

《机器学习数学基础》



MATHEMATICS FOR



MACHINE LEARNING



Marc Peter Deisenroth

A. Aldo Faisal

Cheng Soon Ong

CAMBRIDGE



目录

前言 1

第一部分 数学基础 9

1 介绍与动机 11 1.1 为直觉寻找语言 12 1.2 阅读本书的两种方式 13 1.3 练习与反馈 16

2 线性代数 17 2.1 线性方程组 19 2.2 矩阵 22 2.3 求解线性方程组 27 2.4 向量空间 35 2.5 线性无关 40 2.6 基与秩 44 2.7 线性映射 48 2.8 仿射空间 61 2.9 延伸阅读 63 练习 64

3 解析几何 70 3.1 范数 71 3.2 内积 72 3.3 长度与距离 75 3.4 角度与正交性 76 3.5 标准正交基 78 3.6 正交补 79 3.7 函数的内积 80 3.8 正交投影 81 3.9 旋转 91 3.10 延伸阅读 94 练习 96

4 矩阵分解 98 4.1 行列式与迹 99

i 本资料由 Cambridge University Press 出版，书名为《Mathematics for Machine Learning》，作者为 Marc Peter Deisenroth、A. Aldo Faisal 和 Cheng Soon Ong (2020)。此版本免费供个人使用查看和下载。不得重新分发、转售或用于衍生作品。©by M. P. Deisenroth, A. A. Faisal, and C. S. Ong, 2023. <https://mml-book.com>

ii 目录

4.2 特征值与特征向量 105 4.3 Cholesky 分解 114 4.4 特征分解与对角化 115 4.5 奇异值分解 119 4.6 矩阵近似 129 4.7 矩阵系统发生学 134 4.8 延伸阅读 135 练习 137

5 向量微积分 139 5.1 单变量函数的微分 141 5.2 偏微分与梯度 146 5.3 向量值函数的梯度 149 5.4 矩阵的梯度 155 5.5 计算梯度的有用恒等式 158 5.6 反向传播与自动微分 159 5.7 高阶导数 164 5.8 线性化与多变量 Taylor 级数 165 5.9 延伸阅读 170 练习 170

6 概率与分布 172 6.1 概率空间的构造 172 6.2 离散与连续概率 178 6.3 和法则、积法则与 Bayes 定理 183 6.4 汇总统计与独立性 186 6.5 Gaussian 分布 197 6.6 共轭性与指数族 205 6.7 变量变换/逆变换 214 6.8 延伸阅读 221 练习 222

7 连续优化 225 7.1 使用梯度下降的优化 227 7.2 约束优化与 Lagrange 乘数 233 7.3 凸优化 236 7.4 延伸阅读 246 练习 247

第二部分 Machine Learning 核心问题 249

8 当模型遇到数据 251 8.1 数据、模型与学习 251 8.2 经验风险最小化 258 8.3 参数估计 265 8.4 概率建模与推断 272 8.5 有向图模型 278

《Mathematics for Machine Learning》草稿 (2023-10-18)。反馈：<https://mml-book.com> 目录 iii

8.6 模型选择 283

9 线性回归 289 9.1 问题表述 291 9.2 参数估计 292 9.3 Bayesian 线性回归 303 9.4 最大似然作为正交投影 313 9.5 延伸阅读 315

10 主成分分析降维 317 10.1 问题设定 318 10.2 最大方差视角 320 10.3 投影视角 325 10.4 特征向量计算与低秩近似 333 10.5 高维PCA 335 10.6 PCA实践中的关键步骤 336 10.7 潜变量视角 339 10.8 延伸阅读 343

11 Gaussian混合模型密度估计 348 11.1 Gaussian混合模型 349 11.2 通过最大似然进行参数学习 350 11.3 EM算法 360 11.4 潜变量视角 363 11.5 延伸阅读 368

12 支持向量机分类 370 12.1 分离超平面 372 12.2 原始支持向量机 374 12.3 对偶支持向量机 383 12.4 核 388 12.5 数值解 390 12.6 延伸阅读 392

参考文献 395 **索引** 407

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由Cambridge University Press出版 (2020)。

前言

Machine learning是人类尝试将人类知识和推理提炼成适合构建机器人和工程自动化系统形式的最新尝试。随着machine learning变得更加无处不在，其软件包变得更易于使用，将低级技术细节抽象化并对从业者隐藏是自然且理想的。然而，这带来了从业者可能会忽略设计决策因而忽略machine learning算法局限性的危险。

对于有兴趣了解成功machine learning算法背后魔力的热忱从业者，目前面临着一系列令人望而生畏的先决知识：

编程语言和数据分析工具

大规模计算及其相关框架

数学和统计学以及machine learning如何建立在其基础之上

在大学中，machine learning入门课程往往在课程的早期部分涵盖这些先决条件中的一些。由于历史原因，machine learning课程通常在计算机科学系教授，那里的学生通常在前两个知识领域接受培训，但在数学和统计学方面则不那么多。

当前的machine learning教科书主要关注machine learning算法和方法论，并假设读者在数学和统计学方面是有能力的。因此，这些书籍只花一两章介绍背景数学，要么在书的开头，要么作为附录。我们发现许多想要深入研究基本machine learning方法基础的人在阅读machine learning教科书所需的数学知识方面存在困难。在大学教授本科和研究生课程后，我们发现高中数学与阅读标准machine learning教科书所需数学水平之间的差距对许多人来说太大了。

本书将基本machine learning概念的数学基础提到前台，并将信息收集在一个地方，以便缩小甚至消除这种技能差距。

[本材料由Cambridge University Press出版，书名为] [*Mathematics for Machine Learning*] [作者为] [Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong (2020). 此版本仅供个人使用免费查看] [和下载。不得重新分发、转售或用于衍生作品。]

[©by M. P. Deisenroth, A. A. Faisal, and C. S. Ong, 2023.] [<https://mml-book.com>]

前言

为什么又要写一本关于Machine Learning的书?

Machine learning建立在数学语言之上来表达那些看似直观明显但却难以形式化的概念。一旦正确形式化, 我们就能深入了解我们想要解决的任务。全球数学学生的一个常见抱怨是, 所涵盖的主题似乎与实际问题的几乎没有关联。我们相信machine learning是人们学习数学的一个明显而直接的动机。

[“数学在大众心理中”][本书旨在成为构成现代machine learning基础的庞大数学文]

[与恐惧症和][献的指南书。我们通过直接指出数学概念在基本]

[焦虑联系在一起。你会][machine learning问题背景下的有用性来激发对数学概念的需求。为了]

[认为我们在讨论]

[蜘蛛。”][Strogatz,][保持本书简短, 许多细节和更高级的概念被省略了。配备了这里]

[2014,] 第281页][呈现的基本概念, 以及它们如何融入machine learning的更大背景,]

读者可以找到许多进一步学习的资源, 我们在相应章节的末尾提供了这些资源。对于有数学背景的读者, 本书提供了machine learning的简要但精确的概述。与其他专注于machine learning方法和模型 [(MacKay, 2003;] [Bishop, 2006;] [Alpaydin, 2010; Bar-] [ber, 2012;] [Murphy, 2012;] [Shalev-Shwartz and Ben-David, 2014;] [Rogers] [and Girolami, 2016)] 或machine learning编程方面 [(Müller] [and Guido, 2016;] [Raschka and Mirjalili, 2017;] [Chollet and Allaire, 2018)] 的书籍不同, 我们只提供四个machine learning算法的代表性例子。相反, 我们专注于模型本身背后的数学概念。我们希望读者能够更深入地理解machine learning中的基本问题, 并将使用machine learning产生的实际问题与数学模型中的基本选择联系起来。

我们的目标不是写一本经典的machine learning书籍。相反, 我们的意图是提供数学背景, 应用于四个核心machine learning问题, 以便更容易阅读其他machine learning教科书。

目标读者是谁?

随着machine learning的应用在社会中变得广泛, 我们相信每个人都应该对其基本原理有一些理解。本书以学术数学风格编写, 这使我们能够对machine learning背后的概念保持精确。我们鼓励不熟悉这种看似简洁风格的读者坚持下去, 并牢记每个主题的目标。我们在整个文本中穿插评论和备注, 希望能为大局提供有用的指导。

本书假设读者具有通常的数学知识

[草稿 (2023-10-18) “Mathematics for Machine Learning” . 反馈:] [<https://mml-book.com>] [前言]

已在高中数学和物理中涵盖。例如, 读者应该之前接触过导数和积分, 以及二维或三维的几何向量。从这些基础开始, 我们将推广这些概念。因此, 本书的目标读者包括大学本科生、夜校学习者和参加在线机器学习课程的学习者。

以音乐为类比，人们与机器学习有三种类型的互动：

敏锐的听众 [机器学习的民主化通过开源软件、在线教程和基于云的工具的提供，让用户无需担心管道的具体细节。用户可以专注于使用现成工具从数据中提取见解。这使得非技术领域专家能够从机器学习中受益。这类似于听音乐；用户能够在不同类型的机器学习之间进行选择和辨别，并从中受益。更有经验的用户就像音乐评论家，对机器学习在社会中的应用提出重要问题，如伦理、公平性和个人隐私。我们希望这本书为思考机器学习系统的认证和风险管理提供基础，并允许他们使用自己的领域专业知识来构建更好的机器学习系统。]

经验丰富的艺术家 [机器学习的熟练从业者可以将不同的工具和库插入到分析管道中。典型的从业者是数据科学家或工程师，他们理解机器学习接口及其用例，能够从数据中进行出色的预测。这类似于演奏音乐的大师，高技能的从业者可以让现有乐器焕发生机，为观众带来享受。使用这里提供的数学作为入门，从业者将能够理解他们喜爱方法的优点和局限性，并扩展和推广现有的机器学习算法。我们希望这本书能为更严格和有原则的机器学习方法开发提供推动力。]

初出茅庐的作曲家 [随着机器学习应用到新领域，机器学习的开发者需要开发新方法并扩展现有算法。他们通常是需要理解机器学习数学基础并揭示不同任务之间关系的研究人员。这类似于音乐作曲家，在音乐理论的规则和结构内，创作新的令人惊叹的作品。我们希望这本书为想要成为机器学习作曲家的人提供其他技术书籍的高级概述。社会非常需要能够提出和探索新方法以应对从数据中学习的诸多挑战的新研究人员。]

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由剑桥大学出版社出版 (2020).]

