p(\mu \mid x_1, \dots, x_N)$.

6.4 There are two bags. The first bag contains four mangos and two apples; the second bag contains four mangos and four apples.
We also have a biased coin, which shows "heads" with probability 0.6 and "tails" with probability 0.4. If the coin shows "heads". we pick a fruit at random from bag 1; otherwise we pick a fruit at random from bag 2.
Your friend flips the coin (you cannot see the result), picks a fruit at random from the corresponding bag, and presents you a mango.
What is the probability that the mango was picked from bag 2?
Hint: Use Bayes' theorem.

6.5 Consider the time-series model

$$\begin{aligned}\boldsymbol{x}_{t+1} &= \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{w}, \quad \boldsymbol{w} \sim \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{Q}\big)\\ \boldsymbol{y}_t &= \boldsymbol{C}\boldsymbol{x}_t + \boldsymbol{v}, \quad \boldsymbol{v} \sim \mathcal{N}\big(\boldsymbol{0}, \boldsymbol{R}\big),\end{aligned}$$

where $\boldsymbol{w}, \boldsymbol{v}$ are i.i.d. Gaussian noise variables. Further, assume that $p(\boldsymbol{x}_0) = \mathcal{N}\big(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\big)$.

a. What is the form of $p(\boldsymbol{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_T)$? Justify your answer (you do not have to explicitly compute the joint distribution).
b. Assume that $p(\boldsymbol{x}_t \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_t) = \mathcal{N}\big(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t\big)$.
    1. Compute $p(\boldsymbol{x}_{t+1} \mid \boldsymbol{y}_1, \dots, \boldsymbol{y}_t)$.

2. Compute $p(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{t+1} \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$.
3. At time $t+1$, we observe the value $\boldsymbol{y}_{t+1} = \hat{\boldsymbol{y}}$. Compute the conditional distribution $p(\boldsymbol{x}_{t+1} \mid \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t+1})$.

6.6 Prove the relationship in (6.44), which relates the standard definition of the variance to the raw-score expression for the variance.

6.7 Prove the relationship in (6.45), which relates the pairwise difference between examples in a dataset with the raw-score expression for the variance.

6.8 Express the Bernoulli distribution in the natural parameter form of the exponential family, see (6.107).

6.9 Express the Binomial distribution as an exponential family distribution. Also express the Beta distribution is an exponential family distribution. Show that the product of the Beta and the Binomial distribution is also a member of the exponential family.

6.10 Derive the relationship in Section 6.5.2 in two ways:

a. By completing the square
b. By expressing the Gaussian in its exponential family form

The *product* of two Gaussians $\mathcal{N}\big(\boldsymbol{x} \mid \boldsymbol{a},\, \boldsymbol{A}\big)\mathcal{N}\big(\boldsymbol{x} \mid \boldsymbol{b},\, \boldsymbol{B}\big)$ is an unnormalized Gaussian distribution $c\mathcal{N}\big(\boldsymbol{x} \mid \boldsymbol{c},\, \boldsymbol{C}\big)$ with

$$
\begin{aligned}
\boldsymbol{C} &= (\boldsymbol{A}^{-1} + \boldsymbol{B}^{-1})^{-1} \\
\boldsymbol{c} &= \boldsymbol{C}(\boldsymbol{A}^{-1}\boldsymbol{a} + \boldsymbol{B}^{-1}\boldsymbol{b}) \\
c &= (2\pi)^{-\frac{D}{2}} \mid \boldsymbol{A} + \boldsymbol{B} \mid^{-\frac{1}{2}} \exp\big(-\tfrac{1}{2}(\boldsymbol{a} - \boldsymbol{b})^\top (\boldsymbol{A} + \boldsymbol{B})^{-1} (\boldsymbol{a} - \boldsymbol{b})\big) \,.
\end{aligned}
$$

Note that the normalizing constant c itself can be considered a (normalized) Gaussian distribution either in $\boldsymbol{a}$ or in $\boldsymbol{b}$ with an "inflated" covariance matrix $\boldsymbol{A} + \boldsymbol{B}$, i.e., $c = \mathcal{N}\big(\boldsymbol{a} \mid \boldsymbol{b},\, \boldsymbol{A} + \boldsymbol{B}\big) = \mathcal{N}\big(\boldsymbol{b} \mid \boldsymbol{a},\, \boldsymbol{A} + \boldsymbol{B}\big)$.

6.11 **Iterated Expectations.**
Consider two random variables x, y with joint distribution $p(x, y)$. Show that

$$
\mathbb{E}_X[x] = \mathbb{E}_Y\big[\mathbb{E}_X[x \mid y]\big] \,.
$$

Here, $\mathbb{E}_X[x \mid y]$ denotes the expected value of x under the conditional distribution $p(x \mid y)$.

6.12 **Manipulation of Gaussian Random Variables.**
Consider a Gaussian random variable $\boldsymbol{x} \sim \mathcal{N}\big(\boldsymbol{x} \mid \boldsymbol{\mu}_x,\, \boldsymbol{\Sigma}_x\big)$, where $\boldsymbol{x} \in \mathbb{R}^D$. Furthermore, we have

$$
\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} + \boldsymbol{w} \,,
$$

where $\boldsymbol{y} \in \mathbb{R}^E$, $\boldsymbol{A} \in \mathbb{R}^{E \times D}$, $\boldsymbol{b} \in \mathbb{R}^E$, and $\boldsymbol{w} \sim \mathcal{N}\big(\boldsymbol{w} \mid \boldsymbol{0},\, \boldsymbol{Q}\big)$ is independent Gaussian noise. "Independent" implies that $\boldsymbol{x}$ and $\boldsymbol{w}$ are independent random variables and that $\boldsymbol{Q}$ is diagonal.

a. Write down the likelihood $p(\boldsymbol{y} \mid \boldsymbol{x})$.
b. The distribution $p(\boldsymbol{y}) = \int p(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}$ is Gaussian. Compute the mean $\boldsymbol{\mu}_y$ and the covariance $\boldsymbol{\Sigma}_y$. Derive your result in detail.
c. The random variable $\boldsymbol{y}$ is being transformed according to the measurement mapping

$$
\boldsymbol{z} = \boldsymbol{C}\boldsymbol{y} + \boldsymbol{v} \,,
$$

where $\boldsymbol{z} \in \mathbb{R}^F$, $\boldsymbol{C} \in \mathbb{R}^{F \times E}$, and $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{v} \mid \boldsymbol{0}, \boldsymbol{R})$ is independent Gaussian (measurement) noise.

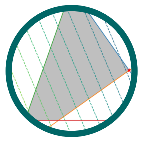
- Write down $p(\boldsymbol{z} \mid \boldsymbol{y})$.
- Compute $p(\boldsymbol{z})$, i.e., the mean $\boldsymbol{\mu}_z$ and the covariance $\boldsymbol{\Sigma}_z$. Derive your result in detail.

d. Now, a value $\hat{\boldsymbol{y}}$ is measured. Compute the posterior distribution $p(\boldsymbol{x} \mid \hat{\boldsymbol{y}})$. *Hint for solution:* This posterior is also Gaussian, i.e., we need to determine only its mean and covariance matrix. Start by explicitly computing the joint Gaussian $p(\boldsymbol{x}, \boldsymbol{y})$. This also requires us to compute the cross-covariances $\mathrm{Cov}_{\boldsymbol{x},\boldsymbol{y}}[\boldsymbol{x}, \boldsymbol{y}]$ and $\mathrm{Cov}_{\boldsymbol{y},\boldsymbol{x}}[\boldsymbol{y}, \boldsymbol{x}]$. Then apply the rules for Gaussian conditioning.

6.13 **Probability Integral Transformation**
Given a continuous random variable X, with cdf $F_X(x)$, show that the random variable $Y := F_X(X)$ is uniformly distributed (Theorem 6.15).

7

Continuous Optimization



Since machine learning algorithms are implemented on a computer, the mathematical formulations are expressed as numerical optimization methods. This chapter describes the basic numerical methods for training machine learning models. Training a machine learning model often boils down to finding a good set of parameters. The notion of "good" is determined by the objective function or the probabilistic model, which we will see examples of in the second part of this book. Given an objective function, finding the best value is done using optimization algorithms.

Since we consider data and models in $\mathbb{R}^D$, the optimization problems we face are *continuous* optimization problems, as opposed to *combinatorial* optimization problems for discrete variables.

