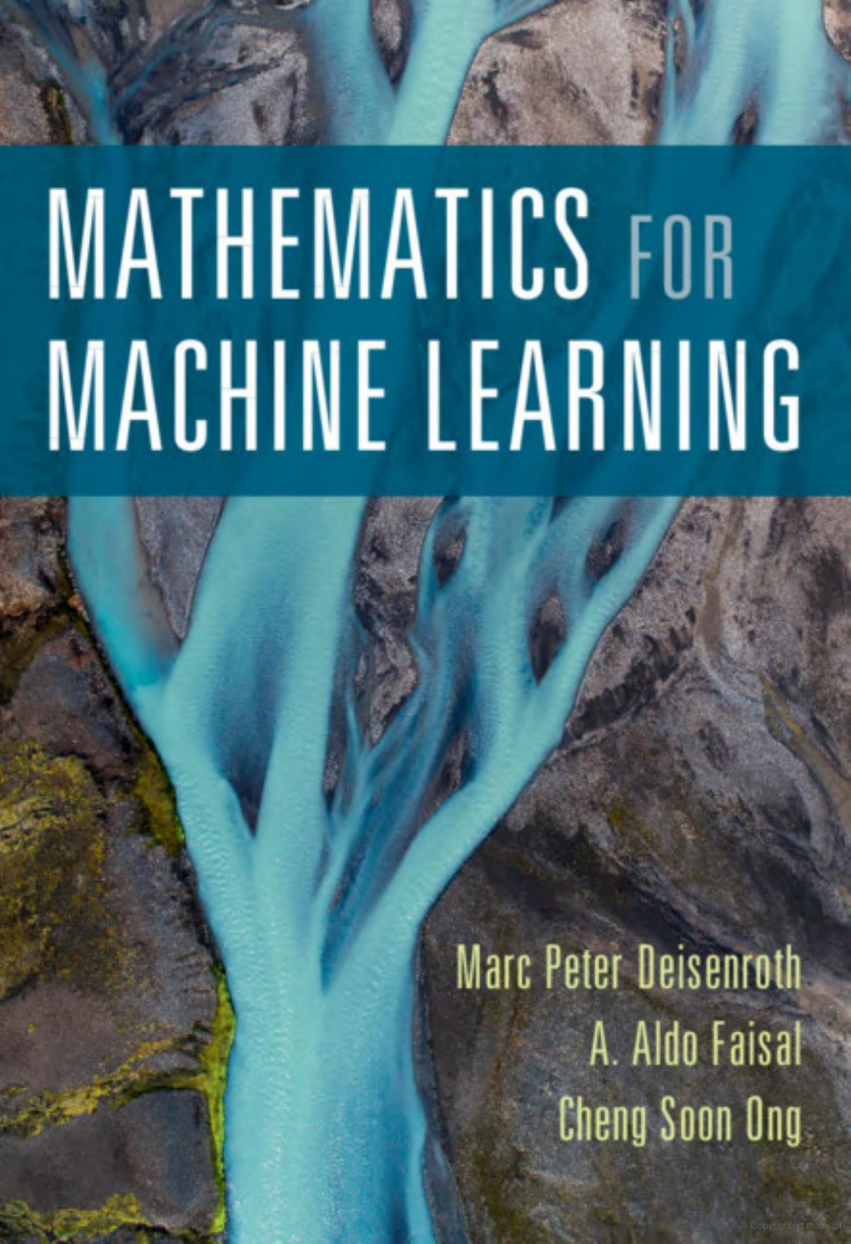


Mathematics for Machine Learning

Marc Peter Deisenroth
University College London

A. Aldo Faisal
Imperial College London

Cheng Soon Ong
Data61, CSIRO

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India

79 Anson Road, #06–04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9781108470049
DOI: 10.1017/9781108679930

First published 2020

Printed in Singapore by Markono Print Media Pte Ltd

A catalogue record for this publication is available from the British Library.