前言

致谢

我们感谢许多人查看了本书的早期草稿，并忍受了痛苦的概念阐述。我们试图实施他们提出的我们并不强烈反对的想法。我们要特别感谢Christfried Webers仔细阅读了本书的许多部分，以及他对结构和呈现的详细建议。许多朋友和同事也慷慨地为每章的不同版本提供了他们的时间和精力。我们很幸运地受益于在线社区的慷慨，他们通过<https://github.com>提出了改进建议，这大大改进了这本书。

以下人员发现了错误，提出了澄清建议并推荐了相关文献，无论是通过<https://github.com>还是个人交流。他们的姓名按字母顺序排列。

Abdul-Ganiy Usman Ellen Broad Adam Gaier Fengkuangtian Zhu Adele Jackson Fiona Condon Aditya Menon Georgios Theodorou Alasdair Tran He Xin Aleksandar Krnjaic Irene Raissa Kameni Alexander Makrigiorgos Jakub Nabaglo Alfredo Canziani James Hensman Ali Shafiti Jamie Liu Amr Khalifa Jean Kaddour Andrew Tanggara Jean-Paul Ebejer Angus Gruen Jerry Qiang Antal A. Buss Jitesh Sindhare Antoine Toisoul Le Cann John Lloyd Areg Sarvazyan Jonas Ngnawe Artem Artemev Jon Martin Artyom Stepanov Justin Hsi Bill Kromydas Kai Arulkumaran Bob Williamson Kamil Dreczkowski Boon Ping Lim Lily Wang Chao Qu Lionel Tondji Ngoupeyou Cheng Li Lydia Knüfing Chris Sherlock Mahmoud Aslan Christopher Gray Mark Hartenstein Daniel McNamara Mark van der Wilk Daniel Wood Markus Hegland Darren Siegel Martin Hewing David Johnston Matthew Alger Dawei Chen Matthew Lee

Maximus McCann Shakir Mohamed

Mengyan Zhang Shawn Berry

Michael Bennett Sheikh Abdul Raheem Ali

Michael Pedersen Sheng Xue

Minjeong Shin Sridhar Thiagarajan

Mohammad Malekzadeh Syed Nouman Hasany

Naveen Kumar Szymon Brych

Nico Montali Thomas Bühler

Oscar Armas Timur Sharapov

Patrick Henriksen Tom Melamed

Patrick Wiescholke

Vincent Adam

Pattarawat Chormai

Vincent Dutordoir

Paul Kelly

Vu Minh

Petros Christodoulou

Wasim Aftab

Piotr Januszewski

Wen Zhi

Pranav Subramani

Wojciech Stokowiec

Quyu Kong

Ragib Zaman Xiaonan Chong

Rui Zhang Xiaowei Zhang

Ryan-Rhys Griffiths Yazhou Hao

Salomon Kabongo Yicheng Luo

Samuel Ogunmola Young Lee

Sandeep Mavadia Yu Lu

Sarvesh Nikumbh Yun Cheng

Sebastian Raschka Yuxiao Huang

Senanayak Sesh Kumar Karri Zac Cranko

Seung-Heon Baek Zijian Cao

Shahbaz Chaudhary Zoe Nolan

通过GitHub贡献的人员，其真实姓名未在其GitHub档案中列出的有：

SamDataMad insad empet

bumptiousmonkey HorizonP victorBigand

idoamihai cs-maillist 17SKYE

deepakiim kudo23 jessjing1995

我们也非常感谢Parameswaran Raman和许多匿名审稿人，他们由剑桥大学出版社组织，阅读了早期版本手稿的一个或多个章节，并提供了建设性的批评，这些批评带来了相当大的改进。特别要提到的是Dinesh Singh Negi，我们的L^AT_EX支持，他对L^AT_EX相关问题提供了详细和及时的建议。最后但同样重要的是，我们非常感谢我们的编辑Lauren Cowles，她一直耐心地指导我们完成这本书的创作过程。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

前言

符号表

符号	典型含义
$a, b, c, \alpha, \beta, \gamma$	标量为小写字母
$\mathbf{x}, \mathbf{y}, \mathbf{z}$	向量为粗体小写字母
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	矩阵为粗体大写字母
$\mathbf{x}^T, \mathbf{A}^T$	向量或矩阵的转置
$\mathbf{A}^{-1}$	矩阵的逆
$\langle \mathbf{x}, \mathbf{y} \rangle$	$\mathbf{x}$ 和 $\mathbf{y}$ 的内积
$\mathbf{x}^T \mathbf{y}$	$\mathbf{x}$ 和 $\mathbf{y}$ 的点积
$\mathbf{B} = (b_1, b_2, b_3)$	(有序) 元组
$\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$	水平堆叠的列向量矩阵
$\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$	向量集合 (无序)
$\mathbb{Z}, \mathbb{N}$	分别为整数和自然数
$\mathbb{R}, \mathbb{C}$	分别为实数和复数
$\mathbb{R}^n$	n 维实数向量空间
$\forall \mathbf{x}$	全称量词：对于所有 $\mathbf{x}$
$\exists \mathbf{x}$	存在量词：存在 $\mathbf{x}$
$\mathbf{a} := \mathbf{b}$	$\mathbf{a}$ 定义为 $\mathbf{b}$
$\mathbf{a} \Leftarrow \mathbf{b}$	$\mathbf{b}$ 定义为 $\mathbf{a}$
$\mathbf{a} \propto \mathbf{b}$	$\mathbf{a}$ 与 $\mathbf{b}$ 成正比，即 $\mathbf{a} = \text{常数} \cdot \mathbf{b}$

符号	典型含义
$g \circ f$	函数复合：“f之后的g”
$\Leftarrow \Rightarrow$	当且仅当
$\Rightarrow$	蕴含
A, C	集合
$a \in A$	a是集合A的元素
$\emptyset$	空集
A	A去掉B: A中存在但B中不存在的元素集合
D	维数；由 $d = 1, \dots, D$ 索引
N	数据点数；由 $n = 1, \dots, N$ 索引
I_m	$m \times m$ 大小的单位矩阵
$0_{m,n}$	$m \times n$ 大小的零矩阵
$1_{m,n}$	$m \times n$ 大小的全1矩阵
e_i	标准/规范向量（其中i是值为1的分量）
$\dim$	向量空间的维数
$\text{rk}(A)$	矩阵A的秩
$\text{Im}(\Phi)$	线性映射 Φ 的像
$\text{ker}(\Phi)$	线性映射 Φ 的核（零空间）
$\text{span}[b_1]$	b_1 的张成（生成集）
$\text{tr}(A)$	A的迹
$\det(A)$	A的行列式
$ \cdot $	绝对值或行列式（取决于上下文）

符号	典型含义
$\ \cdot \ $	范数；除非另有说明，指欧几里得范数
λ	特征值或拉格朗日乘子
$E[\lambda]$	对应于特征值 λ 的特征空间

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] [<https://mml-book.com>]

符号	典型含义
$x \perp y$	向量 x 和 y 正交
V	向量空间
$V^\perp$	向量空间 V 的正交补
$\sum_{n=1}^N x_n$	x_n 的和: $x_1 + \dots + x_n$
$\prod_{n=1}^N x_n$	x_n 的积: $x_1 \cdot \dots \cdot x_n$
θ	参数向量
$\partial f / \partial x$	f 关于 x 的偏导数
df/dx	f 关于 x 的全导数
∇	梯度
$f^* = \min_x f(x)$	f 的最小函数值
$x^* \in \arg \min_x f(x)$	使 f 最小的值 x^* （注意： $\arg \min$ 返回值的集合）
L	拉格朗日函数
L	负对数似然
$\binom{n}{k}$	二项式系数， n 选 k
V_X	x 关于随机变量 X 的方差
E_X	x 关于随机变量 X 的期望

符号	典型含义
$\text{Cov}[X, Y][x, y]$	x和y之间的协方差
$X \perp\!\!\!\perp Y \mid Z$	给定Z的条件下，X与Y条件独立
$X \sim p$	随机变量X按分布p分布
$N(\mu, \Sigma)$	均值为 μ 、协方差为 Σ 的高斯分布
$\text{Ber}(\mu)$	参数为 μ 的伯努利分布
$\text{Bin}(N, \mu)$	参数为 N, μ 的二项分布
$\text{Beta}(\alpha, \beta)$	参数为 α, β 的Beta分布

缩写和首字母缩略词表

缩略词	含义
e.g.	Exempli gratia (拉丁语: 例如)
GMM	高斯混合模型
i.e.	Id est (拉丁语: 即)
i.i.d.	独立同分布
MAP	最大后验
MLE	最大似然估计/估计器
ONB	标准正交基
PCA	主成分分析
PPCA	概率主成分分析
REF	行阶梯形式
SPD	对称正定
SVM	支持向量机

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

第一部分

数学基础

[This material is published by Cambridge University Press as] [*Mathematics for Machine Learning*] [by] [Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong (2020). This version is free to view] [and download for personal use only. Not for re-distribution, re-sale, or use in derivative works.]

[©by M. P. Deisenroth, A. A. Faisal, and C. S. Ong, 2023.] [<https://mml-book.com>]

引言与动机

Machine learning是关于设计能够自动从数据中提取有价值信息的算法。这里强调的是”自动”，即machine learning关注的是可以应用于许多数据集的通用方法，同时产生有意义的结果。Machine learning的核心有三个概念：数据、模型和学习。

由于machine learning本质上是数据驱动的，数据是machine learning的核心。Machine learning的目标是设计通用方法来从数据中提取有价值的模式，理想情况下不需要太多特定领域的专业知识。例如，给定大量文档语料库（如许多图书馆中的书籍），machine learning方法可以用来自动发现文档间共享的相关主题(Hoffman et al, 2010)。为了实现这个目标，我们设计模型，这些模型通常与生成数据的过程相关，类似于我们给定的数据集。例如，在回归设置中，模型将描述一个将输入映射到实值输出的函数。

引用Mitchell (1997)的话：如果一个模型在考虑数据后在给定任务上的性能有所改善，那么就说该模型从数据中学习。目标是找到能够很好地泛化到未见数据的好模型，这些数据是我们将来可能关心的。*Learning*可以理解为一种[[learning]]通过优化模型参数来自动发现数据中的模式和结构的方法。

虽然machine learning已经有许多成功案例，并且有现成的软件可以设计和训练丰富灵活的machine learning系统，但我们相信machine learning的数学基础对于理解更复杂的machine learning系统所建立的基本原理很重要。理解这些原理可以促进创建新的machine learning解决方案，理解和调试现有方法，以及了解我们所使用方法的内在假设和局限性。

Mathematics for Machine Learning 由 Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong (2020) 著，Cambridge University Press出版。此版本免费供个人查看和下载使用。不得重新分发、转售或用于衍生作品。

©by M. P. Deisenroth, A. A. Faisal, and C. S. Ong, 2023. <https://mml-book.com>

介绍和动机

1.1 为直觉寻找词汇

我们在machine learning中经常面临的一个挑战是概念和词汇很滑溜，machine learning系统的特定组件可以抽象为不同的数学概念。例如，“algorithm”这个词在machine learning的语境中至少有两种不同的含义。第一种含义是，我们使用“machine learning algorithm”这个短语来指基于输入数据进行预测的系统。我们将这些algorithm称为 *predictor*。第二种含义是，我们使用完全相同的短语“machine learning algorithm”来指调整predictor的一些内部参数使其在未来未见输入数据上表现良好的系统。这里我们将这种调整称为 *training* 系统。

本书不会解决歧义问题，但我们想预先强调，根据上下文，相同的表达可能意味着不同的事情。然而，我们试图使上下文足够清晰以减少歧义程度。

本书的第一部分介绍了讨论machine learning系统三个主要组件所需的数学概念和基础：数据、模型和学习。我们将在这里简要概述这些组件，并在讨论了必要的数学概念后在第8章重新审视它们。

虽然不是所有数据都是数值的，但将数据考虑为数字格式通常是有用的。在本书中，我们假设 *data* 已经被适当地转换为适合读入计算机程序的数值表示。因此，我们将数据视为向量。作为词汇微妙性的另一个例证，至少有三种不同的方式来思考向量：向量作为数字数组（计算机科学观点），向量作为具有方向和大小的箭头（物理学观点），以及向量作为遵循加法和缩放的对象（数学观点）。

model 通常用于描述生成数据的过程，类似于手头的数据集。因此，好的模型也可以被认为是真实（未知）数据生成过程的简化版本，捕获与建模数据和从中提取隐藏模式相关的方面。好的模型然后可以用于预测在现实世界中会发生什么，而无需进行现实世界的实验。