This chapter covers two main branches of continuous optimization (Figure 7.2): unconstrained and constrained optimization. We will assume in this chapter that our objective function is differentiable (see Chapter 5), hence we have access to a gradient at each location in the space to help us find the optimum value. By convention, most objective functions in machine learning are intended to be minimized, that is, the best value is the minimum value. Intuitively finding the best value is like finding the valleys of the objective function, and the gradients point us uphill. The idea is to move downhill (opposite to the gradient) and hope to find the deepest point. For unconstrained optimization, this is the only concept we need, but there are several design choices, which we discuss in Section 7.1. For constrained optimization, we need to introduce other concepts to manage the constraints (Section 7.2). We will also introduce a special class of problems (convex optimization problems in Section 7.3) where we can make statements about reaching the global optimum.

global minimum

Consider the function in Figure 7.2. The function has a *global minimum* around $x = -4.5$, with a function value of approximately -47. Since the function is "smooth," the gradients can be used to help find the minimum by indicating whether we should take a step to the right or left. This assumes that we are in the correct bowl, as there exists another *local minimum* around $x = 0.7$. Recall that we can solve for all the stationary points of a function by calculating its derivative and setting it to zero. For

local minimum

Stationary points are the real roots of the derivative, that is, points that have zero gradient.

$$\ell(x) = x^4 + 7x^3 + 5x^2 - 17x + 3\,, \tag{7.1}$$

we obtain the corresponding gradient as

$$\frac{\mathrm{d}\ell(x)}{\mathrm{d}x} = 4x^3 + 21x^2 + 10x - 17\,. \tag{7.2}$$

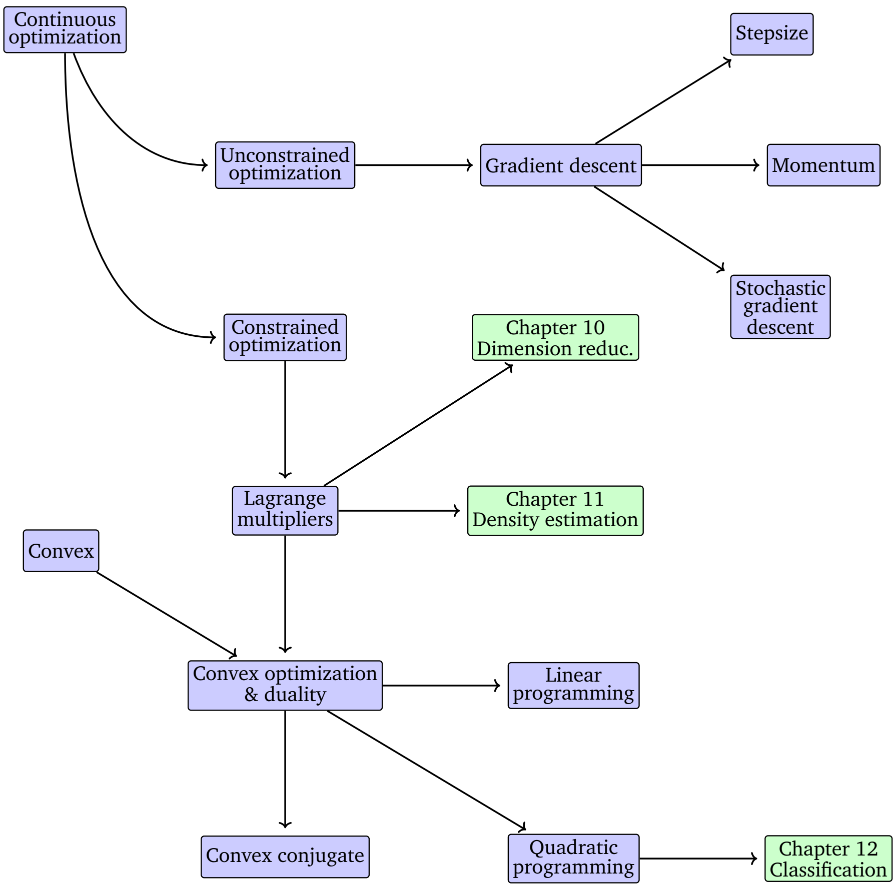


Figure 7.2 A mind map of the concepts related to optimization, as presented in this chapter. There are two main ideas: gradient descent and convex optimization.

Since this is a cubic equation, it has in general three solutions when set to zero. In the example, two of them are minimums and one is a maximum (around $x = -1.4$). To check whether a stationary point is a minimum or maximum, we need to take the derivative a second time and check whether the second derivative is positive or negative at the stationary point. In our case, the second derivative is

$$\frac{\mathrm{d}^2\ell(x)}{\mathrm{d}x^2} = 12x^2 + 42x + 10\,. \tag{7.3}$$

By substituting our visually estimated values of $x = -4.5, -1.4, 0.7$, we will observe that as expected the middle point is a maximum $\left(\frac{\mathrm{d}^2\ell(x)}{\mathrm{d}x^2} < 0\right)$ and the other two stationary points are minimums.

Note that we have avoided analytically solving for values of x in the previous discussion, although for low-order polynomials such as the preceding we could do so. In general, we are unable to find analytic solutions, and hence we need to start at some value, say $x_0 = -6$, and follow the negative gradient. The negative gradient indicates that we should go

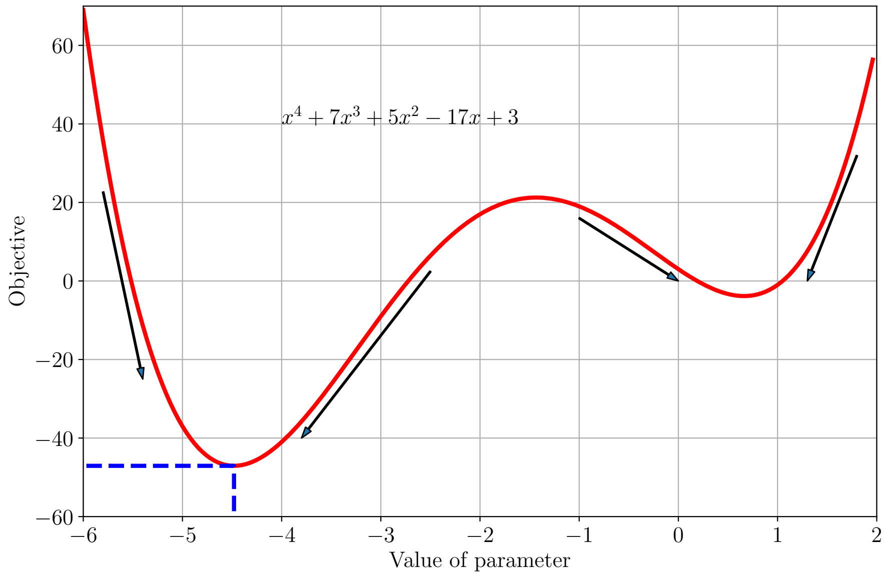


Figure 7.2 Example objective function. Negative gradients are indicated by arrows, and the global minimum is indicated by the dashed blue line.

right, but not how far (this is called the step-size). Furthermore, if we had started at the right side (e.g., $x_0 = 0$) the negative gradient would have led us to the wrong minimum. Figure 7.2 illustrates the fact that for $x > -1$, the negative gradient points toward the minimum on the right of the figure, which has a larger objective value.

According to the Abel–Ruffini theorem, there is in general no algebraic solution for polynomials of degree 5 or more (Abel, 1826).

In Section 7.3, we will learn about a class of functions, called convex functions, that do not exhibit this tricky dependency on the starting point of the optimization algorithm. For convex functions, all local minimums are global minimum. It turns out that many machine learning objective functions are designed such that they are convex, and we will see an example in Chapter 12.

For convex functions all local minima are global minimum.

The discussion in this chapter so far was about a one-dimensional function, where we are able to visualize the ideas of gradients, descent directions, and optimal values. In the rest of this chapter we develop the same ideas in high dimensions. Unfortunately, we can only visualize the concepts in one dimension, but some concepts do not generalize directly to higher dimensions, therefore some care needs to be taken when reading.

7.1 Optimization Using Gradient Descent

We now consider the problem of solving for the minimum of a real-valued function

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}), \tag{7.4}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is an objective function that captures the machine learning problem at hand. We assume that our function f is differentiable, and we are unable to analytically find a solution in closed form.

Gradient descent is a first-order optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient of the function at the current point. Recall from Section 5.1 that the gradient points in the direction of the steepest ascent. Another useful intuition is to consider the set of lines where the function is at a certain value ($f(\boldsymbol{x}) = c$ for some value $c \in \mathbb{R}$), which are known as the contour lines. The gradient points in a direction that is orthogonal to the contour lines of the function we wish to optimize.

We use the convention of row vectors for gradients.

Let us consider multivariate functions. Imagine a surface (described by the function $f(\boldsymbol{x})$) with a ball starting at a particular location $\boldsymbol{x}_0$. When the ball is released, it will move downhill in the direction of steepest descent. Gradient descent exploits the fact that $f(\boldsymbol{x}_0)$ decreases fastest if one moves from $\boldsymbol{x}_0$ in the direction of the negative gradient $-((\nabla f)(\boldsymbol{x}_0))^\top$ of f at $\boldsymbol{x}_0$. We assume in this book that the functions are differentiable, and refer the reader to more general settings in Section 7.4. Then, if

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - \gamma((\nabla f)(\boldsymbol{x}_0))^\top \tag{7.5}$$

for a small *step-size* $\gamma \geqslant 0$, then $f(\boldsymbol{x}_1) \leqslant f(\boldsymbol{x}_0)$. Note that we use the transpose for the gradient since otherwise the dimensions will not work out.

This observation allows us to define a simple gradient descent algorithm: If we want to find a local optimum $f(\boldsymbol{x}_*)$ of a function $f : \mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{x} \mapsto f(\boldsymbol{x})$, we start with an initial guess $\boldsymbol{x}_0$ of the parameters we wish to optimize and then iterate according to

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \gamma_i((\nabla f)(\boldsymbol{x}_i))^\top . \tag{7.6}$$

For suitable step-size γ_i, the sequence $f(\boldsymbol{x}_0) \geqslant f(\boldsymbol{x}_1) \geqslant \ldots$ converges to a local minimum.