Library of Congress Cataloging-in-Publication Data
Names: Deisenroth, Marc Peter, author. | Faisal, A. Aldo, author. | Ong, Cheng Soon, author.
Title: Mathematics for machine learning / Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong.
Description: Cambridge ; New York, NY : Cambridge University Press, 2020. |
Includes bibliographical references and index.
Identifiers: LCCN 2019040762 (print) | LCCN 2019040763 (ebook) |
ISBN 9781108470049 (hardback) | ISBN 9781108455145 (paperback) | ISBN 9781108679930 (epub)
Subjects: LCSH: Machine learning–Mathematics.
Classification: LCC Q325.5 .D45 2020 (print) | LCC Q325.5 (ebook) | DDC 006.3/1–dc23
LC record available at https://lccn.loc.gov/2019040762
LC ebook record available at https://lccn.loc.gov/2019040763

ISBN 978-1-108-47004-9 Hardback
ISBN 978-1-108-45514-5 Paperback

Additional resources for this publication at https://mml-book.com.

Contents

List of Symbols ix
Preface xi
Acknowledgments xv

Part I Mathematical Foundations

1 Introduction and Motivation 3
1.1 Finding Words for Intuitions 3
1.2 Two Ways to Read This Book 5
1.3 Exercises and Feedback 7

2 Linear Algebra 8
2.1 Systems of Linear Equations 10
2.2 Matrices 12
2.3 Solving Systems of Linear Equations 17
2.4 Vector Spaces 24
2.5 Linear Independence 29
2.6 Basis and Rank 33
2.7 Linear Mappings 36
2.8 Affine Spaces 48
2.9 Further Reading 50
Exercises 51

3 Analytic Geometry 57
3.1 Norms 58
3.2 Inner Products 59
3.3 Lengths and Distances 61
3.4 Angles and Orthogonality 63
3.5 Orthonormal Basis 65
3.6 Orthogonal Complement 65
3.7 Inner Product of Functions 66
3.8 Orthogonal Projections 67
3.9 Rotations 76
3.10 Further Reading 79
Exercises 80

4 Matrix Decompositions 82
4.1 Determinant and Trace 83
4.2 Eigenvalues and Eigenvectors 88
4.3 Cholesky Decomposition 96
4.4 Eigendecomposition and Diagonalization 98
4.5 Singular Value Decomposition 101
4.6 Matrix Approximation 111
4.7 Matrix Phylogeny 115
4.8 Further Reading 116
Exercises 118

5 Vector Calculus 120
5.1 Differentiation of Univariate Functions 122
5.2 Partial Differentiation and Gradients 126
5.3 Gradients of Vector-Valued Functions 129
5.4 Gradients of Matrices 135
5.5 Useful Identities for Computing Gradients 138
5.6 Backpropagation and Automatic Differentiation 138
5.7 Higher-Order Derivatives 143
5.8 Linearization and Multivariate Taylor Series 144
5.9 Further Reading 149
Exercises 150

6 Probability and Distributions 152
6.1 Construction of a Probability Space 152
6.2 Discrete and Continuous Probabilities 157
6.3 Sum Rule, Product Rule, and Bayes' Theorem 163
6.4 Summary Statistics and Independence 165
6.5 Gaussian Distribution 175
6.6 Conjugacy and the Exponential Family 182
6.7 Change of Variables/Inverse Transform 191
6.8 Further Reading 197
Exercises 198

7 Continuous Optimization 201
7.1 Optimization Using Gradient Descent 203
7.2 Constrained Optimization and Lagrange Multipliers 208
7.3 Convex Optimization 211
7.4 Further Reading 220
Exercises 221

Part II Central Machine Learning Problems

8 When Models Meet Data 225
8.1 Data, Models, and Learning 225
8.2 Empirical Risk Minimization 232

8.3 Parameter Estimation 238
8.4 Probabilistic Modeling and Inference 244
8.5 Directed Graphical Models 249
8.6 Model Selection 254

9 Linear Regression 260
9.1 Problem Formulation 261
9.2 Parameter Estimation 263
9.3 Bayesian Linear Regression 273
9.4 Maximum Likelihood as Orthogonal Projection 282
9.5 Further Reading 283

10 Dimensionality Reduction with Principal Component Analysis 286
10.1 Problem Setting 286
10.2 Maximum Variance Perspective 289
10.3 Projection Perspective 293
10.4 Eigenvector Computation and Low-Rank Approximations 300
10.5 PCA in High Dimensions 302
10.6 Key Steps of PCA in Practice 303
10.7 Latent Variable Perspective 306
10.8 Further Reading 310