我们现在来到问题的关键，machine learning的 *learning* 组件。假设我们得到一个数据集和一个合适的模型。*Training* 模型意味着使用可用数据来优化模型的一些参数，相对于评估模型预测训练数据能力的效用函数。大多数训练方法可以被认为是类似于爬山到达山峰的方法。在这个类比中，山峰对应于某个期望性能度量的最大值。然而，在实践中，我们感兴趣的是模型在未见数据上表现良好。在我们已经见过的数据（训练数据）上表现良好可能只意味着我们找到了记忆数据的好方法。然而，这可能无法很好地泛化到未见数据，并且在实际应用中，我们经常需要将我们的machine learning系统暴露于它以前没有遇到过的情况。

让我们总结一下本书涵盖的machine learning的主要概念：

我们将数据表示为向量。

我们选择适当的模型，使用概率或优化观点。

我们通过使用数值优化方法从可用数据中学习，目标是模型在未用于训练的数据上表现良好。

1.2 阅读本书的两种方式

我们可以考虑理解machine learning数学的两种策略：

自下而上：从基础概念构建到更高级的概念。这通常是更技术性领域（如数学）中的首选方法。这种策略的优势是读者在任何时候都能够依赖他们以前学过的概念。不幸的是，对于从业者来说，许多基础概念并不

自身特别有趣，缺乏动机意味着大多数基础定义很快就会被遗忘。

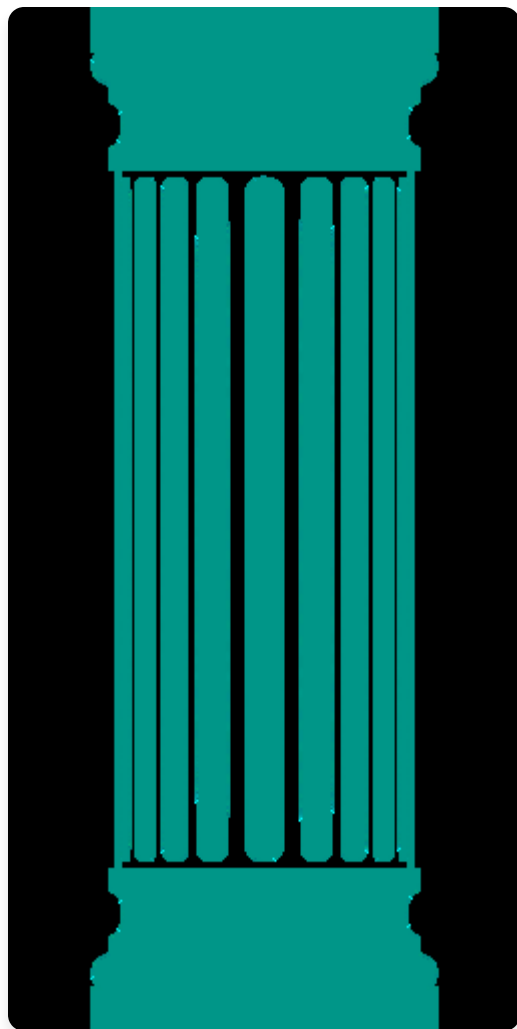
自上而下：从实际需求深入到更基本的要求。这种目标驱动的方法的优势在于读者始终知道为什么需要学习特定概念，并且有清晰的所需知识路径。这种策略的缺点是知识建立在可能不稳固的基础之上，读者必须记住一系列他们无法理解的词汇。

我们决定以模块化的方式编写本书，将基础（数学）概念与应用分离，这样本书可以用两种方式阅读。本书分为两个部分，第一部分奠定数学基础，第二部分将第一部分的`概念`应用到一系列基础机器学习问题中，这些问题构成了机器学习的四个支柱，如图1.1所示：回归、降维、密度估计和分类。第一部分的章节主要建立在前面章节的基础上，但如果需要，可以跳过某个章节并向后回顾。第二部分的章节之间联系较松散，可以按任何顺序阅读。两个部分之间有许多前向和后向指针

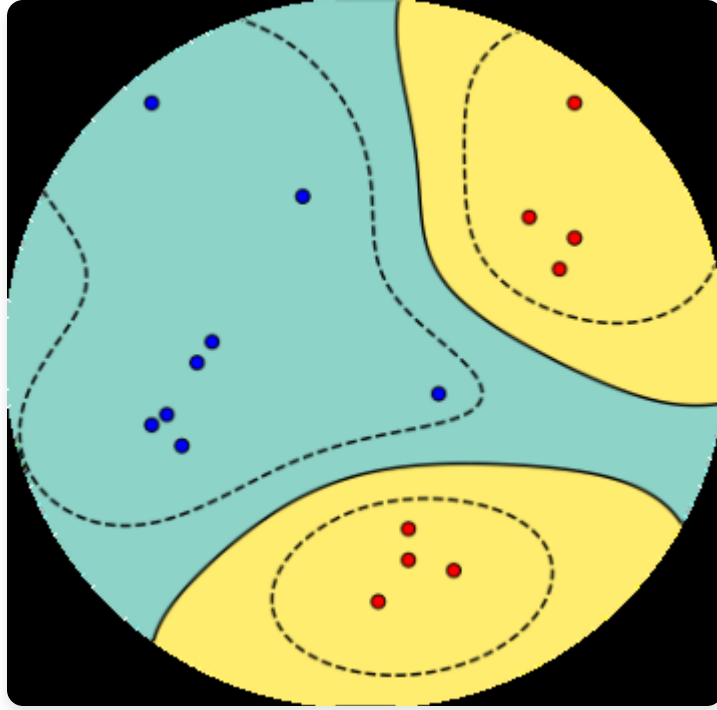
[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[14] [*Introduction and Motivation*]]