Example 7.1

Consider a quadratic function in two dimensions

$$f\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right) = \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^\top \begin{bmatrix}2 & 1\\1 & 20\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} - \begin{bmatrix}5\\3\end{bmatrix}^\top \begin{bmatrix}x_1\\x_2\end{bmatrix} \tag{7.7}$$

with gradient

$$\nabla f\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right) = \begin{bmatrix}x_1\\x_2\end{bmatrix}^\top \begin{bmatrix}2 & 1\\1 & 20\end{bmatrix} - \begin{bmatrix}5\\3\end{bmatrix}^\top . \tag{7.8}$$

Starting at the initial location $\boldsymbol{x}_0 = [-3, -1]^\top$, we iteratively apply (7.6) to obtain a sequence of estimates that converge to the minimum value

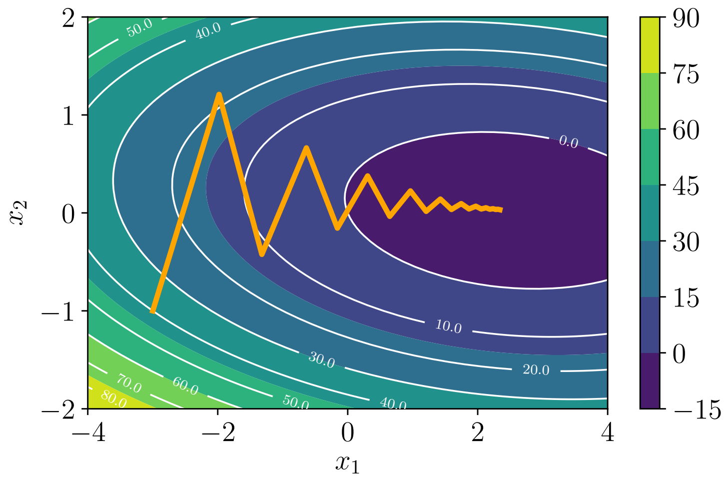


Figure 7.2 Gradient descent on a two-dimensional quadratic surface (shown as a heatmap). See Example 7.1 for a description.

(illustrated in Figure 7.2). We can see (both from the figure and by plugging $\boldsymbol{x}_0$ into (7.8) with $\gamma = 0.085$) that the negative gradient at $\boldsymbol{x}_0$ points north and east, leading to $\boldsymbol{x}_1 = [-1.98, 1.21]^\top$. Repeating that argument gives us $\boldsymbol{x}_2 = [-1.32, -0.42]^\top$, and so on.

Remark. Gradient descent can be relatively slow close to the minimum: Its asymptotic rate of convergence is inferior to many other methods. Using the ball rolling down the hill analogy, when the surface is a long, thin valley, the problem is poorly conditioned (Trefethen and Bau III, 1997). For poorly conditioned convex problems, gradient descent increasingly "zigzags" as the gradients point nearly orthogonally to the shortest direction to a minimum point; see Figure 7.2. $\diamondsuit$

7.1.1 Step-size

As mentioned earlier, choosing a good step-size is important in gradient descent. If the step-size is too small, gradient descent can be slow. If the step-size is chosen too large, gradient descent can overshoot, fail to converge, or even diverge. We will discuss the use of momentum in the next section. It is a method that smoothes out erratic behavior of gradient updates and dampens oscillations.

The step-size is also called the learning rate.

Adaptive gradient methods rescale the step-size at each iteration, depending on local properties of the function. There are two simple heuristics (Toussaint, 2012):

- When the function value increases after a gradient step, the step-size was too large. Undo the step and decrease the step-size.
- When the function value decreases the step could have been larger. Try to increase the step-size.

Although the "undo" step seems to be a waste of resources, using this heuristic guarantees monotonic convergence.

Example 7.2 (Solving a Linear Equation System)
When we solve linear equations of the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, in practice we solve $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{0}$ approximately by finding $\boldsymbol{x}_*$ that minimizes the squared error

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \tag{7.9}$$

if we use the Euclidean norm. The gradient of (7.9) with respect to $\boldsymbol{x}$ is

$$\nabla_{\boldsymbol{x}} = 2(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})^\top \boldsymbol{A}\,. \tag{7.10}$$

We can use this gradient directly in a gradient descent algorithm. However, for this particular special case, it turns out that there is an analytic solution, which can be found by setting the gradient to zero. We will see more on solving squared error problems in Chapter 9.

condition number

Remark. When applied to the solution of linear systems of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, gradient descent may converge slowly. The speed of convergence of gradient descent is dependent on the *condition number* $\kappa = \frac{\sigma(\boldsymbol{A})_{\max}}{\sigma(\boldsymbol{A})_{\min}}$, which is the ratio of the maximum to the minimum singular value (Section 4.5) of $\boldsymbol{A}$. The condition number essentially measures the ratio of the most curved direction versus the least curved direction, which corresponds to our imagery that poorly conditioned problems are long, thin valleys: They are very curved in one direction, but very flat in the other. Instead of directly solving $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, one could instead solve $\boldsymbol{P}^{-1}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = \boldsymbol{0}$, where $\boldsymbol{P}$ is called the *preconditioner*. The goal is to design $\boldsymbol{P}^{-1}$ such that $\boldsymbol{P}^{-1}\boldsymbol{A}$ has a better condition number, but at the same time $\boldsymbol{P}^{-1}$ is easy to compute. For further information on gradient descent, preconditioning, and convergence we refer to Boyd and Vandenberghe (2004, chapter 9). $\diamondsuit$

preconditioner

7.1.2 Gradient Descent With Momentum

As illustrated in Figure 7.2, the convergence of gradient descent may be very slow if the curvature of the optimization surface is such that there are regions that are poorly scaled. The curvature is such that the gradient descent steps hops between the walls of the valley and approaches the optimum in small steps. The proposed tweak to improve convergence is to give gradient descent some memory.

Goh (2017) wrote an intuitive blog post on gradient descent with momentum.

Gradient descent with momentum (Rumelhart et al., 1986) is a method that introduces an additional term to remember what happened in the previous iteration. This memory dampens oscillations and smoothes out the gradient updates. Continuing the ball analogy, the momentum term emulates the phenomenon of a heavy ball that is reluctant to change directions. The idea is to have a gradient update with memory to implement a moving average. The momentum-based method remembers the update $\Delta\boldsymbol{x}_i$ at each iteration i and determines the next update as a linear combination of the current and previous gradients

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i - \gamma_i((\nabla f)(\boldsymbol{x}_i))^\top + \alpha\Delta\boldsymbol{x}_i \tag{7.11}$$

$$\Delta\boldsymbol{x}_i = \boldsymbol{x}_i - \boldsymbol{x}_{i-1} = \alpha\Delta\boldsymbol{x}_{i-1} - \gamma_{i-1}((\nabla f)(\boldsymbol{x}_{i-1}))^\top , \tag{7.12}$$

where $\alpha \in [0, 1]$. Sometimes we will only know the gradient approximately. In such cases, the momentum term is useful since it averages out different noisy estimates of the gradient. One particularly useful way to obtain an approximate gradient is by using a stochastic approximation, which we discuss next.

7.1.3 Stochastic Gradient Descent

Computing the gradient can be very time consuming. However, often it is possible to find a "cheap" approximation of the gradient. Approximating the gradient is still useful as long as it points in roughly the same direction as the true gradient.

stochastic gradient descent

Stochastic gradient descent (often shortened as SGD) is a stochastic approximation of the gradient descent method for minimizing an objective function that is written as a sum of differentiable functions. The word stochastic here refers to the fact that we acknowledge that we do not know the gradient precisely, but instead only know a noisy approximation to it. By constraining the probability distribution of the approximate gradients, we can still theoretically guarantee that SGD will converge.

In machine learning, given $n = 1, \ldots, N$ data points, we often consider objective functions that are the sum of the losses L_n incurred by each example n. In mathematical notation, we have the form

$$L(\boldsymbol{\theta}) = \sum_{n=1}^{N} L_n(\boldsymbol{\theta}) , \tag{7.13}$$

where $\boldsymbol{\theta}$ is the vector of parameters of interest, i.e., we want to find $\boldsymbol{\theta}$ that minimizes L. An example from regression (Chapter 9) is the negative log-likelihood, which is expressed as a sum over log-likelihoods of individual examples so that

$$L(\boldsymbol{\theta}) = -\sum_{n=1}^{N} \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}) , \tag{7.14}$$

where $\boldsymbol{x}_n \in \mathbb{R}^D$ are the training inputs, y_n are the training targets, and $\boldsymbol{\theta}$ are the parameters of the regression model.

Standard gradient descent, as introduced previously, is a "batch" optimization method, i.e., optimization is performed using the full training set by updating the vector of parameters according to

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \gamma_i (\nabla L(\boldsymbol{\theta}_i))^\top = \boldsymbol{\theta}_i - \gamma_i \sum_{n=1}^{N} (\nabla L_n(\boldsymbol{\theta}_i))^\top \tag{7.15}$$

for a suitable step-size parameter γ_i. Evaluating the sum gradient may require expensive evaluations of the gradients from all individual functions L_n. When the training set is enormous and/or no simple formulas exist, evaluating the sums of gradients becomes very expensive.

Consider the term $\sum_{n=1}^{N}(\nabla L_n(\boldsymbol{\theta}_i))$ in (7.15). We can reduce the amount of computation by taking a sum over a smaller set of L_n. In contrast to batch gradient descent, which uses all L_n for $n = 1, \ldots, N$, we randomly choose a subset of L_n for mini-batch gradient descent. In the extreme case, we randomly select only a single L_n to estimate the gradient. The key insight about why taking a subset of data is sensible is to realize that for gradient descent to converge, we only require that the gradient is an unbiased estimate of the true gradient. In fact the term $\sum_{n=1}^{N}(\nabla L_n(\boldsymbol{\theta}_i))$ in (7.15) is an empirical estimate of the expected value (Section 6.4.1) of the gradient. Therefore, any other unbiased empirical estimate of the expected value, for example using any subsample of the data, would suffice for convergence of gradient descent.