11 Density Estimation with Gaussian Mixture Models 314
11.1 Gaussian Mixture Model 315
11.2 Parameter Learning via Maximum Likelihood 316
11.3 EM Algorithm 325
11.4 Latent-Variable Perspective 328
11.5 Further Reading 332

12 Classification with Support Vector Machines 335
12.1 Separating Hyperplanes 337
12.2 Primal Support Vector Machine 338
12.3 Dual Support Vector Machine 347
12.4 Kernels 351
12.5 Numerical Solution 353
12.6 Further Reading 355

References 357
Index 367

List of Symbols

Symbol	Typical meaning
$a, b, c, \alpha, \beta, \gamma$	Scalars are lowercase
$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$	Vectors are bold lowercase
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$	Matrices are bold uppercase
$\boldsymbol{x}^\top, \boldsymbol{A}^\top$	Transpose of a vector or matrix
$\boldsymbol{A}^{-1}$	Inverse of a matrix
$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$	Inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$
$\boldsymbol{x}^\top \boldsymbol{y}$	Dot product of $\boldsymbol{x}$ and $\boldsymbol{y}$
$B = (\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3)$	(Ordered) tuple
$\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3]$	Matrix of column vectors stacked horizontally
$\mathcal{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3\}$	Set of vectors (unordered)
$\mathbb{Z}, \mathbb{N}$	Integers and natural numbers, respectively
$\mathbb{R}, \mathbb{C}$	Real and complex numbers, respectively
$\mathbb{R}^n$	n-dimensional vector space of real numbers
$\forall x$	Universal quantifier: for all x
$\exists x$	Existential quantifier: there exists x
$a := b$	a is defined as b
$a =: b$	b is defined as a
$a \propto b$	a is proportional to b, i.e., $a = \text{constant} \cdot b$
$g \circ f$	Function composition: "g after f"
$\iff$	If and only if
$\implies$	Implies
$\mathcal{A}, \mathcal{C}$	Sets
$a \in \mathcal{A}$	a is an element of the set $\mathcal{A}$
$\emptyset$	Empty set
D	Number of dimensions; indexed by $d = 1, \ldots, D$
N	Number of data points; indexed by $n = 1, \ldots, N$
$\boldsymbol{I}_m$	Identity matrix of size $m \times m$
$\boldsymbol{0}_{m,n}$	Matrix of zeros of size $m \times n$
$\boldsymbol{1}_{m,n}$	Matrix of ones of size $m \times n$
$\boldsymbol{e}_i$	Standard/canonical vector (where i is the component that is 1)
$\dim(\mathrm{V})$	Dimensionality of vector space V

Symbol	Typical meaning
$\mathrm{rk}(\boldsymbol{A})$	Rank of matrix $\boldsymbol{A}$
$\mathrm{Im}(\Phi)$	Image of linear mapping Φ
$\ker(\Phi)$	Kernel (null space) of a linear mapping Φ
$\mathrm{span}[\boldsymbol{b}_1]$	Span (generating set) of $\boldsymbol{b}_1$
$\mathrm{tr}(\boldsymbol{A})$	Trace of $\boldsymbol{A}$
$\det(\boldsymbol{A})$	Determinant of $\boldsymbol{A}$
$\lvert \cdot \rvert$	Absolute value or determinant (depending on context)
$\lVert \cdot \rVert$	Norm; Euclidean unless specified
λ	Eigenvalue or Lagrange multiplier
E_λ	Eigenspace corresponding to eigenvalue λ
$\boldsymbol{\theta}$	Parameter vector
$\frac{\partial f}{\partial x}$	Partial derivative of f with respect to x
$\frac{\mathrm{d}f}{\mathrm{d}x}$	Total derivative of f with respect to x
∇	Gradient
$\mathfrak{L}$	Lagrangian
$\mathcal{L}$	Negative log-likelihood
$\binom{n}{k}$	Binomial coefficient, n choose k
$\mathbb{V}_X[\boldsymbol{x}]$	Variance of $\boldsymbol{x}$ with respect to the random variable X
$\mathbb{E}_X[\boldsymbol{x}]$	Expectation of $\boldsymbol{x}$ with respect to the random variable X
$\mathrm{Cov}_{X,Y}[\boldsymbol{x}, \boldsymbol{y}]$	Covariance between $\boldsymbol{x}$ and $\boldsymbol{y}$.
$X \perp\!\!\!\perp Y \mid Z$	X is conditionally independent of Y given Z
$X \sim p$	Random variable X is distributed according to p
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\mathrm{Ber}(\mu)$	Bernoulli distribution with parameter μ
$\mathrm{Bin}(N, \mu)$	Binomial distribution with parameters N, μ
$\mathrm{Beta}(\alpha, \beta)$	Beta distribution with parameters α, β