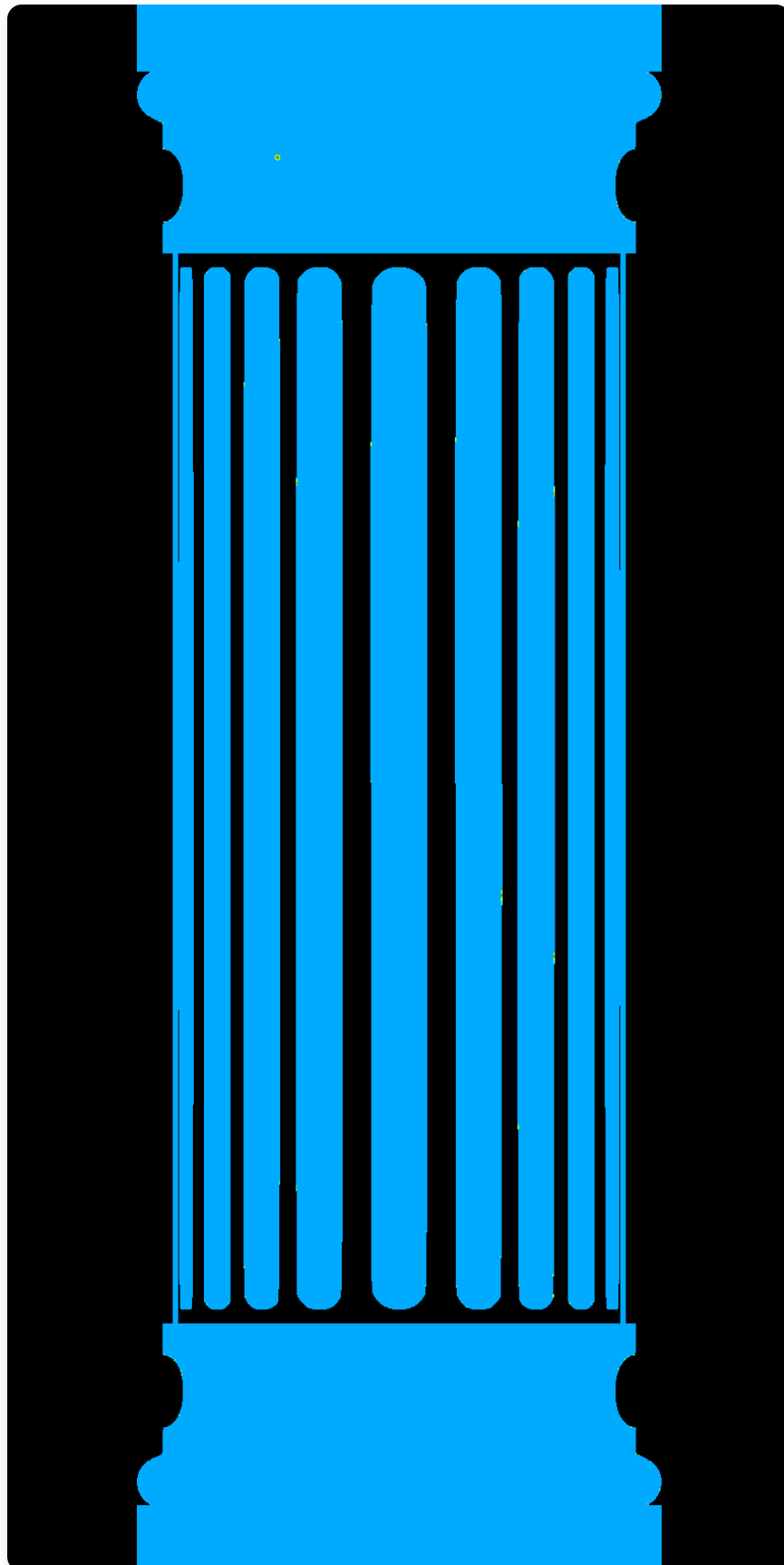
图1.1 机器学习的基础和四个支柱。



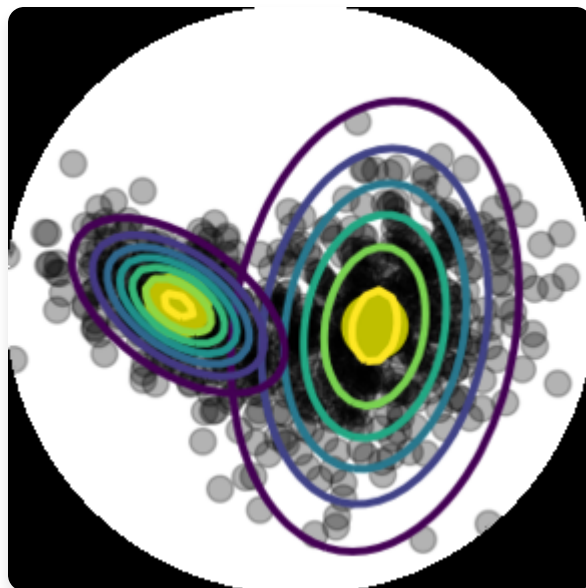
降维 密度



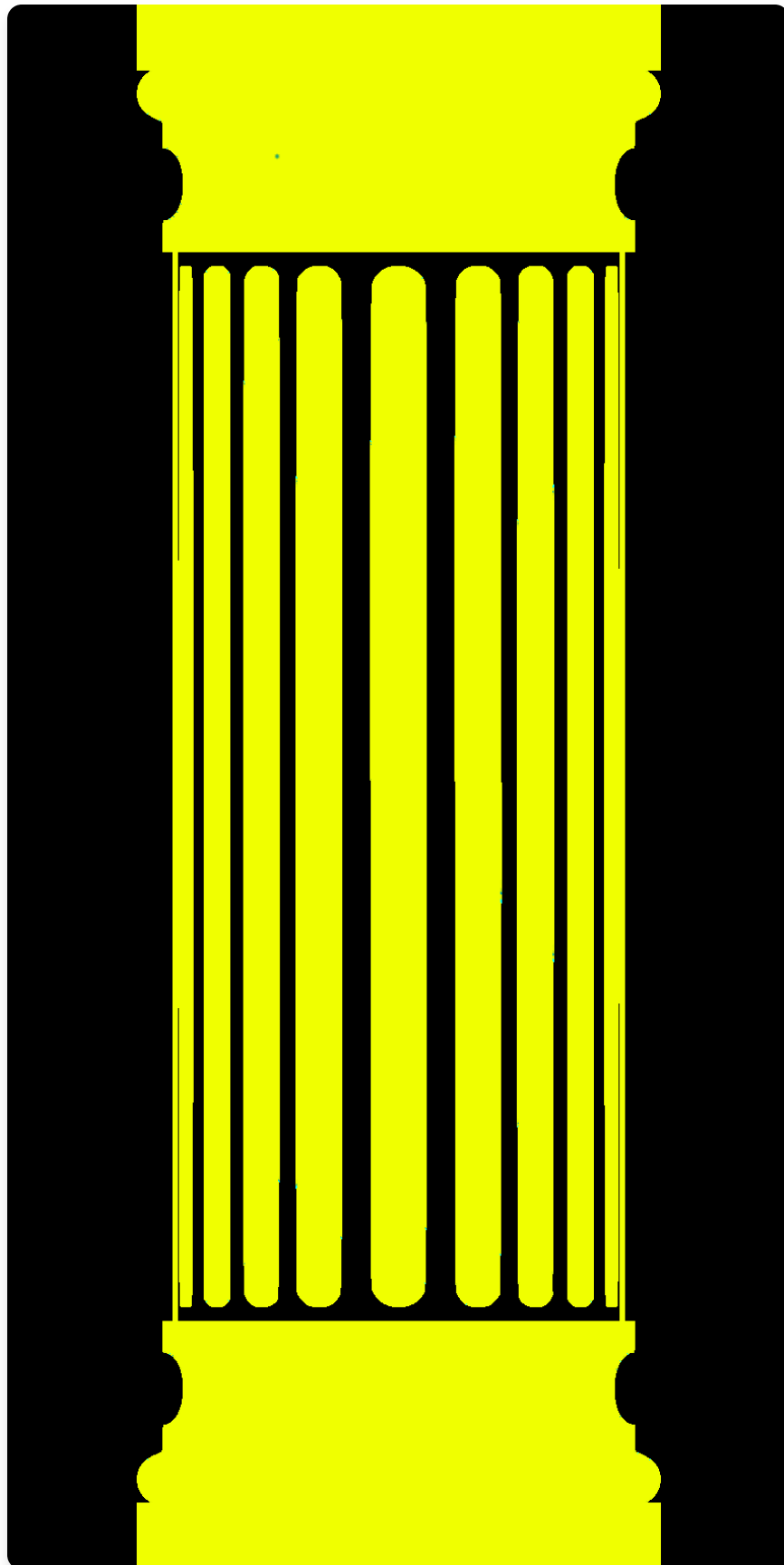
回归 降维 估计 分类



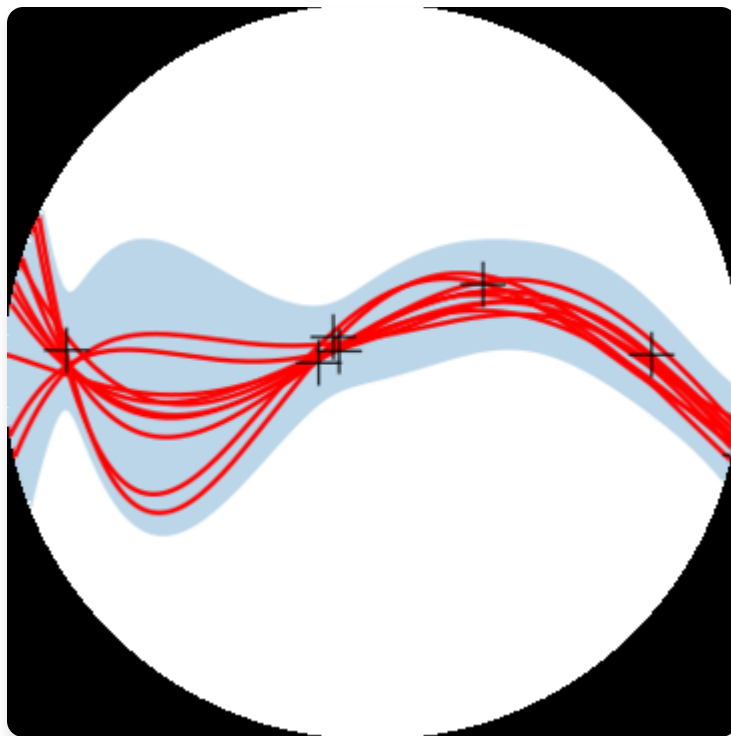
向量微积分 概率与分布 优化



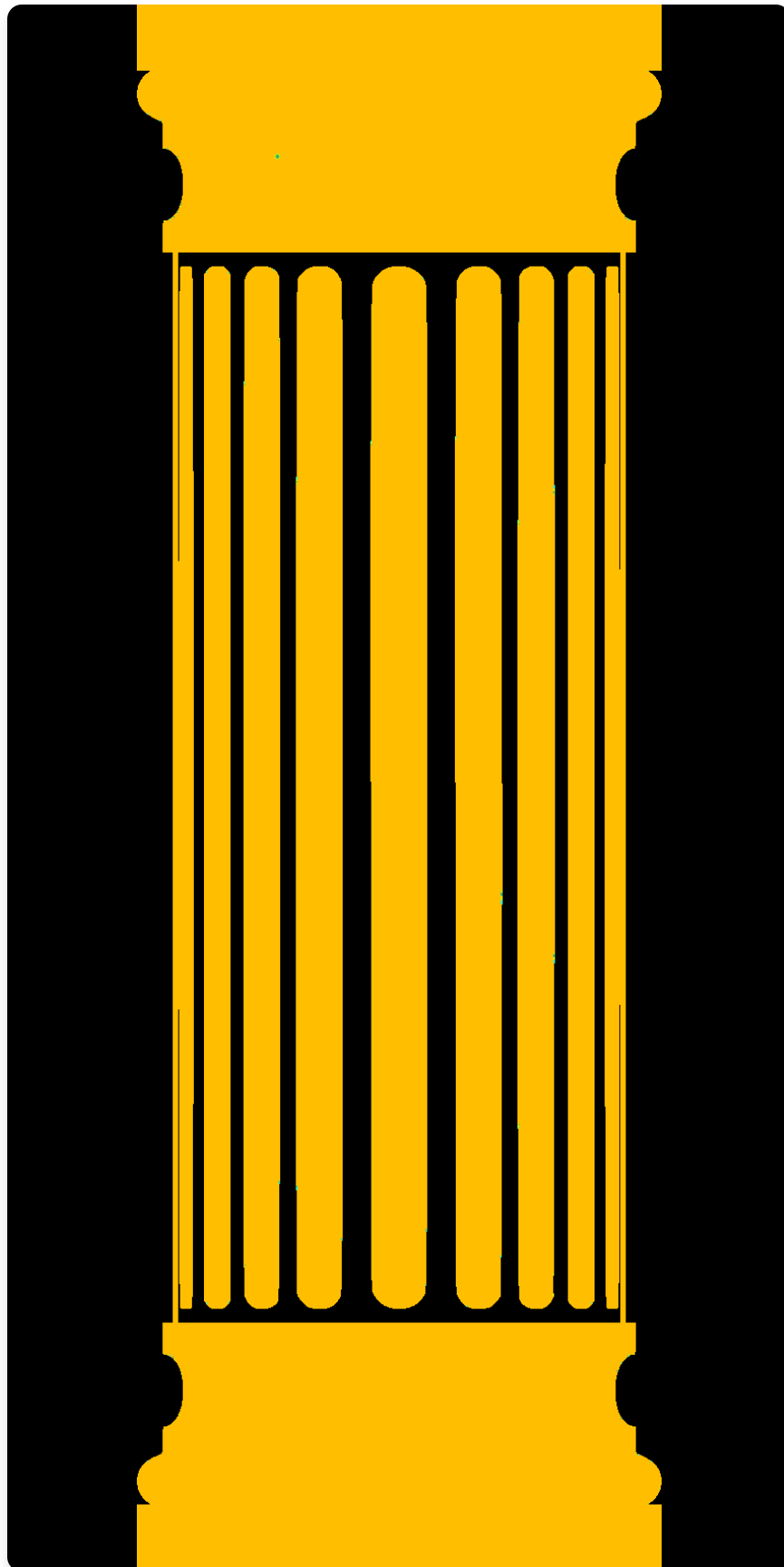
线性代数 解析几何 矩阵分解



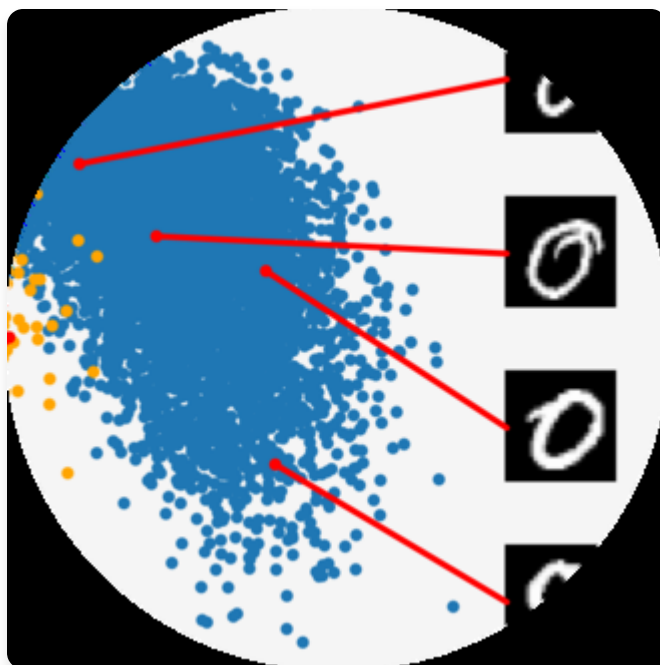
来连接数学概念与



机器学习算法。



当然阅读本书的方式不止两种。大多数读者



使用自上而下和自下而上方法的结合来学习，有时在尝试更复杂的概念之前建立基础数学技能，但也会根据机器学习的应用来选择主题。

第一部分关于数学

我们在本书中涵盖的机器学习四个支柱（见图1.1）需要坚实的数学基础，这在第一部分中进行了阐述。

我们将数值数据表示为向量，将此类数据的表格表示为矩阵。向量和矩阵的研究称为线性代数，我们在第2章中介绍。向量集合作为矩阵也在那里描述。

给定代表现实世界中两个对象的两个向量，我们想要对它们的相似性做出陈述。其思想是相似的向量应该被我们的机器学习算法（我们的预测器）预测为具有相似的输出。为了形式化向量之间相似性的概念，我们需要引入以两个向量作为输入并返回表示其相似性的数值的运算。相似性和距离的构造是解析几何的核心，在第3章中讨论。