Remark. When the learning rate decreases at an appropriate rate, and subject to relatively mild assumptions, stochastic gradient descent converges almost surely to local minimum (Bottou, 1998). $\diamondsuit$

Why should one consider using an approximate gradient? A major reason is practical implementation constraints, such as the size of central processing unit (CPU)/graphics processing unit (GPU) memory or limits on computational time. We can think of the size of the subset used to estimate the gradient in the same way that we thought of the size of a sample when estimating empirical means (Section 6.4.1). Large mini-batch sizes will provide accurate estimates of the gradient, reducing the variance in the parameter update. Furthermore, large mini-batches take advantage of highly optimized matrix operations in vectorized implementations of the cost and gradient. The reduction in variance leads to more stable convergence, but each gradient calculation will be more expensive.

In contrast, small mini-batches are quick to estimate. If we keep the mini-batch size small, the noise in our gradient estimate will allow us to get out of some bad local optima, which we may otherwise get stuck in. In machine learning, optimization methods are used for training by minimizing an objective function on the training data, but the overall goal is to improve generalization performance (Chapter 8). Since the goal in machine learning does not necessarily need a precise estimate of the minimum of the objective function, approximate gradients using mini-batch approaches have been widely used. Stochastic gradient descent is very effective in large-scale machine learning problems (Bottou et al., 2018),

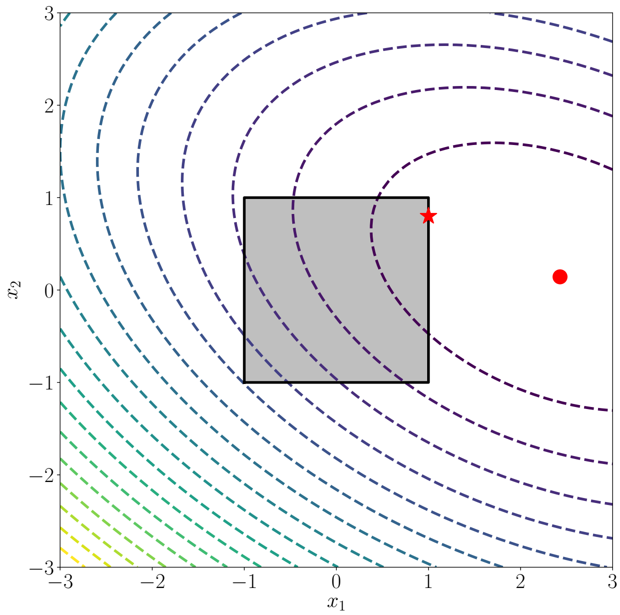


Figure 7.1 Illustration of constrained optimization. The unconstrained problem (indicated by the contour lines) has a minimum on the right side (indicated by the circle). The box constraints ($-1 \leqslant x \leqslant 1$ and $-1 \leqslant y \leqslant 1$) require that the optimal solution is within the box, resulting in an optimal value indicated by the star.

such as training deep neural networks on millions of images (Dean et al., 2012), topic models (Hoffman et al., 2013), reinforcement learning (Mnih et al., 2015), or training of large-scale Gaussian process models (Hensman et al., 2013; Gal et al., 2014).

7.2 Constrained Optimization and Lagrange Multipliers

In the previous section, we considered the problem of solving for the minimum of a function

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}), \tag{7.16}$$

where $f : \mathbb{R}^D \to \mathbb{R}$.

In this section, we have additional constraints. That is, for real-valued functions $g_i : \mathbb{R}^D \to \mathbb{R}$ for $i = 1, \ldots, m$, we consider the constrained optimization problem (see Figure 7.1 for an illustration)

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leqslant 0 \quad \text{for all} \quad i = 1, \ldots, m. \end{aligned} \tag{7.17}$$

It is worth pointing out that the functions f and g_i could be non-convex in general, and we will consider the convex case in the next section.

One obvious, but not very practical, way of converting the constrained problem (7.17) into an unconstrained one is to use an indicator function

$$J(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \mathbf{1}(g_i(\boldsymbol{x})), \tag{7.18}$$

where $\mathbf{1}(z)$ is an infinite step function

$$\mathbf{1}(z) = \begin{cases} 0 & \text{if } z \leqslant 0 \\ \infty & \text{otherwise} \end{cases}. \tag{7.19}$$

This gives infinite penalty if the constraint is not satisfied, and hence would provide the same solution. However, this infinite step function is equally difficult to optimize. We can overcome this difficulty by introducing *Lagrange multipliers*. The idea of Lagrange multipliers is to replace the step function with a linear function.

Lagrange multiplier

Lagrangian

We associate to problem (7.17) the *Lagrangian* by introducing the Lagrange multipliers $\lambda_i \geqslant 0$ corresponding to each inequality constraint respectively (Boyd and Vandenberghe, 2004, chapter 4) so that

$$\begin{aligned} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) &= f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}) && (7.20a) \\ &= f(\boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{g}(\boldsymbol{x}), && (7.20b) \end{aligned}$$

where in the last line we have concatenated all constraints $g_i(\boldsymbol{x})$ into a vector $\boldsymbol{g}(\boldsymbol{x})$, and all the Lagrange multipliers into a vector $\boldsymbol{\lambda} \in \mathbb{R}^m$.

We now introduce the idea of Lagrangian duality. In general, duality in optimization is the idea of converting an optimization problem in one set of variables $\boldsymbol{x}$ (called the primal variables), into another optimization problem in a different set of variables $\boldsymbol{\lambda}$ (called the dual variables). We introduce two different approaches to duality: In this section, we discuss Lagrangian duality; in Section 7.3.3, we discuss Legendre-Fenchel duality.

Definition 7.1. The problem in (7.17)

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leqslant 0 \quad \text{for all} \quad i = 1, \dots, m \end{aligned} \tag{7.21}$$

primal problem

Lagrangian dual problem

is known as the *primal problem*, corresponding to the primal variables x. The associated *Lagrangian dual problem* is given by

$$\begin{aligned} \max_{\boldsymbol{\lambda} \in \mathbb{R}^m} \quad & \mathfrak{D}(\boldsymbol{\lambda}) \\ \text{subject to} \quad & \boldsymbol{\lambda} \geqslant \boldsymbol{0}, \end{aligned} \tag{7.22}$$

where $\boldsymbol{\lambda}$ are the dual variables and $\mathfrak{D}(\boldsymbol{\lambda}) = \min_{\boldsymbol{x} \in \mathbb{R}^d} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$.

Remark. In the discussion of Definition 7.1, we use two concepts that are also of independent interest (Boyd and Vandenberghe, 2004).

minimax inequality

First is the *minimax inequality*, which says that for any function with two arguments $\varphi(\boldsymbol{x}, \boldsymbol{y})$, the maximin is less than the minimax, i.e.,

$$\max_{\boldsymbol{y}} \min_{\boldsymbol{x}} \varphi(\boldsymbol{x}, \boldsymbol{y}) \leqslant \min_{\boldsymbol{x}} \max_{\boldsymbol{y}} \varphi(\boldsymbol{x}, \boldsymbol{y}). \tag{7.23}$$

This inequality can be proved by considering the inequality

$$\text{For all } \boldsymbol{x}, \boldsymbol{y} \qquad \min_{\boldsymbol{x}} \varphi(\boldsymbol{x}, \boldsymbol{y}) \leqslant \max_{\boldsymbol{y}} \varphi(\boldsymbol{x}, \boldsymbol{y}) . \tag{7.24}$$

Note that taking the maximum over $\boldsymbol{y}$ of the left-hand side of (7.24) maintains the inequality since the inequality is true for all $\boldsymbol{y}$. Similarly, we can take the minimum over $\boldsymbol{x}$ of the right-hand side of (7.24) to obtain (7.23).

The second concept is *weak duality*, which uses (7.23) to show that primal values are always greater than or equal to dual values. This is described in more detail in (7.27). ♢

weak duality

Recall that the difference between $J(\boldsymbol{x})$ in (7.18) and the Lagrangian in (7.20b) is that we have relaxed the indicator function to a linear function. Therefore, when $\lambda \geqslant 0$, the Lagrangian $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is a lower bound of $J(\boldsymbol{x})$. Hence, the maximum of $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ with respect to $\boldsymbol{\lambda}$ is

$$J(\boldsymbol{x}) = \max_{\boldsymbol{\lambda} \geqslant \boldsymbol{0}} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) . \tag{7.25}$$

Recall that the original problem was minimizing $J(\boldsymbol{x})$,

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \max_{\boldsymbol{\lambda} \geqslant \boldsymbol{0}} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) . \tag{7.26}$$

By the minimax inequality (7.23), it follows that swapping the order of the minimum and maximum results in a smaller value, i.e.,

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \max_{\boldsymbol{\lambda} \geqslant \boldsymbol{0}} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \geqslant \max_{\boldsymbol{\lambda} \geqslant \boldsymbol{0}} \min_{\boldsymbol{x} \in \mathbb{R}^d} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) . \tag{7.27}$$

This is also known as *weak duality*. Note that the inner part of the right-hand side is the dual objective function $\mathfrak{D}(\boldsymbol{\lambda})$ and the definition follows.

weak duality

In contrast to the original optimization problem, which has constraints, $\min_{\boldsymbol{x} \in \mathbb{R}^d} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is an unconstrained optimization problem for a given value of $\boldsymbol{\lambda}$. If solving $\min_{\boldsymbol{x} \in \mathbb{R}^d} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is easy, then the overall problem is easy to solve. We can see this by observing from (7.20b) that $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is affine with respect to $\boldsymbol{\lambda}$. Therefore $\min_{\boldsymbol{x} \in \mathbb{R}^d} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ is a pointwise minimum of affine functions of $\boldsymbol{\lambda}$, and hence $\mathfrak{D}(\boldsymbol{\lambda})$ is concave even though $f(\cdot)$ and $g_i(\cdot)$ may be nonconvex. The outer problem, maximization over $\boldsymbol{\lambda}$, is the maximum of a concave function and can be efficiently computed.