List of Abbreviations and Acronyms

Acronym	Meaning
e.g.	Exempli gratia (Latin: for example)
GMM	Gaussian mixture model
i.e.	Id est (Latin: this means)
i.i.d.	Independent, identically distributed
MAP	Maximum a posteriori
MLE	Maximum likelihood estimation/estimator
ONB	Orthonormal basis
PCA	Principal component analysis
PPCA	Probabilistic principal component analysis
REF	Row-echelon form
SPD	Symmetric, positive definite
SVM	Support vector machine

target audience of the book includes undergraduate university students, evening learners and learners participating in online machine learning courses.

In analogy to music, there are three types of interaction that people have with machine learning:

Astute Listener The democratization of machine learning by the provision of open-source software, online tutorials and cloud-based tools allows users to not worry about the specifics of pipelines. Users can focus on extracting insights from data using off-the-shelf tools. This enables non-tech-savvy domain experts to benefit from machine learning. This is similar to listening to music; the user is able to choose and discern between different types of machine learning, and benefits from it. More experienced users are like music critics, asking important questions about the application of machine learning in society such as ethics, fairness and privacy of the individual. We hope that this book provides a foundation for thinking about the certification and risk management of machine learning systems and allows them to use their domain expertise to build better machine learning systems.

Experienced Artist Skilled practitioners of machine learning can plug and play different tools and libraries into an analysis pipeline. The stereotypical practitioner would be a data scientist or engineer who understands machine learning interfaces and their use cases and is able to perform wonderful feats of prediction from data. This is similar to a virtuoso playing music, where highly skilled practitioners can bring existing instruments to life and bring enjoyment to their audience. Using the mathematics presented here as a primer, practitioners would be able to understand the benefits and limits of their favourite method, and to extend and generalize existing machine learning algorithms. We hope that this book provides the impetus for more rigorous and principled development of machine learning methods.

Fledgling Composer As machine learning is applied to new domains, developers of machine learning need to develop new methods and extend existing algorithms. They are often researchers who need to understand the mathematical basis of machine learning and uncover relationships between different tasks. This is similar to composers of music who, within the rules and structure of musical theory, create new and amazing pieces. We hope this book provides a high-level overview of other technical books for people who want to become composers of machine learning. There is a great need in society for new researchers who are able to propose and explore novel approaches for attacking the many challenges of learning from data.

Acknowledgments

We are grateful to many people who looked at early drafts of the book and suffered through painful expositions of concepts. We tried to implement their ideas that we did not vehemently disagree with. We would like to especially acknowledge Christfried Webers for his careful reading of many parts of the book, and his detailed suggestions on structure and presentation. Many friends and colleagues have also been kind enough to provide their time and energy on different versions of each chapter. We have been lucky to benefit from the generosity of the online community, who have suggested improvements via github.com, which greatly improved the book.

The following people have found bugs, proposed clarifications and suggested relevant literature, either via github.com or personal communication. Their names are sorted alphabetically.

Abdul-Ganiy Usman
Adam Gaier
Adele Jackson
Aditya Menon
Alasdair Tran
Aleksandar Krnjaic
Alexander Makrigiorgos
Alfredo Canziani
Ali Shafti
Amr Khalifa
Andrew Tanggara
Angus Gruen
Antal A. Buss
Antoine Toisoul Le Cann
Areg Sarvazyan
Artem Artemev
Artyom Stepanov
Bill Kromydas
Bob Williamson
Boon Ping Lim
Chao Qu
Cheng Li
Chris Sherlock
Christopher Gray
Daniel McNamara
Daniel Wood
Darren Siegel
David Johnston
Dawei Chen
Ellen Broad
Fengkuangtian Zhu
Fiona Condon
Georgios Theodorou
He Xin
Irene Raissa Kameni
Jakub Nabaglo
James Hensman
Jamie Liu
Jean Kaddour
Jean-Paul Ebejer
Jerry Qiang
Jitesh Sindhare
John Lloyd
Jonas Ngnawe
Jon Martin
Justin Hsi