在第4章中，我们介绍了关于矩阵和矩阵分解的一些基本概念。矩阵上的某些运算在机器学习中极其有用，它们允许对数据进行直观的解释和更高效的学习。

我们经常将数据视为某些真实潜在信号的噪声观测。我们希望通过应用机器学习能够从噪声中识别信号。这要求我们有一种量化“噪声”含义的语言。我们通常还希望有允许我们表达某种不确定性的预测器，例如，量化我们对特定测试数据点预测值的置信度。不确定性的量化是概率论的领域，

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] [<https://mml-book.com>][.] [*1.2 Two Ways to Read This Book*] [15]

在第6章中涵盖。

为了训练机器学习模型，我们通常寻找最大化某种性能指标的参数。许多优化技术需要梯度的概念，它告诉我们搜索解决方案的方向。第5章关于向量微积分，详细介绍了梯度的概念，我们随后在第7章中使用，其中我们讨论寻找函数极大值/极小值的优化。

第二部分关于机器学习

本书的第二部分介绍了机器学习的四个支柱，如图1.1所示。我们说明了本书第一部分介绍的数学概念如何成为每个支柱的基础。从广义上讲，章节按难度排序（按升序）。

在第8章中，我们以数学方式重述机器学习的三个组成部分（数据、模型和参数估计）。此外，我们提供了一些构建实验设置的指导原则，以防止对机器学习系统过于乐观的评估。回想一下，目标是构建在未见数据上表现良好的预测器。

在第9章中，我们将仔细研究线性回归，其中我们的

目标是找到将输入 $[D] \mathbf{x} \in \mathbf{R}$ 映射到相应观测函数值 $y \in \mathbf{R}$ 的函数，我们可以将其解释为各自输入的标签。我们将讨论通过最大似然和最大后验估计进行的经典模型拟合（参数估计），以及贝叶斯线性回归，其中我们对参数进行积分而不是优化它们。

第 [10] 章专注于降维，这是图- [[dimensionality]]

ure [1.1] 中的第二个支柱，使用主成分分析。降维的关键目标是找到高维数据 $[D] \mathbf{x} \in \mathbf{R}$ 的紧凑、低维表示，这通常比原始数据更容易分析。与回归不同，降维仅关注对数据进行建模——没有与数据点 $\mathbf{x}$ 相关联的标签。

在第 [11] 章中，我们将转向第三个支柱：密度估计。[[density estimation]]

密度估计的目标是找到描述给定数据集的概率分布。我们将专注于高斯混合模型，并讨论找到该模型参数的迭代方案。与降维一样，没有与数据点 $[D] \mathbf{x} \in \mathbf{R}$ 相关联的标签。然而，我们不寻求数据的低维表示。相反，我们对描述数据的密度模型感兴趣。

第 [12] 章通过深入讨论第四个支柱来结束本书：

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[16] [[引言和动机]]

[classification] 支柱：分类。我们将在支持向量机的背景下讨论分类。与回归（第 9 章）类似，我们有输入 x 和相应的标签 y 。然而，与标签为实值的回归不同，分类中的标签是整数，这需要特别注意。

1.3 练习和反馈

我们在第一部分提供了一些练习，大部分可以用纸笔完成。对于第二部分，我们提供编程教程（jupyter notebooks）来探索本书讨论的机器学习算法的一些特性。

我们感谢 Cambridge University Press 强烈支持我们的目标，即通过在以下网址免费提供本书下载来实现教育和学习的民主化：

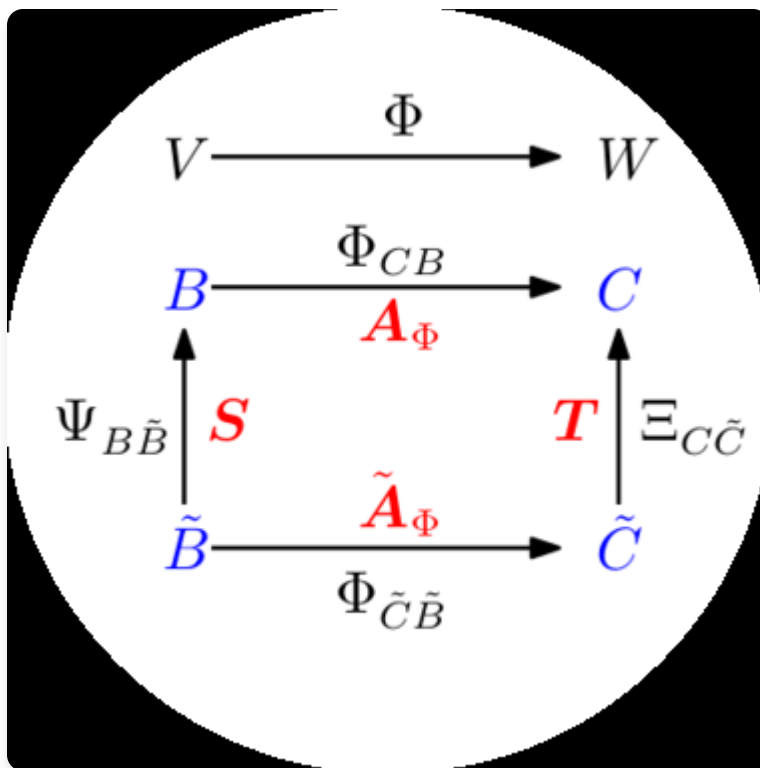
<https://mml-book.com>

在那里可以找到教程、勘误表和其他材料。可以使用上述网址报告错误和提供反馈。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] [<https://mml-book.com>][.]

[2]

线性代数



在形式化直观概念时，一种常见的方法是构造一组对象（符号）和一组操作这些对象的规则。这被称为代数。线性代数是研究向量和操作向量的某些规则的学科。我们许多人从学校了解的向量是

被称为”几何向量”的向量，通常用字母上方的小箭头表示，例如 $\vec{x}$ 和 $\vec{y}$ 。在本书中，我们讨论更一般的向量概念，并使用粗体字母表示它们，例如 $\mathbf{x}$ 和 $\mathbf{y}$ 。

一般来说，向量是可以相加并与标量相乘以产生同种类型的另一个对象的特殊对象。从抽象数学观点来看，满足这两个性质的任何对象都可以被认为是向量。以下是此类向量对象的一些例子：

1. 几何向量。这种向量的例子可能在高中数学和物理中很熟悉。几何向量——见图 [2.1(a)]——是有向线段，可以绘制（至少在二维中）。两个几何向量 $\vec{x}$ 、 $\vec{y}$ 可以相加，使得 $\vec{x} + \vec{y} = \vec{z}$ 是另一个几何向量。此外，与标量 $\lambda \vec{x}$ 、 $\lambda \in \mathbb{R}$ 的乘法也是几何向量。实际上，它是按 λ 缩放的原始向量。因此，几何向量是前面介绍的向量概念的实例。将向量解释为几何向量使我们能够使用关于方向和大小的直觉来推理数学运算。

2. 多项式也是向量；见图 [2.1(b)]：两个多项式可以

[→] [4] [图 2.1] [→]

$\mathbf{x} + \mathbf{y}$ [[不同类型的]]

[2] [[向量。向量可以]]

[0] [是令人惊讶的]

[y] [[对象，包括]]

[→] [-][2] [(a)] 几何] x [→] [向量]

[y] [4]

[和 [(b)] 多项式。]

- [2] [-] [0] [2]

x

[(a) 几何向量。] [(b) 多项式。]

[17]

[This material is published by Cambridge University Press as] [*Mathematics for Machine Learning*] [by] [Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong (2020). This version is free to view] [and download for personal use only. Not for re-distribution, re-sale, or use in derivative works.]

[©by M. P. Deisenroth, A. A. Faisal, and C. S. Ong, 2023.] [<https://mml-book.com>][.]

[18] [[线性代数]]

相加，结果是另一个多项式；它们可以与标量 $\lambda \in \mathbb{R}$ 相乘，结果也是多项式。因此，多项式是向量的（相当不寻常的）实例。注意多项式与几何向量非常不同。虽然几何向量是具体的“图形”，但多项式是抽象概念。然而，它们都是前面描述意义上的向量。

3. 音频信号是向量。音频信号表示为一系列

数字。我们可以将音频信号相加，它们的和是一个新的音频信号。如果我们缩放一个音频信号，我们也会得到一个音频信号。因此，音频信号也是一种向量。

4. $\mathbb{R}^n$ 的元素（ n 个实数的元组）是向量。 $\mathbb{R}^n$ 比多项式更抽象，它是在本书中关注的概念。例如，

$$\mathbf{a} = [1, 2, 3]^T \in \mathbb{R}^3 \quad (2.1)$$

是一个三元数组的例子。将两个向量 $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ 按分量相加会得到另一个向量： $\mathbf{a} + \mathbf{b} = \mathbf{c} \in \mathbb{R}^n$ 。此外，将 $\mathbf{a} \in \mathbb{R}^n$ 乘以 $\lambda \in \mathbb{R}$ 会得到一个缩放向量 $\lambda \mathbf{a} \in \mathbb{R}^n$ 。

将向量视为 $\mathbb{R}^n$ 的元素还有一个额外的好处，即它大致对应于计算机上的实数数组。许多编程语言支持数组运算，这允许方便地实现涉及向量运算的算法。

线性代数专注于这些向量概念之间的相似性。我们可以将它们相加并乘以标量。我们将主要关注 $\mathbb{R}^n$ 中的向量，因为线性代数中的大多数算法都是在 $\mathbb{R}^n$ 中制定的。我们将在第8章中看到，我们经常将数据视为 $\mathbb{R}^n$ 中的向量。在本书中，我们将专注于有限维向量空间，在这种情况下，任何类型的向量与 $\mathbb{R}^n$ 之间存在一一对应关系。当方便时，我们将使用关于几何向量的直觉并考虑基于数组的算法。

数学中的一个主要思想是“闭合”的概念。这是一个问题：我提出的运算可能产生的所有事物的集合是什么？在向量的情况下：从一小组向量开始，将它们相加并缩放它们，可能产生的向量集合是什么？这导致了向量空间（第2.4

节)。向量空间的概念及其性质是机器学习的基础。本章介绍的概念总结在图2.2中。

本章主要基于 Drumm 和 Weil (2001)、Strang (2003)、Hogben (2013)、Liesen 和 Mehrmann (2015) 的讲义和书籍，以及 Pavel Grinfeld 的线性代数系列。其他优秀的

图2.2 本章介绍的概念的思维导图，以及它们在本书其他部分中的使用位置。

资源包括 Gilbert Strang 在 MIT 的线性代数课程和 3Blue1Brown 的线性代数系列。

线性代数在机器学习和一般数学中发挥着重要作用。本章介绍的概念在第3章中进一步扩展，包括几何的概念。在第5章中，我们将讨论向量微积分，其中矩阵运算的有原则的知识是必不可少的。在第10章中，我们将使用投影（将在第3.8节中介绍）进行主成分分析(PCA)的降维。在第9章中，我们将讨论线性回归，其中线性代数在解决最小二乘问题中发挥核心作用。

2.1 线性方程组

线性方程组是线性代数的核心部分。许多问题可以表述为线性方程组，线性代数为我们提供了解决它们的工具。

例 2.1

一家公司生产产品 $N_1, \dots, N_n$ ，需要资源 $R_1, \dots, R_m$ 。为了生产一单位产品 N_j ，需要 a_{ij} 单位资源 R_i ，其中 $i = 1, \dots, m$ 和 $j = 1, \dots, n$ 。