Assuming $f(\cdot)$ and $g_i(\cdot)$ are differentiable, we find the Lagrange dual problem by differentiating the Lagrangian with respect to $\boldsymbol{x}$, setting the differential to zero, and solving for the optimal value. We will discuss two concrete examples in Sections 7.3.1 and 7.3.2, where $f(\cdot)$ and $g_i(\cdot)$ are convex.

Remark (Equality Constraints). Consider (7.17) with additional equality constraints

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leqslant 0 \quad \text{for all} \quad i = 1, \dots, m \\ & h_j(\boldsymbol{x}) = 0 \quad \text{for all} \quad j = 1, \dots, n . \end{aligned} \tag{7.28}$$

We can model equality constraints by replacing them with two inequality constraints. That is for each equality constraint $h_j(\boldsymbol{x}) = 0$ we equivalently replace it by two constraints $h_j(\boldsymbol{x}) \leqslant 0$ and $h_j(\boldsymbol{x}) \geqslant 0$. It turns out that the resulting Lagrange multipliers are then unconstrained.

Therefore, we constrain the Lagrange multipliers corresponding to the inequality constraints in (7.28) to be non-negative, and leave the Lagrange multipliers corresponding to the equality constraints unconstrained. ♢

7.3 Convex Optimization

convex optimization problem

strong duality

We focus our attention of a particularly useful class of optimization problems, where we can guarantee global optimality. When $f(\cdot)$ is a convex function, and when the constraints involving $g(\cdot)$ and $h(\cdot)$ are convex sets, this is called a *convex optimization problem*. In this setting, we have *strong duality*: The optimal solution of the dual problem is the same as the optimal solution of the primal problem. The distinction between convex functions and convex sets are often not strictly presented in machine learning literature, but one can often infer the implied meaning from context.

convex set

Definition 7.2. A set $\mathcal{C}$ is a *convex set* if for any $x, y \in \mathcal{C}$ and for any scalar θ with $0 \leqslant \theta \leqslant 1$, we have

$$\theta x + (1-\theta)y \in \mathcal{C} . \tag{7.29}$$

Figure 7.2 Example of a convex set.

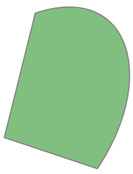
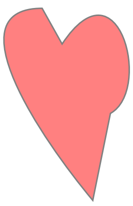


Figure 7.3 Example of a nonconvex set.



Convex sets are sets such that a straight line connecting any two elements of the set lie inside the set. Figures 7.2 and 7.3 illustrate convex and nonconvex sets, respectively.

Convex functions are functions such that a straight line between any two points of the function lie above the function. Figure 7.2 shows a nonconvex function, and Figure 7.2 shows a convex function. Another convex function is shown in Figure 7.2.

Definition 7.3. Let function $f : \mathbb{R}^D \to \mathbb{R}$ be a function whose domain is a convex set. The function f is a *convex function* if for all $\boldsymbol{x}, \boldsymbol{y}$ in the domain of f, and for any scalar θ with $0 \leqslant \theta \leqslant 1$, we have

$$f(\theta \boldsymbol{x} + (1-\theta)\boldsymbol{y}) \leqslant \theta f(\boldsymbol{x}) + (1-\theta) f(\boldsymbol{y}) . \tag{7.30}$$

Remark. A *concave function* is the negative of a convex function. ♢

convex function

concave function

The constraints involving $g(\cdot)$ and $h(\cdot)$ in (7.28) truncate functions at a scalar value, resulting in sets. Another relation between convex functions and convex sets is to consider the set obtained by "filling in" a convex function. A convex function is a bowl-like object, and we imagine pouring water into it to fill it up. This resulting filled-in set, called the *epigraph* of the convex function, is a convex set.

epigraph

If a function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable, we can specify convexity in

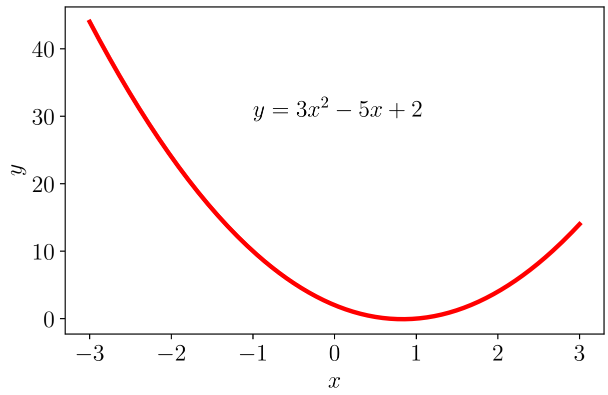


Figure 7.2 Example of a convex function.

terms of its gradient $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$ (Section 5.2). A function $f(\boldsymbol{x})$ is convex if and only if for any two points $\boldsymbol{x}, \boldsymbol{y}$ it holds that

$$f(\boldsymbol{y}) \geqslant f(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) . \tag{7.31}$$

If we further know that a function $f(\boldsymbol{x})$ is twice differentiable, that is, the Hessian (5.147) exists for all values in the domain of $\boldsymbol{x}$, then the function $f(\boldsymbol{x})$ is convex if and only if $\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x})$ is positive semidefinite (Boyd and Vandenberghe, 2004).

Example 7.3

The negative entropy $f(x) = x \log_2 x$ is convex for $x > 0$. A visualization of the function is shown in Figure 7.2, and we can see that the function is convex. To illustrate the previous definitions of convexity, let us check the calculations for two points $x = 2$ and $x = 4$. Note that to prove convexity of $f(x)$ we would need to check for all points $x \in \mathbb{R}$.

Recall Definition 7.3. Consider a point midway between the two points (that is $\theta = 0.5$); then the left-hand side is $f(0.5 \cdot 2 + 0.5 \cdot 4) = 3 \log_2 3 \approx 4.75$. The right-hand side is $0.5(2 \log_2 2) + 0.5(4 \log_2 4) = 1 + 4 = 5$. And therefore the definition is satisfied.

Since $f(x)$ is differentiable, we can alternatively use (7.31). Calculating the derivative of $f(x)$, we obtain

$$\nabla_x (x \log_2 x) = 1 \cdot \log_2 x + x \cdot \frac{1}{x \log_e 2} = \log_2 x + \frac{1}{\log_e 2} . \tag{7.32}$$

Using the same two test points $x = 2$ and $x = 4$, the left-hand side of (7.31) is given by $f(4) = 8$. The right-hand side is

$$f(\boldsymbol{x}) + \nabla^\top_{\boldsymbol{x}} (\boldsymbol{y} - \boldsymbol{x}) = f(2) + \nabla f(2) \cdot (4 - 2) \tag{7.33a}$$

$$= 2 + (1 + \frac{1}{\log_e 2}) \cdot 2 \approx 6.9 . \tag{7.33b}$$

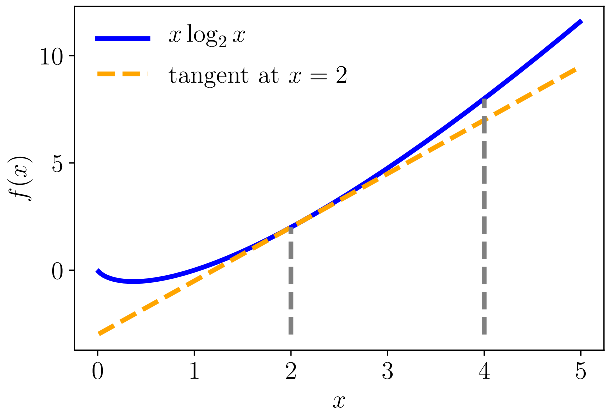


Figure 7.2 The negative entropy function (which is convex) and its tangent at $x = 2$.

We can check that a function or set is convex from first principles by recalling the definitions. In practice, we often rely on operations that preserve convexity to check that a particular function or set is convex. Although the details are vastly different, this is again the idea of closure that we introduced in Chapter 2 for vector spaces.

Example 7.4

A nonnegative weighted sum of convex functions is convex. Observe that if f is a convex function, and $\alpha \geqslant 0$ is a nonnegative scalar, then the function αf is convex. We can see this by multiplying α to both sides of the equation in Definition 7.3, and recalling that multiplying a nonnegative number does not change the inequality.