Kai Arulkumaran
Kamil Dreczkowski
Lily Wang
Lionel Tondji Ngoupeyou
Lydia Knüfing
Mahmoud Aslan
Mark Hartenstein
Mark van der Wilk
Markus Hegland
Martin Hewing
Matthew Alger
Matthew Lee
Maximus McCann
Mengyan Zhang
Michael Bennett
Michael Pedersen
Minjeong Shin
Mohammad Malekzadeh
Naveen Kumar
Nico Montali
Oscar Armas
Patrick Henriksen
Patrick Wieschollek
Pattarawat Chormai
Paul Kelly
Petros Christodoulou
Piotr Januszewski
Pranav Subramani
Quyu Kong
Ragib Zaman
Rui Zhang
Ryan-Rhys Griffiths
Salomon Kabongo
Samuel Ogunmola
Sandeep Mavadia
Sarvesh Nikumbh
Sebastian Raschka
Senanayak Sesh Kumar Karri
Seung-Heon Baek
Shahbaz Chaudhary
Shakir Mohamed
Shawn Berry
Sheikh Abdul Raheem Ali
Sheng Xue
Sridhar Thiagarajan
Syed Nouman Hasany
Szymon Brych
Thomas Bühler
Timur Sharapov
Tom Melamed
Vincent Adam
Vincent Dutordoir
Vu Minh
Wasim Aftab
Wen Zhi
Wojciech Stokowiec
Xiaonan Chong
Xiaowei Zhang
Yazhou Hao
Yicheng Luo
Young Lee
Yu Lu
Yun Cheng
Yuxiao Huang
Zac Cranko
Zijian Cao
Zoe Nolan

Contributors through github, whose real names were not listed on their github profile, are the following:

SamDataMad	insad	empet
bumptiousmonkey	HorizonP	victorBigand
idoamihai	cs-maillist	17SKYE
deepakiim	kudo23	jessjing1995

We are also very grateful to Parameswaran Raman and the many anonymous reviewers, organized by Cambridge University Press, who read one or more

chapters of earlier versions of the manuscript, and provided constructive criticism that led to considerable improvements. A special mention goes to Dinesh Singh Negi, our LaTeX support for detailed and prompt advice about LaTeX-related issues. Last but not least, we are very grateful to our editor Lauren Cowles, who has been patiently guiding us through the gestation process of this book.

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 *predictor*.

predictor

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.

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

model

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 them. A good model can then be used to predict what would happen in the real world without performing real-world experiments.

learning

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 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 1.1: 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 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 1.1) 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

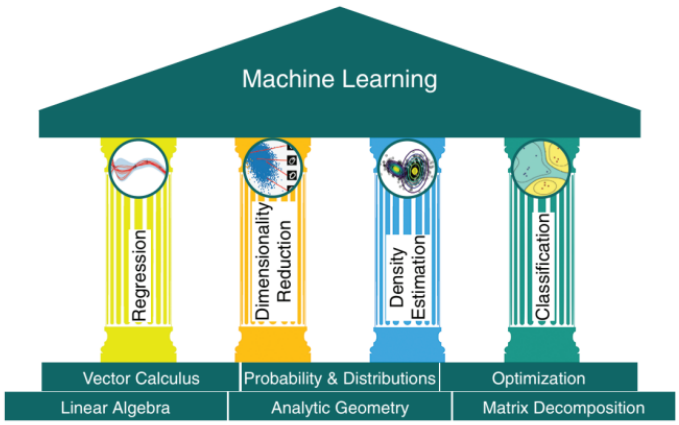


Figure 1.1 The foundations and four pillars of machine learning.

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 1.1. 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 setups 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

linear regression

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 $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ componentwise 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.

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.