目标是找到一个最优生产计划，即如果总共有 b_i 单位资源 R_i 可用且（理想情况下）没有资源剩余，应该生产多少单位 x_j 的产品 N_j 的计划。

如果我们生产 $x_1, \dots, x_n$ 单位相应的产品，我们需要总共

$$a_{i1}x_1 + \dots + a_{in}x_n \quad (2.2)$$

单位资源 R_i 。因此，一个最优生产计划 $(x_1, \dots, x_n) \in \mathbb{R}^n$ 必须满足以下方程组：

$$a_{i1}x_1 + \dots + a_{in}x_n = b_i \quad ; \quad a_{m1}x_1 + \dots + a_{mn}x_n = b_m \quad (2.3)$$

其中 $a_{ij} \in \mathbb{R}$ 和 $b_i \in \mathbb{R}$ 。

方程(2.3)是**线性方程组**的一般形式， $x_1, \dots, x_n$ 是该系统的**未知数**。满足(2.3)的每个 n 元组 $(x_1, \dots, x_n) \in \mathbb{R}^n$ 都是线性方程组的**解**。

例 2.2

线性方程组

$$x_1 + x[2] + x_3 = 3 \quad (1) \quad x_1 - x[2] + 2x_3 = 2 \quad (2) \quad (2.4)$$

$$2x_1 + 3x_3 = 1 \quad (3)$$

无解：将前两个方程相加得到 $2x_1 + 3x_3 = 5$ ，这与第三个方程 (3) 矛盾。

让我们看看线性方程组

$$x_1 + x[2] + x_3 = 3 \quad (1) \quad x_1 - x[2] + 2x_3 = 2 \quad (2) \quad (2.5)$$

$$x[2] + x_3 = 2 \quad (3)$$

从第一个和第三个方程，可以得出 $x_1 = 1$ 。从 (1)+(2)，我们得到 $2x_1 + 3x_3 = 5$ ，即 $x_3 = 1$ 。从 (3)，我们得到 $x[2] = 1$ 。因此， $(1, 1, 1)$ 是唯一可能的**唯一解**（通过代入验证 $(1, 1, 1)$ 是一个解）。

作为第三个例子，我们考虑

$$x_1 + x[2] + x_3 = 3 \quad (1) \quad x_1 - x[2] + 2x_3 = 2 \quad (2) \quad (2.6)$$

$$2x_1 + 3x_3 = 5 \quad (3)$$

由于 (1)+(2)=(3)，我们可以省略第三个方程（冗余）。从 (1) 和 (2)，我们得到 $2x_1 = 5 - 3x_3$ 和 $2x_2 = 1 + x_3$ 。我们定义 $x_3 = a \in \mathbb{R}$ 作为自由变量，这样任何三元组

$$5/2 - 3a/2, 1/2 + a/2, a, a \in \mathbb{R} \quad (2.7)$$

都是线性方程组的解，即我们得到一个包含**无穷多解**的解集。

一般来说，对于实值线性方程组，我们得到要么无解、恰好一个解，要么无穷多解。线性回归（第9章）在我们无法求解线性方程组时解决例2.1的一个版本。

备注（线性方程组的几何解释）。在有两个变量 x_1, x_2 的线性方程组中，每个线性方程在 x_1x_2 -平面上定义一条直线。由于线性方程组的解必须同时满足所有方程，解集是这些直线的交集。这个交集可以是一条直线（如果线性方程描述同一条直线）、一个点，或空集（当直线平行时）。图2.3给出了系统的示例

$$4x_1 + 4x_2 = 5 \quad 2x_1 - 4x_2 = 1 \quad (2.8)$$

其中解空间是点 $(x_1, x_2) = (1, 1/4)$ 。类似地，对于三个变量，每个线性方程在三维空间中确定一个平面。当我们求这些平面的交集，即同时满足所有线性方程时，我们可以得到一个解集，它是一个平面、一条直线、一个点或空集（当平面没有公共交点时）。◇

为了系统地求解线性方程组，我们将介绍一个有用的紧凑记号。我们将系数 a_{ij} 收集到向量中，并将向量收集到矩阵中。换句话说，我们将来自(2.3)的系统写成以下形式：

$$\begin{array}{ccccccc} a_{11} & a_{12} & \cdots & a_{1n} & & b_1 \\ \vdots & \vdots & & \vdots & \cdot x_1 & + x_2 & + \cdots + x_n = \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & & b_m \end{array} \quad (2.9)$$

$$\Leftrightarrow \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \quad (2.10)$$

在下面，我们将仔细看看这些**矩阵**并定义计算规则。我们将在第2.3节回到求解线性方程。

2.2 矩阵

矩阵在线性代数中起着核心作用。它们可以用来紧凑地表示线性方程组，但它们也表示线性函数（线性映射），我们将在第2.7节看到。在我们讨论这些有趣的话题之前，让我们首先定义什么是矩阵以及我们可以对矩阵进行什么样的运算。我们将在第4章看到矩阵的更多性质。

定义2.1（矩阵）。对于 $m, n \in \mathbb{N}$ ，实值 (m, n) **矩阵** A 是元素 $a[ij], i = 1, \dots, m, j = 1, \dots, n$ 的 $m \cdot n$ -元组，按照由 m 行和 n 列组成的矩形方案排列：

$$a[11] \ a[12] \ \cdots \ a[1n] \ a[21] \ a[22] \ \cdots \ a[2n] \ A = \begin{pmatrix} \vdots & \vdots & \ddots & \vdots \\ a[ij] \in \mathbb{R} \end{pmatrix} \quad (2.11) \quad a[m1] \ a[m2] \ \cdots \ a[mn]$$

按照惯例， $(1, n)$ -矩阵称为**行**， $(m, 1)$ -矩阵称为**列**。这些特殊矩阵也称为**行/列向量**。 $\mathbb{R}[m \times n]$ 是所有实值 (m, n) -矩阵的集合。 $A \in \mathbb{R}[m \times n]$ 可以等价地表示为 $a \in \mathbb{R}[mn]$ ，通过将矩阵的所有 n 列堆叠成一个长向量；见图2.4。



图2.4 通过堆叠其列，矩阵A可以表示为长向量a。

图2.4 通过堆叠其列，矩阵A可以表示为长向量a。

2.2.1 矩阵加法和乘法

两个矩阵 $A \in \mathbb{R}[m \times n], B \in \mathbb{R}[m \times n]$ 的和定义为逐元素和，即

$$a[11] + b[11] \ \cdots \ a[1n] + b[1n] \ A + B := \begin{pmatrix} \vdots & \vdots & \vdots \\ a[ij] + b[ij] \end{pmatrix} \in \mathbb{R}[m \times n] \quad (2.12) \quad a[m1] + b[m1] \ \cdots \ a[mn] + b[mn]$$

对于矩阵 $A \in \mathbb{R}[m \times n], B \in \mathbb{R}[n \times k]$ ，乘积 $C = AB \in \mathbb{R}[m \times k]$ 的元素 $c[ij]$ 计算为

$$c[ij] = \sum_{l=1 \text{ to } n} a[il]b[lj], i = 1, \dots, m, j = 1, \dots, k. \quad (2.13)$$

这意味着，要计算元素 $c[ij]$ ，我们将第 i

第 $[i]$ 行 $[A]$ 和 $[B]$ 的第 $[j]$ 列的 $[n]$ 行与求和。稍后在3.2节中，

我们称这为对应行和列的**点积**。在需要明确表示我们正在执行乘法的情况下，我们使用符号 $A \cdot B$ 来表示乘法（明确显示“ $\cdot$ ”）。

备注。矩阵只有在在其“相邻”维数匹配时才能相乘。例如， $n \times k$ 矩阵 A 可以与 $k \times m$ 矩阵 B 相乘，但只能从左侧：

$$A B = C \quad (2.14) \quad \{z\} \mid \{z\} \mid \{z\} \quad n \times k \quad k \times m \quad n \times m$$

如果 $m \neq n$ ，由于相邻维数不匹配，乘积 BA 未定义。◇

备注。矩阵乘法**不是**定义为矩阵元素的逐元素运算，即 $c_{ij} \neq a_{ij}b_{ij}$ （即使 A, B 的大小选择得当）。这种逐元素乘法在编程语言中经常出现，当我们将（多维）数组相乘时，称为**Hadamard乘积**。◇

例2.3

对于 $A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix} \in \mathbb{R}^{(2 \times 3)}$, $B = \begin{bmatrix} 0 & 2; 1 & -1; 0 & 1 \end{bmatrix} \in \mathbb{R}^{(3 \times 2)}$, 我们得到

$$AB = \begin{bmatrix} 1 & 2 & 3; 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2; 1 & -1; 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3; 2 & 5 \end{bmatrix} \in \mathbb{R}^{(2 \times 2)}, \quad (2.15)$$

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

图2.5 即使两个矩阵乘法AB和BA都有定义, 结果的维数也可能不同。

从这个例子中, 我们已经可以看出矩阵乘法不满足交换律, 即 $AB \neq BA$; 也见图2.5的图解。

定义2.2 (单位矩阵)。在 $\mathbb{R}^{(n \times n)}$ 中, 我们定义**单位矩阵**

$$I_n := \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0; 0 & 1 & \cdots & 0 & \cdots & 0; \dots & \dots & \dots & \dots & \dots & \dots; 0 & 0 & \cdots & 1 & \cdots & 0; \dots & \dots & \dots & \dots & \dots & \dots; 0 & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{(n \times n)} \quad (2.17)$$

作为对角线上包含1、其他地方都是0的 $n \times n$ 矩阵。

现在我们定义了矩阵乘法、矩阵加法和单位矩阵, 让我们看看矩阵的一些性质:

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

$$\text{分配律: } \forall A, B \in \mathbb{R}^{(m \times n)}, C, D \in \mathbb{R}^{(n \times p)}: (A+B)C = AC+BC \quad (2.19a) \quad A(C+D) = AC+AD \quad (2.19b)$$

$$\text{与单位矩阵的乘法: } \forall A \in \mathbb{R}^{(m \times n)}: I_m A = A I_n = A \quad (2.20)$$

注意对于 $m \neq n$, $I_m \neq I_n$ 。

2.2.2 逆矩阵和转置

定义2.3（逆矩阵）。考虑方阵 $A \in \mathbb{R}^{(n \times n)}$ 。设矩阵 $B \in \mathbb{R}^{(n \times n)}$ 具有性质 $AB = I_n = BA$ 。B 称为 A 的**逆矩阵**，记作 A^{-1} 。

不幸的是，并非每个矩阵 A 都有逆矩阵 A^{-1} 。如果这个逆矩阵存在，A 称为**正则/可逆/非奇异的**，否则称为**奇异/不可逆的**。当矩阵逆存在时，它是唯一的。在 2.3 节中，我们将讨论通过求解线性方程组来计算矩阵逆的一般方法。

备注（ 2×2 矩阵逆矩阵的存在性）。考虑矩阵

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

如果我们将 A 与

$$A' := \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \quad (2.22)$$

相乘，我们得到

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

因此，

$$A^{-1} = 1/(a_{11}a_{22} - a_{12}a_{21}) \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \quad (2.24)$$

当且仅当 $a_{11}a_{22} - a_{12}a_{21} \neq 0$ 。在 4.1 节中，我们将看到 $a_{11}a_{22} -$

$a_{12}a_{21}$ 是 2×2 矩阵的行列式。此外，我们通常可以使用行列式来检查矩阵是否可逆。◇

例2.4（逆矩阵）

矩阵

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

互为逆矩阵，因为 $AB = I = BA$ 。

定义2.4（转置）。对于 $A \in \mathbb{R}^{(m \times n)}$ ，矩阵 $B \in \mathbb{R}^{(n \times m)}$ 中 $b_{ij} = a_{ji}$ 称为 A 的**转置**。我们写作 $B = A^T$ 。