If f_1 and f_2 are convex functions, then we have by the definition

$$f_1(\theta \boldsymbol{x} + (1-\theta)\boldsymbol{y}) \leqslant \theta f_1(\boldsymbol{x}) + (1-\theta) f_1(\boldsymbol{y}) \tag{7.34}$$

$$f_2(\theta \boldsymbol{x} + (1-\theta)\boldsymbol{y}) \leqslant \theta f_2(\boldsymbol{x}) + (1-\theta) f_2(\boldsymbol{y})\,. \tag{7.35}$$

Summing up both sides gives us

$$\begin{aligned} &f_1(\theta \boldsymbol{x} + (1-\theta)\boldsymbol{y}) + f_2(\theta \boldsymbol{x} + (1-\theta)\boldsymbol{y}) \\ &\leqslant \theta f_1(\boldsymbol{x}) + (1-\theta) f_1(\boldsymbol{y}) + \theta f_2(\boldsymbol{x}) + (1-\theta) f_2(\boldsymbol{y})\,, \end{aligned} \tag{7.36}$$

where the right-hand side can be rearranged to

$$\theta(f_1(\boldsymbol{x}) + f_2(\boldsymbol{x})) + (1-\theta)(f_1(\boldsymbol{y}) + f_2(\boldsymbol{y}))\,, \tag{7.37}$$

completing the proof that the sum of convex functions is convex.

Combining the preceding two facts, we see that $\alpha f_1(\boldsymbol{x}) + \beta f_2(\boldsymbol{x})$ is convex for $\alpha, \beta \geqslant 0$. This closure property can be extended using a similar argument for nonnegative weighted sums of more than two convex functions.

Remark. The inequality in (7.30) is sometimes called *Jensen's inequality*. In fact, a whole class of inequalities for taking nonnegative weighted sums of convex functions are all called Jensen's inequality. ♢

Jensen's inequality

In summary, a constrained optimization problem is called a *convex optimization problem* if

convex optimization problem

$$\begin{aligned} \min_{\boldsymbol{x}} & f(\boldsymbol{x}) \\ \text{subject to } g_i(\boldsymbol{x}) \leqslant 0 & \quad \text{for all} \quad i = 1, \dots, m \\ h_j(\boldsymbol{x}) = 0 & \quad \text{for all} \quad j = 1, \dots, n\,, \end{aligned} \tag{7.38}$$

where all functions $f(\boldsymbol{x})$ and $g_i(\boldsymbol{x})$ are convex functions, and all $h_j(\boldsymbol{x}) = 0$ are convex sets. In the following, we will describe two classes of convex optimization problems that are widely used and well understood.

7.3.1 Linear Programming

Consider the special case when all the preceding functions are linear, i.e.,

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} & \quad \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \quad \boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}\,, \end{aligned} \tag{7.39}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{b} \in \mathbb{R}^m$. This is known as a *linear program*. It has d variables and m linear constraints. The Lagrangian is given by

linear program

Linear programs are one of the most widely used approaches in industry.

$$\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\,, \tag{7.40}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of non-negative Lagrange multipliers. Rearranging the terms corresponding to $\boldsymbol{x}$ yields

$$\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = (\boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\lambda})^\top \boldsymbol{x} - \boldsymbol{\lambda}^\top \boldsymbol{b}\,. \tag{7.41}$$

Taking the derivative of $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ with respect to $\boldsymbol{x}$ and setting it to zero gives us

$$\boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\lambda} = \boldsymbol{0}\,. \tag{7.42}$$

Therefore, the dual Lagrangian is $\mathfrak{D}(\boldsymbol{\lambda}) = -\boldsymbol{\lambda}^\top \boldsymbol{b}$. Recall we would like to maximize $\mathfrak{D}(\boldsymbol{\lambda})$. In addition to the constraint due to the derivative of $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ being zero, we also have the fact that $\boldsymbol{\lambda} \geqslant \boldsymbol{0}$, resulting in the following dual optimization problem

It is convention to minimize the primal and maximize the dual.

$$\begin{aligned} \max_{\boldsymbol{\lambda} \in \mathbb{R}^m} & \quad -\boldsymbol{b}^\top \boldsymbol{\lambda} \\ \text{subject to} & \quad \boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\lambda} = \boldsymbol{0} \\ & \quad \boldsymbol{\lambda} \geqslant \boldsymbol{0}\,. \end{aligned} \tag{7.43}$$

This is also a linear program, but with m variables. We have the choice of solving the primal (7.39) or the dual (7.43) program depending on whether m or d is larger. Recall that d is the number of variables and m is the number of constraints in the primal linear program.

Example 7.5 (Linear Program)
Consider the linear program

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^2} \quad & -\begin{bmatrix}5\\3\end{bmatrix}^\top \begin{bmatrix}x_1\\x_2\end{bmatrix} \\ \text{subject to} \quad & \begin{bmatrix}2 & 2\\ 2 & -4\\ -2 & 1\\ 0 & -1\\ 0 & 1\end{bmatrix} \begin{bmatrix}x_1\\x_2\end{bmatrix} \leqslant \begin{bmatrix}33\\8\\5\\-1\\8\end{bmatrix} \end{aligned} \tag{7.44}$$

with two variables. This program is also shown in Figure 7.1. The objective function is linear, resulting in linear contour lines. The constraint set in standard form is translated into the legend. The optimal value must lie in the shaded (feasible) region, and is indicated by the star.

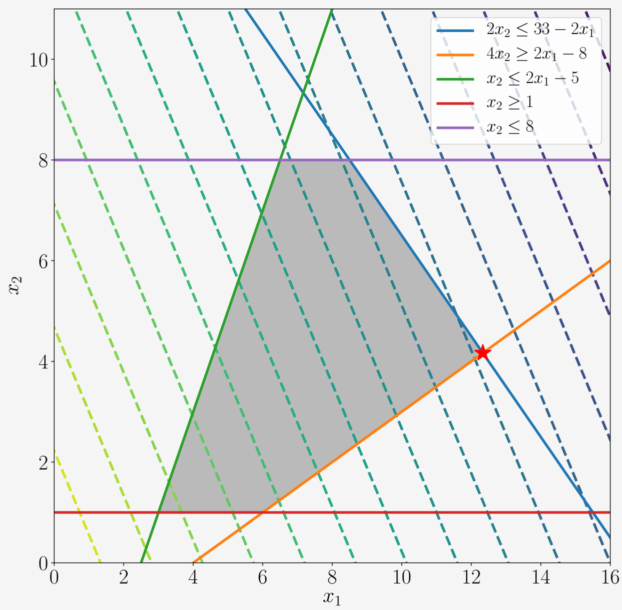


Figure 7.1 Illustration of a linear program. The unconstrained problem (indicated by the contour lines) has a minimum on the right side. The optimal value given the constraints are shown by the star.

7.3.2 Quadratic Programming

Consider the case of a convex quadratic objective function, where the constraints are affine, i.e.,

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^d} \quad & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}\,, \end{aligned} \tag{7.45}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times d}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{c} \in \mathbb{R}^d$. The square symmetric matrix $\boldsymbol{Q} \in \mathbb{R}^{d\times d}$ is positive definite, and therefore the objective function is convex. This is known as a *quadratic program*. Observe that it has d variables and m linear constraints.

Example 7.6 (Quadratic Program)
Consider the quadratic program

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} \quad \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^\top \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 5 \\ 3 \end{bmatrix}^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{7.46}$$

$$\text{subject to} \quad \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leqslant \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{7.47}$$

of two variables. The program is also illustrated in Figure 7.1. The objective function is quadratic with a positive semidefinite matrix $\boldsymbol{Q}$, resulting in elliptical contour lines. The optimal value must lie in the shaded (feasible) region, and is indicated by the star.

The Lagrangian is given by

$$\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \tag{7.48a}$$

$$= \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + (\boldsymbol{c} + \boldsymbol{A}^\top\boldsymbol{\lambda})^\top \boldsymbol{x} - \boldsymbol{\lambda}^\top \boldsymbol{b}\,, \tag{7.48b}$$

where again we have rearranged the terms. Taking the derivative of $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ with respect to $\boldsymbol{x}$ and setting it to zero gives

$$\boldsymbol{Q}\boldsymbol{x} + (\boldsymbol{c} + \boldsymbol{A}^\top\boldsymbol{\lambda}) = \boldsymbol{0}\,. \tag{7.49}$$

Assuming that $\boldsymbol{Q}$ is invertible, we get

$$\boldsymbol{x} = -\boldsymbol{Q}^{-1}(\boldsymbol{c} + \boldsymbol{A}^\top\boldsymbol{\lambda})\,. \tag{7.50}$$

Substituting (7.50) into the primal Lagrangian $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda})$, we get the dual Lagrangian

$$\mathfrak{D}(\boldsymbol{\lambda}) = -\frac{1}{2}(\boldsymbol{c} + \boldsymbol{A}^\top\boldsymbol{\lambda})^\top \boldsymbol{Q}^{-1}(\boldsymbol{c} + \boldsymbol{A}^\top\boldsymbol{\lambda}) - \boldsymbol{\lambda}^\top \boldsymbol{b}\,. \tag{7.51}$$

Therefore, the dual optimization problem is given by

$$\begin{aligned} \max_{\boldsymbol{\lambda}\in\mathbb{R}^m} \quad & -\frac{1}{2}(\boldsymbol{c}+\boldsymbol{A}^\top\boldsymbol{\lambda})^\top\boldsymbol{Q}^{-1}(\boldsymbol{c}+\boldsymbol{A}^\top\boldsymbol{\lambda})-\boldsymbol{\lambda}^\top\boldsymbol{b} \\ \text{subject to} \quad & \boldsymbol{\lambda}\geqslant\boldsymbol{0}\,. \end{aligned} \tag{7.52}$$

We will see an application of quadratic programming in machine learning in Chapter 12.