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`

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 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,

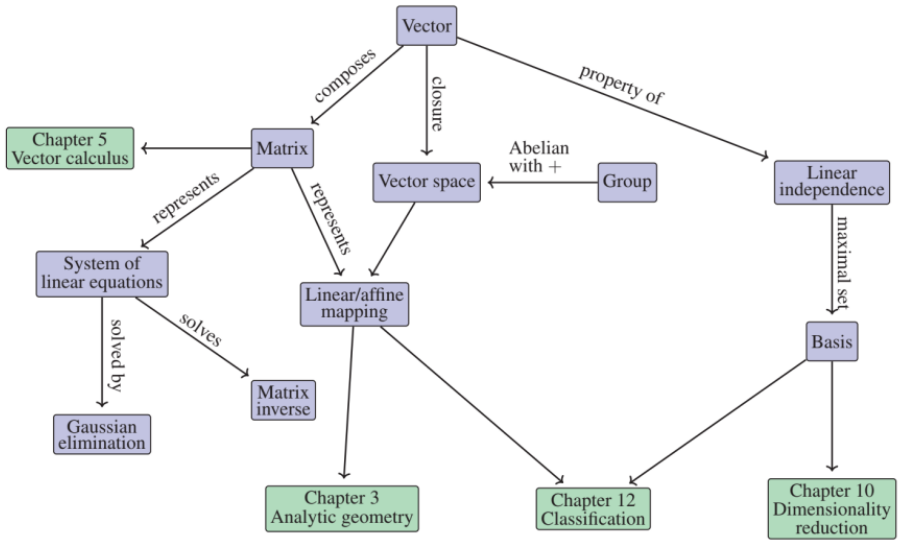


Figure 2.2 A mind map of the concepts introduced in this chapter, along with where they are used in other parts of the book.

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 $b_i \in \mathbb{R}$.

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.

system of linear equations

solution

Example 2.2
The system of linear equations

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

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

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

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

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

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

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

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 2.3 for the system

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

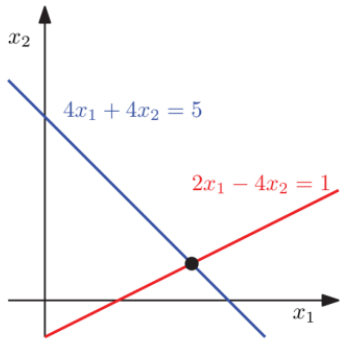


Figure 2.3 The solution space of a system of two linear equations with two variables can be geometrically interpreted as the intersection of two lines. Every linear equation represents a line.

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). ♢

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:

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

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

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:

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

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

row
column
row vector
column vector

$\mathbb{R}^{m\times n}$ 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 2.4.

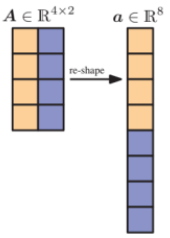


Figure 2.4 By stacking its columns, a matrix $\boldsymbol{A}$ can be represented as a long vector $\boldsymbol{a}$.

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.,

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

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 $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B} \in \mathbb{R}^{m\times k}$ are computed as

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

Note the size of the matrices.

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

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 "$\cdot$").

There are n columns in $\boldsymbol{A}$ and n rows in $\boldsymbol{B}$ so that we can compute $a_{il}b_{lj}$ for $l = 1, \ldots, n$.

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 $\boldsymbol{B}\boldsymbol{A}$ is not defined if $m \neq n$ since the neighboring dimensions do not match. ♢

Remark. Matrix multiplication is *not* defined as an elementwise 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 elementwise multiplication often appears in programming languages when we multiply (multidimensional) 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

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

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

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

distributivity

- $(\lambda\boldsymbol{C})^\top = \boldsymbol{C}^\top \lambda^\top = \boldsymbol{C}^\top \lambda = \lambda \boldsymbol{C}^\top$ 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 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$; see (2.3), and the product $\boldsymbol{A}\boldsymbol{x}$ 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, $i = 1, \ldots, m$, $j = 1, \ldots, n$. Thus far, we saw that matrices can be used as a compact way of formulating systems of linear equations so that we can write $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$; 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 $x_1, \ldots, x_4$, 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 nontrivial 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 nontrivial versions of $\mathbf{0}$ as

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

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

general solution

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

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. ♢

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 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. We no longer mention the variables $\boldsymbol{x}$ explicitly and build the *augmented matrix* (in the form $\left[\boldsymbol{A} \,|\, \boldsymbol{b}\right]$)

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 $\left[\boldsymbol{A} \,|\, \boldsymbol{b}\right]$ compactly represents the system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.

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

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

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

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

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

row-echelon form

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

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

particular solution

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

general solution

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

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

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.