一般来说， A^T 可以通过将 A 的列写作 A^T 的行来获得。以下是逆矩阵和转置的重要性质：

矩阵的**主对角线**（有时称为“主对角线”、“首对角线”、“前导对角线”或“主要对角线”）是矩阵 A 中 $i=j$ 的元素 A_{ij} 的集合。

$$AA^{-1} = I = A^{-1}A \quad (2.26)$$

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

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

(2.28)的标量情况是 $1/(2+4) \neq 1/2 + 1/4 = 6$ 。

$$(A^T)^T = A \quad (2.29)$$

$$(AB)^T = B^T A^T \quad (2.30)$$

$$(A+B)^T = A^T + B^T \quad (2.31)$$

定义2.5（对称矩阵）。矩阵 $A \in \mathbb{R}^{(n \times n)}$ 是**对称的**，如果 $A^T = A$ 。

注意只有 (n,n) 矩阵才能是对称的。一般来说，我们也称 (n,n) 矩阵为**方阵**，因为它们具有相同的行数和列数。此外，如果 A^T 可逆，那么 A 也可逆，且 $(A^{-1})^T = (A^T)^{-1} =: A^{-T}$ 。

备注（对称矩阵的和与积）：对称矩阵 $A, B \in \mathbb{R}^1$ 的和总是对称的。然而，虽然它们的乘积总是有定义的，但通常不是对称的：

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \neq \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (2.32)$$

◇

2.2.3 标量乘法

让我们看看当矩阵与标量相乘时会发生什么。设 $A \in \mathbb{R}^2$ 和 $\lambda \in \mathbb{R}$ 。则 $\lambda A = K$ ，其中 $K_{ij} = \lambda a_{ij}$ 。

实际上， λ 缩放 A 的每个元素。对于 $\lambda, \psi \in \mathbb{R}$ ，以下关系成立：

结合律(Associativity):

$(\lambda\psi)C = \lambda(\psi C)$, $C \in \mathbb{R}^3$ $\lambda(BC) = (\lambda B)C = B(\lambda C) = (BC)\lambda$, $B \in \mathbb{R}^4$, $C \in \mathbb{R}^5$ 注意这允许我们移动标量值。

$(\lambda C)^T = C^T \lambda = C^T \lambda = \lambda C^T$ ，因为对于所有 $\lambda \in \mathbb{R}$, $\lambda^T = \lambda$ 。

分配律(Distributivity):

$(\lambda + \psi)C = \lambda C + \psi C$, $C \in \mathbb{R}^6$ $\lambda(B + C) = \lambda B + \lambda C$, $B, C \in \mathbb{R}^7$

例 2.5 (分配律)

如果我们定义

$$C = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad (2.33)$$

那么对于任意 $\lambda, \psi \in \mathbb{R}$ ，我们得到

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

$$(\lambda + \psi)C = \begin{bmatrix} \lambda + \psi & 2\lambda + 2\psi \\ 3\lambda + 3\psi & 4\lambda + 4\psi \end{bmatrix} = \begin{bmatrix} \lambda & 2\lambda \\ \psi & 2\psi \end{bmatrix} + \begin{bmatrix} \psi & 2\psi \\ 3\psi & 4\psi \end{bmatrix} = \lambda C + \psi C \quad (2.34b)$$

$$= \begin{bmatrix} \lambda & 2\lambda \\ \psi & 2\psi \end{bmatrix} + \begin{bmatrix} \psi & 2\psi \\ 3\psi & 4\psi \end{bmatrix} = \lambda C + \psi C \quad (2.34b)$$

2.2.4 线性方程组的紧凑表示

如果我们考虑线性方程组

$$2x_1 + 3x_2 + 5x_3 = 1 \quad 4x_1 - 2x_2 - 7x_3 = 8 \quad (2.35) \quad 9x_1 + 5x_2 - 3x_3 = 2$$

并使用矩阵乘法的规则，我们可以将这个方程组写成更紧凑的形式：

$$\begin{array}{ccc|ccc} 2 & 3 & 5 & x_1 & & 1 \\ 4 & -2 & -7 & x_2 & = & 8 \\ 9 & 5 & -3 & x_3 & & 2 \end{array} \quad (2.36)$$

注意 x_1 缩放第一列， x_2 缩放第二列， x_3 缩放第三列。

一般来说，线性方程组可以用其矩阵形式紧凑地表示为 $Ax = b$ ；参见 (2.3)，乘积 Ax 是 A 的列的（线性）组合。我们将在第 2.5 节中更详细地讨论线性组合。

2.3 求解线性方程组

在 (2.3) 中，我们介绍了方程组的一般形式，即

$$a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \quad ; \quad (2.37) \quad a_{m1}x_1 + \cdots + a_{mn}x_n = b_m$$

其中 $a_{ij} \in \mathbb{R}$ 和 $b_i \in \mathbb{R}$ 是已知常数， x_j 是未知数， $i = 1, \dots, m$ ， $j = 1, \dots, n$ 。到目前为止，我们看到矩阵可以用作表述线性方程组的紧凑方式，因此我们可以写成 $Ax = b$ ，参见 (2.10)。此外，我们定义了基本的矩阵运算，如矩阵的加法和乘法。接下来，我们将专注于求解线性方程组，并提供一个求矩阵逆的算法。

2.3.1 特解与通解

在讨论如何一般性地求解线性方程组之前，让我们看一个例子。考虑方程组

$$\begin{array}{cccc} 1 & 0 & 8 & -4 \\ 0 & 1 & 2 & 12 \end{array} \begin{array}{l} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} = \begin{array}{l} 42 \\ 8 \end{array} . \quad (2.38)$$

该系统有两个方程和四个未知数。因此，一般来说我们期望有无穷多个解。这个方程组具有特别简单的形式，其中前两列由 1 和 0 组成。记住我们想要找到标量 $x_1, \dots, x_4$ ，使得 $\sum_{i=1}^4 x_i c_i = b$ ，其中我们定义 c_i 为矩阵的第 i 列， b 为 (2.38) 的右端。问题 (2.38) 的解可以通过取第一列的 42 倍和第二列的 8 倍立即找到，使得

$$42 \quad 1 \quad 0$$

$$b = 8 = 42 \times 0 + 8 \times 1. \quad (2.39)$$

因此，一个解是 $[42, 8, 0, 0]^T$ 。这个解称为特解或特殊解。但是，这不是该线性方程组的唯一解。为了捕获所有其他解，我们需要巧妙地使用矩阵的列以非平凡的方式生成 0：向我们的特解添加 0 不会改变特解。为此，我们使用前两列（它们具有这种非常简单的形式）来表示第三列

$$8 \quad 1 \quad 0$$

$$= 8 \times 0 + 2 \times 1 \quad (2.40)$$

因此 $0 = 8c_1 + 2c_2 - 1c_3 + 0c_4$ ，即 $(x_1, x_2, x_3, x_4) = (8, 2, -1, 0)$ 。实际上，这个解的任意标量倍数 $\lambda_1 \in \mathbb{R}$ 都产生 0 向量，即

$$\begin{array}{cccc} 1 & 0 & 8 & -4 \\ 0 & 1 & 2 & 12 \end{array} \begin{array}{l} 8 \\ \lambda_1 \quad 2 \\ -1 \\ 0 \end{array} = \lambda_1 (8c_1 + 2c_2 - c_3) = 0. \quad (2.41)$$

按照同样的推理路线，我们使用前两列来表示 (2.38) 中矩阵的第四列，并生成另一组 0 的非平凡版本

$$\begin{array}{cccc} 1 & 0 & 8 & -4 \\ 0 & 1 & 2 & 12 \end{array} \begin{array}{l} -4 \\ \lambda_2 \quad 12 \\ 0 \\ -1 \end{array} = \lambda_2 (-4c_1 + 12c_2 - c_4) = 0 \quad (2.42)$$

对于任意 $\lambda_2 \in \mathbb{R}$ 。综合所有内容，我们得到 (2.38) 中方程组的所有解，称为通解，为集合

42 8 -4

$x \in \mathbb{R}^4: x = 8 + \lambda_1 \begin{bmatrix} 2 \\ 2 \\ 1 \\ 2 \end{bmatrix} + \lambda_2 \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix}, \lambda_1, \lambda_2 \in \mathbb{R} . \quad (2.43)$

备注：我们遵循的一般方法包括以下三个步骤：

- 1. 找到 $Ax = b$ 的一个特解。
- 2. 找到 $Ax = 0$ 的所有解。
- 3. 将步骤 1 和步骤 2 的解组合成通解。

通解和特解都不是唯一的。[◇]

前面例子中的线性方程组很容易求解，因为 (2.38) 中的矩阵具有这种特别方便的形式，

这使我们能够通过检查找到特解和通解。然而，一般的方程组并不具有这种简单的形式。

幸运的是，存在一种构造性的算法方法，可以将任何线性方程组转换为这种特别简单的形式：高斯消元法。高斯消元法的关键是线性方程组的基本变换，这些变换将方程组转换为简单形式。然后，我们可以将刚才在[(2.38)]例子中讨论的三个步骤应用到简单形式上。

2.3.2 基本变换

解线性方程组系统的关键是基本变换，这些变换保持解集不变，但将方程组转换为更简单的形式：

[Mathematics for Machine Learning草稿(2023-10-18)。反馈：<https://mml-book.com>. 2.3 解线性方程组 29]

交换两个方程（表示方程组的矩阵中的行）

用常数 $\lambda \in \mathbb{R} \setminus \{0\}$ 乘以一个方程（行）

两个方程（行）相加

例2.6

对于 $a \in \mathbb{R}$ ，我们求以下方程组的所有解：

$-2x_1 + 4x_2 - 2x_3 - x_4 + 4x_5 = -3$

$4x_1 - 8x_2 + 3x_3 - 3x_4 + x_5 = 2 . \quad (2.44)$

$x_1 - 2x_2 - 3x_4 + 4x_5 = a$

我们首先将这个方程组转换为紧凑的矩阵记号 $Ax = b$ 。我们不再明确提及变量 x ，并构建增广矩阵（形式为 $A \mid b$ ）

$-24 - 2 - 14 - 3$ 与 R_3 交换 $4 - 83 - 312$

$1 - 21 - 110$ 与 R_1 交换 $1 - 20 - 34a$

其中我们使用竖线来分隔[(2.44)]中的左边和右边。我们使用 $\rightarrow$ 来表示使用基本变换对增广矩阵的变换。增广矩阵[A | b]紧凑地表示线性方程组 $Ax = b$ 。

交换第1行和第3行得到

$1 - 21 - 110$

$4 - 83 - 312 + 2R_1 - 24 - 2 - 14 - 3$

$1 - 20 - 34a - R_1$

当我们在应用指定的变换（例如，从第2行减去第1行的四倍）时，我们得到

$1 - 21 - 110$

$00 - 11 - 32$

$000 - 36 - 3$

$00 - 1 - 23a - R[2] - R_3$

$\rightarrow 1 - 21 - 110$

$00 - 11 - 32 \cdot (-1) 000363 \cdot (-1/3) 00000a + 1$

$\rightarrow 1 - 21 - 110$

$001 - 13 - 200012100000a + 1$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