7.3.3 Legendre–Fenchel Transform and Convex Conjugate

Let us revisit the idea of duality from Section 7.2, without considering constraints. One useful fact about a convex set is that it can be equivalently described by its supporting hyperplanes. A hyperplane is called a *supporting hyperplane* of a convex set if it intersects the convex set, and the convex set is contained on just one side of it. Recall that we can fill up a convex function to obtain the epigraph, which is a convex set. Therefore, we can also describe convex functions in terms of their supporting hyperplanes. Furthermore, observe that the supporting hyperplane just touches the convex function, and is in fact the tangent to the function at that point. And recall that the tangent of a function $f(\boldsymbol{x})$ at a given point $\boldsymbol{x}_0$ is the evaluation of the gradient of that function at that point $\left.\frac{\mathrm{d}f(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_0}$. In summary, because convex sets can be equivalently described by their supporting hyperplanes, convex functions can be equivalently described by a function of their gradient. The *Legendre transform* formalizes this concept.

supporting hyperplane

Legendre transform

Physics students are often introduced to the Legendre transform as relating the Lagrangian and the Hamiltonian in classical mechanics.

Legendre-Fenchel transform

convex conjugate

We begin with the most general definition, which unfortunately has a counter-intuitive form, and look at special cases to relate the definition to the intuition described in the preceding paragraph. The *Legendre-Fenchel transform* is a transformation (in the sense of a Fourier transform) from a convex differentiable function $f(\boldsymbol{x})$ to a function that depends on the tangents $s(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$. It is worth stressing that this is a transformation of the function $f(\cdot)$ and not the variable $\boldsymbol{x}$ or the function evaluated at $\boldsymbol{x}$. The Legendre-Fenchel transform is also known as the *convex conjugate* (for reasons we will see soon) and is closely related to duality (Hiriart-Urruty and Lemaréchal, 2001, chapter 5).

convex conjugate

Definition 7.4. The *convex conjugate* of a function $f : \mathbb{R}^D \to \mathbb{R}$ is a function f^* defined by

$$f^*(\boldsymbol{s}) = \sup_{\boldsymbol{x}\in\mathbb{R}^D} \left(\langle \boldsymbol{s}, \boldsymbol{x}\rangle - f(\boldsymbol{x})\right)\,. \tag{7.53}$$

Note that the preceding convex conjugate definition does not need the function f to be convex nor differentiable. In Definition 7.4, we have used a general inner product (Section 3.2) but in the rest of this section we will consider the standard dot product between finite-dimensional vectors ($\langle \boldsymbol{s}, \boldsymbol{x} \rangle = \boldsymbol{s}^\top \boldsymbol{x}$) to avoid too many technical details.

To understand Definition 7.4 in a geometric fashion, consider a nice simple one-dimensional convex and differentiable function, for example $f(x) = x^2$. Note that since we are looking at a one-dimensional problem, hyperplanes reduce to a line. Consider a line $y = sx + c$. Recall that we are able to describe convex functions by their supporting hyperplanes, so let us try to describe this function $f(x)$ by its supporting lines. Fix the gradient of the line $s \in \mathbb{R}$ and for each point $(x_0, f(x_0))$ on the graph of f, find the minimum value of c such that the line still intersects $(x_0, f(x_0))$. Note that the minimum value of c is the place where a line with slope s "just touches" the function $f(x) = x^2$. The line passing through $(x_0, f(x_0))$ with gradient s is given by

This derivation is easiest to understand by drawing the reasoning as it progresses.

$$y - f(x_0) = s(x - x_0)\,. \tag{7.54}$$

The y-intercept of this line is $-sx_0 + f(x_0)$. The minimum of c for which $y = sx + c$ intersects with the graph of f is therefore

$$\inf_{x_0} -sx_0 + f(x_0)\,. \tag{7.55}$$

The preceding convex conjugate is by convention defined to be the negative of this. The reasoning in this paragraph did not rely on the fact that we chose a one-dimensional convex and differentiable function, and holds for $f : \mathbb{R}^D \to \mathbb{R}$, which are nonconvex and non-differentiable.

The classical Legendre transform is defined on convex differentiable functions in $\mathbb{R}^D$.

Remark. Convex differentiable functions such as the example $f(x) = x^2$ is a nice special case, where there is no need for the supremum, and there is a one-to-one correspondence between a function and its Legendre transform. Let us derive this from first principles. For a convex differentiable function, we know that at x_0 the tangent touches $f(x_0)$ so that

$$f(x_0) = sx_0 + c\,. \tag{7.56}$$

Recall that we want to describe the convex function $f(x)$ in terms of its gradient $\nabla_x f(x)$, and that $s = \nabla_x f(x_0)$. We rearrange to get an expression for $-c$ to obtain

$$-c = sx_0 - f(x_0)\,. \tag{7.57}$$

Note that $-c$ changes with x_0 and therefore with s, which is why we can think of it as a function of s, which we call

$$f^*(s) := sx_0 - f(x_0)\,. \tag{7.58}$$

Comparing (7.58) with Definition 7.4, we see that (7.58) is a special case (without the supremum). $\diamondsuit$

The conjugate function has nice properties; for example, for convex functions, applying the Legendre transform again gets us back to the original function. In the same way that the slope of $f(x)$ is s, the slope of $f^*(s)$

is x. The following two examples show common uses of convex conjugates in machine learning.

Example 7.7 (Convex Conjugates)
To illustrate the application of convex conjugates, consider the quadratic function

$$f(\boldsymbol{y}) = \frac{\lambda}{2}\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y} \tag{7.59}$$

based on a positive definite matrix $\boldsymbol{K} \in \mathbb{R}^{n\times n}$. We denote the primal variable to be $\boldsymbol{y} \in \mathbb{R}^n$ and the dual variable to be $\boldsymbol{\alpha} \in \mathbb{R}^n$.

Applying Definition 7.4, we obtain the function

$$f^*(\boldsymbol{\alpha}) = \sup_{\boldsymbol{y}\in\mathbb{R}^n} \langle \boldsymbol{y}, \boldsymbol{\alpha}\rangle - \frac{\lambda}{2}\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y}\,. \tag{7.60}$$

Since the function is differentiable, we can find the maximum by taking the derivative and with respect to $\boldsymbol{y}$ setting it to zero.

$$\frac{\partial\left[\langle \boldsymbol{y}, \boldsymbol{\alpha}\rangle - \frac{\lambda}{2}\boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y}\right]}{\partial \boldsymbol{y}} = (\boldsymbol{\alpha} - \lambda \boldsymbol{K}^{-1}\boldsymbol{y})^\top \tag{7.61}$$

and hence when the gradient is zero we have $\boldsymbol{y} = \frac{1}{\lambda}\boldsymbol{K}\boldsymbol{\alpha}$. Substituting into (7.60) yields

$$f^*(\boldsymbol{\alpha}) = \frac{1}{\lambda}\boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} - \frac{\lambda}{2}\left(\frac{1}{\lambda}\boldsymbol{K}\boldsymbol{\alpha}\right)^\top \boldsymbol{K}^{-1}\left(\frac{1}{\lambda}\boldsymbol{K}\boldsymbol{\alpha}\right) = \frac{1}{2\lambda}\boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha}\,. \tag{7.62}$$

Example 7.8
In machine learning, we often use sums of functions; for example, the objective function of the training set includes a sum of the losses for each example in the training set. In the following, we derive the convex conjugate of a sum of losses $\ell(t)$, where $\ell : \mathbb{R} \to \mathbb{R}$. This also illustrates the application of the convex conjugate to the vector case. Let $\mathcal{L}(\boldsymbol{t}) = \sum_{i=1}^n \ell_i(t_i)$. Then,

$$\begin{aligned}
\mathcal{L}^*(\boldsymbol{z}) &= \sup_{\boldsymbol{t}\in\mathbb{R}^n} \langle \boldsymbol{z}, \boldsymbol{t}\rangle - \sum_{i=1}^n \ell_i(t_i) && \text{(7.63a)}\\
&= \sup_{\boldsymbol{t}\in\mathbb{R}^n} \sum_{i=1}^n z_i t_i - \ell_i(t_i) \qquad \text{definition of dot product} && \text{(7.63b)}\\
&= \sum_{i=1}^n \sup_{\boldsymbol{t}\in\mathbb{R}^n} z_i t_i - \ell_i(t_i) && \text{(7.63c)}
\end{aligned}$$

$$= \sum_{i=1}^{n} \ell_i^*(z_i)\,. \qquad \text{definition of conjugate} \tag{7.63d}$$

Recall that in Section 7.2 we derived a dual optimization problem using Lagrange multipliers. Furthermore, for convex optimization problems we have strong duality, that is the solutions of the primal and dual problem match. The Legendre-Fenchel transform described here also can be used to derive a dual optimization problem. Furthermore, when the function is convex and differentiable, the supremum is unique. To further investigate the relation between these two approaches, let us consider a linear equality constrained convex optimization problem.