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

basic variable
free variable

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

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

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 nonpivot columns for which we set the coefficients implicitly to 0. Therefore, we get the particular solution $\boldsymbol{x} = [2, 0, -1, 1, 0]^\top$. ♢

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 $\boldsymbol{AX} = \boldsymbol{I}_n$. Then $\boldsymbol{X} = \boldsymbol{A}^{-1}$. We can write this down as a set of simultaneous linear equations $\boldsymbol{AX} = \boldsymbol{I}_n$, 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

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

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 $\boldsymbol{AA}^{-1}$ 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 $\boldsymbol{Ax} = \boldsymbol{b}$. 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

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

Moore–Penrose pseudo-inverse

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.

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 overrelaxation 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.

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: group

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

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

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

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

- $(\mathbb{Z}, +)$ is a group.
- $(\mathbb{N}_0, +)$ is not a group: Although $(\mathbb{N}_0, +)$ possesses a neutral element (0), the inverse elements are missing. $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$
- $(\mathbb{Z}, \cdot)$ 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.

vector space

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

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

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

where

1. $(\mathcal{V}, +)$ is an Abelian group
2. Distributivity:
   a. $\forall \lambda \in \mathbb{R}, \boldsymbol{x}, \boldsymbol{y} \in \mathcal{V} : \lambda \cdot (\boldsymbol{x} + \boldsymbol{y}) = \lambda \cdot \boldsymbol{x} + \lambda \cdot \boldsymbol{y}$
   b. $\forall \lambda, \psi \in \mathbb{R}, \boldsymbol{x} \in \mathcal{V} : (\lambda + \psi) \cdot \boldsymbol{x} = \lambda \cdot \boldsymbol{x} + \psi \cdot \boldsymbol{x}$
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: $\forall \boldsymbol{x} \in \mathcal{V} : 1 \cdot \boldsymbol{x} = \boldsymbol{x}$

vector
vector addition
scalar multiplication by scalars

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.

Remark. A "vector multiplication" $\boldsymbol{ab}$, $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, is not defined. Theoretically, we could define an elementwise 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). ♢

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: $\boldsymbol{x}+\boldsymbol{y} = (x_1,\dots,x_n)+(y_1,\dots,y_n) = (x_1+y_1,\dots,x_n+y_n)$ for all $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n$
  - Multiplication by scalars: $\lambda\boldsymbol{x} = \lambda(x_1,\dots,x_n) = (\lambda x_1,\dots,\lambda x_n)$ 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. ♢

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 $\boldsymbol{x}$. ♢

row vector

transpose

and vice versa. However, the third "751 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 km West" and "374 km Southwest" can be linearly combined to obtain "506 km Northwest".

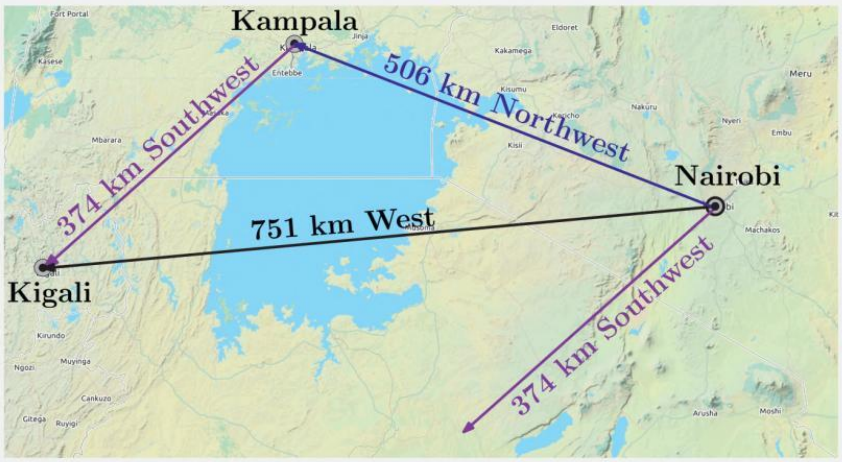


Figure 2.7 Geographic example (with crude approximations to cardinal directions) of linearly dependent vectors in a two-dimensional space (plane).

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 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ is $\mathbf{0}$ then they are linearly dependent. The same holds if two vectors are identical.
- The vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k : \boldsymbol{x}_i \neq \mathbf{0}, i = 1, \ldots, k\}$, $k \geqslant 2$, 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 $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k : \boldsymbol{x}_i \neq \mathbf{0}, i = 1, \ldots, k\}$ is linearly dependent.
- A practical way of checking whether vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in V$ 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 nonpivot 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 nonpivot 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 nonpivot column, the columns (and, therefore, the corresponding vectors) are linearly dependent.