Example 7.9
Let $f(\boldsymbol{y})$ and $g(\boldsymbol{x})$ be convex functions, and $\boldsymbol{A}$ a real matrix of appropriate dimensions such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$. Then

$$\min_{\boldsymbol{x}} f(\boldsymbol{A}\boldsymbol{x}) + g(\boldsymbol{x}) = \min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{y}} f(\boldsymbol{y}) + g(\boldsymbol{x}). \tag{7.64}$$

By introducing the Lagrange multiplier $\boldsymbol{u}$ for the constraints $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$,

$$\min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{y}} f(\boldsymbol{y}) + g(\boldsymbol{x}) = \min_{\boldsymbol{x},\boldsymbol{y}} \max_{\boldsymbol{u}} f(\boldsymbol{y}) + g(\boldsymbol{x}) + (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})^\top \boldsymbol{u} \tag{7.65a}$$

$$= \max_{\boldsymbol{u}} \min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{y}) + g(\boldsymbol{x}) + (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})^\top \boldsymbol{u}\,, \tag{7.65b}$$

where the last step of swapping max and min is due to the fact that $f(\boldsymbol{y})$ and $g(\boldsymbol{x})$ are convex functions. By splitting up the dot product term and collecting $\boldsymbol{x}$ and $\boldsymbol{y}$,

$$\max_{\boldsymbol{u}} \min_{\boldsymbol{x},\boldsymbol{y}} f(\boldsymbol{y}) + g(\boldsymbol{x}) + (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})^\top \boldsymbol{u} \tag{7.66a}$$

$$= \max_{\boldsymbol{u}} \left[\min_{\boldsymbol{y}} -\boldsymbol{y}^\top \boldsymbol{u} + f(\boldsymbol{y})\right] + \left[\min_{\boldsymbol{x}} (\boldsymbol{A}\boldsymbol{x})^\top \boldsymbol{u} + g(\boldsymbol{x})\right] \tag{7.66b}$$

$$= \max_{\boldsymbol{u}} \left[\min_{\boldsymbol{y}} -\boldsymbol{y}^\top \boldsymbol{u} + f(\boldsymbol{y})\right] + \left[\min_{\boldsymbol{x}} \boldsymbol{x}^\top \boldsymbol{A}^\top \boldsymbol{u} + g(\boldsymbol{x})\right] \tag{7.66c}$$

Recall the convex conjugate (Definition 7.4) and the fact that dot products are symmetric,

For general inner products, $\boldsymbol{A}^\top$ is replaced by the adjoint $\boldsymbol{A}^*$.

$$\max_{\boldsymbol{u}} \left[\min_{\boldsymbol{y}} -\boldsymbol{y}^\top \boldsymbol{u} + f(\boldsymbol{y})\right] + \left[\min_{\boldsymbol{x}} \boldsymbol{x}^\top \boldsymbol{A}^\top \boldsymbol{u} + g(\boldsymbol{x})\right] \tag{7.67a}$$

$$= \max_{\boldsymbol{u}} -f^*(\boldsymbol{u}) - g^*(-\boldsymbol{A}^\top \boldsymbol{u})\,. \tag{7.67b}$$

Therefore, we have shown that

$$\min_{\boldsymbol{x}} f(\boldsymbol{A}\boldsymbol{x}) + g(\boldsymbol{x}) = \max_{\boldsymbol{u}} -f^*(\boldsymbol{u}) - g^*(-\boldsymbol{A}^\top \boldsymbol{u})\,. \tag{7.68}$$

The Legendre-Fenchel conjugate turns out to be quite useful for machine learning problems that can be expressed as convex optimization problems. In particular, for convex loss functions that apply independently to each example, the conjugate loss is a convenient way to derive a dual problem.

7.4 Further Reading

Continuous optimization is an active area of research, and we do not try to provide a comprehensive account of recent advances.

From a gradient descent perspective, there are two major weaknesses which each have their own set of literature. The first challenge is the fact that gradient descent is a first-order algorithm, and does not use information about the curvature of the surface. When there are long valleys, the gradient points perpendicularly to the direction of interest. The idea of momentum can be generalized to a general class of acceleration methods (Nesterov, 2018). Conjugate gradient methods avoid the issues faced by gradient descent by taking previous directions into account (Shewchuk, 1994). Second-order methods such as Newton methods use the Hessian to provide information about the curvature. Many of the choices for choosing step-sizes and ideas like momentum arise by considering the curvature of the objective function (Goh, 2017; Bottou et al., 2018). Quasi-Newton methods such as L-BFGS try to use cheaper computational methods to approximate the Hessian (Nocedal and Wright, 2006). Recently there has been interest in other metrics for computing descent directions, resulting in approaches such as mirror descent (Beck and Teboulle, 2003) and natural gradient (Toussaint, 2012).

The second challenge is to handle non-differentiable functions. Gradient methods are not well defined when there are kinks in the function. In these cases, *subgradient methods* can be used (Shor, 1985). For further information and algorithms for optimizing non-differentiable functions, we refer to the book by Bertsekas (1999). There is a vast amount of literature on different approaches for numerically solving continuous optimization problems, including algorithms for constrained optimization problems. Good starting points to appreciate this literature are the books by Luenberger (1969) and Bonnans et al. (2006). A recent survey of continuous optimization is provided by Bubeck (2015).

Hugo Gonçalves' blog is also a good resource for an easier introduction to Legendre–Fenchel transforms: `https://tinyurl.com/ydaal7hj`

Modern applications of machine learning often mean that the size of datasets prohibit the use of batch gradient descent, and hence stochastic gradient descent is the current workhorse of large-scale machine learning methods. Recent surveys of the literature include Hazan (2015) and Bottou et al. (2018).

For duality and convex optimization, the book by Boyd and Vandenberghe (2004) includes lectures and slides online. A more mathematical treatment is provided by Bertsekas (2009), and recent book by one of the key researchers in the area of optimization is Nesterov (2018). Convex optimization is based upon convex analysis, and the reader interested in more foundational results about convex functions is referred to Rockafellar (1970), Hiriart-Urruty and Lemaréchal (2001), and Borwein and Lewis (2006). Legendre–Fenchel transforms are also covered in the aforementioned books on convex analysis, but a more beginner-friendly presentation is available at Zia et al. (2009). The role of Legendre–Fenchel transforms in the analysis of convex optimization algorithms is surveyed in Polyak (2016).

Exercises

7.1 Consider the univariate function

$$f(x) = x^3 + 6x^2 - 3x - 5.$$

Find its stationary points and indicate whether they are maximum, minimum, or saddle points.

7.2 Consider the update equation for stochastic gradient descent (Equation (7.15)). Write down the update when we use a mini-batch size of one.

7.3 Consider whether the following statements are true or false:

a. The intersection of any two convex sets is convex.
b. The union of any two convex sets is convex.
c. The difference of a convex set A from another convex set B is convex.

7.4 Consider whether the following statements are true or false:

a. The sum of any two convex functions is convex.
b. The difference of any two convex functions is convex.
c. The product of any two convex functions is convex.
d. The maximum of any two convex functions is convex.

7.5 Express the following optimization problem as a standard linear program in matrix notation

$$\max_{\boldsymbol{x} \in \mathbb{R}^2,\ \xi \in \mathbb{R}} \boldsymbol{p}^\top \boldsymbol{x} + \xi$$

subject to the constraints that $\xi \geqslant 0$, $x_0 \leqslant 0$ and $x_1 \leqslant 3$.

7.6 Consider the linear program illustrated in Figure 7.1,

$$\min_{\boldsymbol{x} \in \mathbb{R}^2} - \begin{bmatrix} 5 \\ 3 \end{bmatrix}^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\text{subject to} \begin{bmatrix} 2 & 2 \\ 2 & -4 \\ -2 & 1 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leqslant \begin{bmatrix} 33 \\ 8 \\ 5 \\ -1 \\ 8 \end{bmatrix}$$

Derive the dual linear program using Lagrange duality.

7.7 Consider the quadratic program illustrated in Figure 7.1,

$$\min_{\boldsymbol{x}\in\mathbb{R}^2} \frac{1}{2}\begin{bmatrix}x_1\\x_2\end{bmatrix}^\top \begin{bmatrix}2 & 1\\1 & 4\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} + \begin{bmatrix}5\\3\end{bmatrix}^\top \begin{bmatrix}x_1\\x_2\end{bmatrix}$$

$$\text{subject to} \quad \begin{bmatrix}1 & 0\\-1 & 0\\0 & 1\\0 & -1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} \leqslant \begin{bmatrix}1\\1\\1\\1\end{bmatrix}$$

Derive the dual quadratic program using Lagrange duality.

7.8 Consider the following convex optimization problem

$$\min_{\boldsymbol{w}\in\mathbb{R}^D} \quad \frac{1}{2}\boldsymbol{w}^\top\boldsymbol{w}$$

$$\text{subject to} \quad \boldsymbol{w}^\top\boldsymbol{x} \geqslant 1\,.$$

Derive the Lagrangian dual by introducing the Lagrange multiplier λ.

7.9 Consider the negative entropy of $\boldsymbol{x}\in\mathbb{R}^D$,

$$f(\boldsymbol{x}) = \sum_{d=1}^{D} x_d \log x_d\,.$$

Derive the convex conjugate function $f^*(s)$, by assuming the standard dot product.

Hint: Take the gradient of an appropriate function and set the gradient to zero.

7.10 Consider the function

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^\top\boldsymbol{x} + c\,,$$

where $\boldsymbol{A}$ is strictly positive definite, which means that it is invertible. Derive the convex conjugate of $f(\boldsymbol{x})$.

Hint: Take the gradient of an appropriate function and set the gradient to zero.

7.11 The hinge loss (which is the loss used by the support vector machine) is given by

$$L(\alpha) = \max\{0, 1-\alpha\}\,,$$

If we are interested in applying gradient methods such as L-BFGS, and do not want to resort to subgradient methods, we need to smooth the kink in the hinge loss. Compute the convex conjugate of the hinge loss $L^*(\beta)$ where β is the dual variable. Add a ℓ_2 proximal term, and compute the conjugate of the resulting function

$$L^*(\beta) + \frac{\gamma}{2}\beta^2\,,$$

where γ is a given hyperparameter.