♢

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

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

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

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

in a more compact form.

We want to test whether $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ 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 $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ are linearly independent if and only if the column vectors $\{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_m\}$ are linearly independent.

♢

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

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

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

Are the vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_4 \in \mathbb{R}^n$ 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 nontrivially 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, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_4$ are linearly dependent as $\boldsymbol{x}_4$ can be expressed as a linear combination of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_3$.

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 $\mathcal{A} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\} \subseteq \mathcal{V}$. If every vector $\boldsymbol{v} \in \mathcal{V}$ can be expressed as a linear combination of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$, $\mathcal{A}$ 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 $V = \text{span}[\mathcal{A}]$ or $V = \text{span}[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k]$.

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 $\tilde{\mathcal{A}} \subseteq \mathcal{A} \subseteq \mathcal{V}$ 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

canonical basis

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

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

- 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*.

basis vector

♢

dimension

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 $\dim(U) = \dim(V)$ 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. ♢

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.

♢

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

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

linear mapping
vector space homomorphism
linear transformation

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

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

♢

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 $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$. ♢

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.

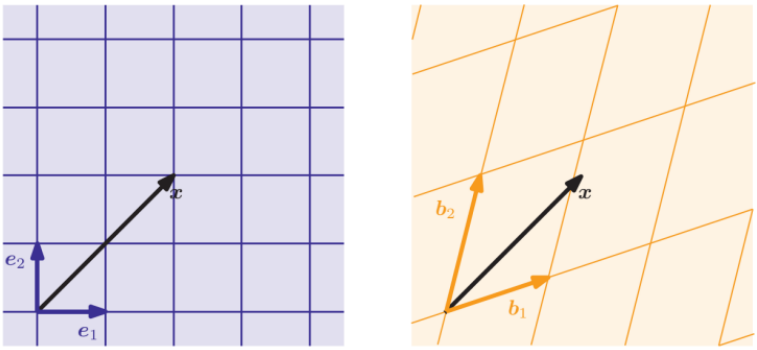


Figure 2.8 Two different coordinate systems defined by two sets of basis vectors. A vector $\boldsymbol{x}$ has different coordinate representations depending on which coordinate system is chosen.

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 2.8, 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 2.9).

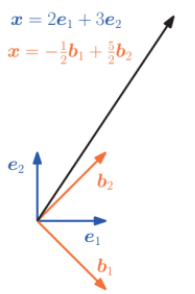


Figure 2.9 Different coordinate representations of a vector $\boldsymbol{x}$, depending on the choice of basis.

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 $\mathbb{R}^n$. ♢

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

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

transformation matrix

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

The coordinates of $\Phi(\boldsymbol{b}_j)$ with respect to the ordered basis C of W are the jth 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

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

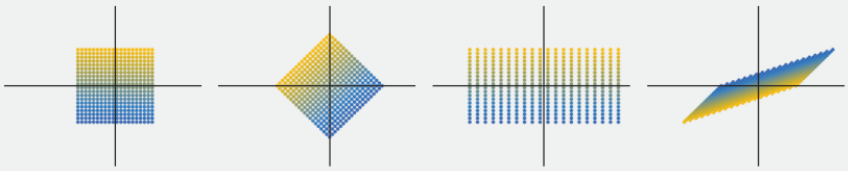
where the $\boldsymbol{\alpha}_j,\ j = 1, 2, 3$, are the coordinate vectors of $\Phi(\boldsymbol{b}_j)$ with respect to C.

Example 2.22 (Linear Transformations of Vectors)
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 2.10 gives three examples of linear transformations of a set of vectors. Figure 2.10(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 2.10(b). If we

apply the linear mapping represented by $\boldsymbol{A}_2$, we obtain the rectangle in Figure 2.10(c) where each x_1-coordinate is stretched by 2. Figure 2.10(d) shows the original square from Figure 2.10(a) when linearly transformed using $\boldsymbol{A}_3$, which is a combination of a reflection, a rotation, and a stretch.



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

Figure 2.10 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; and (d) combination of reflection, rotation, and stretching.

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

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

of V and two ordered bases

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

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 2.9, 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

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