线性代数

这个（增广）矩阵处于一种方便的形式，即行阶梯形式(REF)。将这种紧凑记号转换回包含我们所求变量的显式记号，我们得到

$$x_1 - 2x_2 + x_3 - x_4 + x_5 = 0$$

$$x_3 - x_4 + 3x_5 = -2 \cdot (2.45) x_4 - 2x_5 = 1$$

$$0 = a + 1$$

只有当 $a = -1$ 时，这个系统才能被解决。一个特解是

$$\begin{matrix} x[1] & 2 \\ x[2] & 0 \end{matrix}$$

$$x = x_3 = -1 \cdot (2.46) x_4 \quad 1 \quad x_5 \quad 0$$

通解，它捕获所有可能解的集合，是

$$\begin{array}{ccc} 2 & 2 & 2 \\ 0 & 1 & 0 \end{array}$$

$$x \in \mathbb{R}: x = -1 + \lambda_1 0 + \lambda_2[-1], \lambda_1, \lambda_2 \in \mathbb{R}. \quad (2.47) \quad 1 \ 0 \ 2 \ 0 \ 0 \ 1$$

在下面，我们将详细介绍获得线性方程组特解和通解的构造性方法。

注释（主元和阶梯结构）。一行的首项系数（从左边开始的第一个非零数）称为主元，并且总是严格位于其上一行主元的右侧。因此，任何处于行阶梯形式的方程组总是具有”阶梯”结构。◇

定义2.6（行阶梯形式）。如果矩阵满足以下条件，则称其处于行阶梯形式：

只包含零的所有行都位于矩阵的底部；相应地，至少包含一个非零元素的所有行都位于只包含零的行的上方。

仅看非零行，从左边开始的第一个非零数（也称为主元或首项系数）总是严格位于其上一行主元的右侧。在其他文献中，有时要求主元为1。

注释（基本变量和自由变量）。在行阶梯形式中对应于主元的变量称为基本变量，其他变量是自由变量。例如，在[(2.45)]中， x_1 、 x_3 、 x_4 是基本变量，而 x_2 、 x_5 是自由变量。◇

注释（获得特解）。当我们需要确定特解时，行阶梯形式使

[Mathematics for Machine Learning草稿(2023-10-18)。反馈：<https://mml-book.com>. 2.3 解线性方程组 31]

我们的工作变得更容易。为此，我们使用主元列来表示方程组的右边，使得 $b = \sum_{i=1 \text{ 到 } P} \lambda_i p[i]$ ，其中 $p[i]$ ， $i = 1, \dots, P$ ，是主元列。如果我们从最右边的主元列开始并向左工作， λ_i 最容易确定。

在前面的例子中，我们将尝试找到 λ_1 、 λ_2 、 λ_3 ，使得

$$\begin{array}{cccc} 1 & 1 & -1 & 0 \\ 0 & 1 & -1 & -2 \end{array}$$

$$\lambda_1 + \lambda_2 + \lambda_3 = . \quad (2.48) \quad 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0$$

从这里，我们可以相对直接地找到 $\lambda_3 = 1$ ， $\lambda_2 = -1$ ， $\lambda_1 = 2$ 。当我们把所有内容整合在一起时，我们不能忘记非主元列，对于这些列我们隐式地将系数设为0。因此，我们得到特解 $x_T = [2, 0, -1, 1, 0]$ 。◇

注释（简化行阶梯形式）。一个方程组处于简化行阶梯形式(也称为行简化阶梯形式或行标准形式)，当满足以下条件时：

它处于行阶梯形式。

每个主元都是1。

主元是其所在列中唯一的非零项。

◇

简化行阶梯形式在第2.3.3节中将发挥重要作用，因为它允许我们以直接的方式确定线性方程组的通解。

注释(高斯消元法)。高斯消元法是一种算法，通过执行基本变换将线性方程组转换为简化行阶梯形式。◇

例2.8 (减1技巧)

让我们重新审视矩阵(2.49)，它已经处于简化REF形式：

$$\begin{pmatrix} 1 & 3 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 9 \\ 0 & 0 & 0 & 1 & -4 \end{pmatrix} A = \begin{pmatrix} 0 & 0 & 1 & 0 & 9 \\ 0 & 0 & 0 & 1 & -4 \end{pmatrix} \quad (2.53)$$

我们现在通过在对角线上主元缺失的位置添加形式为(2.52)的行，将此矩阵增广为 5×5 矩阵，得到：

$$\begin{pmatrix} 1 & 3 & 0 & 0 & 3 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \tilde{A} = \begin{pmatrix} 0 & 0 & 1 & 0 & 9 \\ 0 & 0 & 0 & 1 & -4 \end{pmatrix} \quad (2.54)$$

从这种形式，我们可以通过取 $\tilde{A}$ 中在对角线上包含 -1 的列立即读出 $Ax = 0$ 的解：

$$\begin{pmatrix} 3 \\ -1 \\ 0 \end{pmatrix}$$

$$x \in \mathbb{R}^5 : x = \lambda_1 \begin{pmatrix} 3 \\ -1 \\ 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \lambda_1, \lambda_2 \in \mathbb{R}, \quad (2.55)$$

这与我们通过“洞察”在(2.50)中得到的解完全相同。

计算逆矩阵

为了计算 $A \in \mathbb{R}^{(n \times n)}$ 的逆矩阵 A^{-1} ，我们需要找到一个满足 $AX = I_n$ 的矩阵 X 。那么， $X = A^{-1}$ 。我们可以将此写成一组同时线性方程 $AX = I_n$ ，其中我们求解 $X = [x_1 \mid \cdots \mid x_n]$ 。我们使用增广矩阵记号来紧凑表示这组线性方程组，得到：

$$A \mid I_n \rightsquigarrow \cdots \rightsquigarrow I_n \mid A^{-1}. \quad (2.56)$$

这意味着如果我们将增广方程组转换为简化行阶梯形式，我们可以在方程组的右侧读出逆矩阵。因此，确定矩阵的逆等同于求解线性方程组。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由剑桥大学出版社出版 (2020).]

[34] 线性代数

例 2.9 (通过高斯消元法计算逆矩阵)

要确定以下矩阵的逆矩阵：

$$\begin{pmatrix} 1 & 0 & 2 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} A = \begin{pmatrix} 1 & 2 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (2.57)$$

我们写出增广矩阵：


```
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
```

并使用高斯消元法将其化为简化行阶梯形式：

```
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
```

这样所需的逆矩阵就由其右侧给出：

```
- 1 2 - 2 2 A^{-1} = 1 - 1 2 - 2. (2.58) - - 1 1 1 1 - 1 0 - 1 2
```

我们可以通过进行 $A^{-1}A$ 的乘法运算来验证(2.58)确实是逆矩阵，观察到我们恢复了 I_4 。

2.3.4 求解线性方程组的算法

在下文中，我们简要讨论求解形如 $Ax = b$ 的线性方程组的方法。我们假设解存在。如果没有解，我们请求助于近似解，这在本章中不涉及。解决近似问题的一种方式是使用线性回归的方法，我们将在第9章中详细讨论。

在特殊情况下，我们可能能够确定逆矩阵 A^{-1} ，这样 $Ax = b$ 的解就给出为 $x = A^{-1}b$ 。然而，这只有在 A 是方阵且可逆时才可能，而这通常不是情况。否则，在温和的假设下（即 A 需要有线性无关的列），我们可以使用变换：

$$A^T A x = A^T b \Leftrightarrow x = (A^T A)^{-1} A^T b \quad (2.59)$$

[《机器学习数学基础》草稿 (2023-10-18)。反馈：<https://mml-book.com>] 2.4 向量空间 35

并使用Moore-Penrose伪逆 $(A^T A)^{-1} A^T$ 来确定解(2.59)，它解决了 $Ax = b$ ，这也对应于最小范数最小二乘解。这种方法的缺点是它需要大量计算来进行矩阵-矩阵乘积和计算 $A^T A$ 的逆。此外，出于数值精度的原因，通常不建议计算逆或伪逆。因此，在下文中，我们简要讨论求解线性方程组的替代方法。

高斯消元法在计算行列式（第4.1节）、检查向量集是否线性无关（第2.5节）、计算矩阵的逆（第2.2.2节）、计算矩阵的秩（第2.6.2节）以及确定向量空间的基（第2.6.1节）时起着重要作用。高斯消元法是解决有数千个变量的线性方程组的直观且构造性的方法。然而，对于有数百万变量的系统，它是不实用的，因为所需的算术运算数量与同时方程的数量成三次方关系。

在实践中，许多线性方程的系统是间接求解的，要么通过静态迭代方法，如Richardson方法、Jacobi方法、Gauß-Seidel方法和逐次超松弛方法，要么通过Krylov子空间方法，如共轭梯度法、广义最小残差法或双共轭梯度法。我们参考Stoer和Burlirsch (2002)、Strang (2003)以及Liesen和Mehrmann (2015)的书籍以获得更多细节。

设 x^* 是 $Ax = b$ 的解。这些迭代方法的关键思想是建立形式为

$$x^{(k+1)} = Cx^{(k)} + d \quad (2.60)$$

的迭代，对于合适的 C 和 d ，它在每次迭代中减少残差误差 $\|x^{(k+1)} - x\|$ 并收敛到 x 。我们将在第3.1节中介绍范数 $\|\cdot\|$ ，它允许我们计算向量之间的相似性。

2.4 向量空间

到目前为止，我们已经研究了线性方程组以及如何求解它们（第2.3节）。我们看到线性方程组可以用矩阵-向量记号(2.10)紧凑地表示。在下文中，我们将更仔细地研究向量空间，即向量所在的结构化空间。

在本章开始时，我们非正式地将向量描述为可以相加并与标量相乘的对象，并且它们仍然是相同类型的对象。现在，我们准备将此正式化，我们将从引入群的概念开始，群是一组元素和在这些元素上定义的运算，它保持集合的某种结构完整。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由剑桥大学出版社出版 (2020).]

[36] 线性代数

2.4.1 群

群在计算机科学中起着重要作用。除了为集合上的运算提供基本框架外，它们在密码学、编码理论和图形学中被大量使用。

定义 2.7 (群)。 考虑一个集合 G 和在 G 上定义的运算 $\otimes: G \times G \rightarrow G$ 。如果以下条件成立，则 $G := (G, \otimes)$ 被称为群：

1. G 在 $\otimes$ 下的封闭性： $\forall x, y \in G: x \otimes y \in G$
2. 结合律： $\forall x, y, z \in G: (x \otimes y) \otimes z = x \otimes (y \otimes z)$
3. 中性元素： $\exists e \in G \forall x \in G: x \otimes e = x$ 且 $e \otimes x = x$
4. 逆元素： $\forall x \in G \exists y \in G: x \otimes y = e$ 且 $y \otimes x = e$ ，其中 e 是中性元素。我们通常写 x^{-1} 来表示 x 的逆元素。

注释。 逆元素是相对于运算 $\otimes$ 定义的，不一定意味着 $1/x$ 。

如果另外 $\forall x, y \in G: x \otimes y = y \otimes x$ ，则 $G = (G, \otimes)$ 是 Abel 群（可交换的）。

例 2.10 (群)

让我们看一些集合及其相关运算的例子，看看它们是否为群：

$(\mathbb{Z}, +)$ 是一个 Abelian group。

$(\mathbb{N}[0], +)$ 不是群：虽然 $(\mathbb{N}[0], +)$ 拥有中性元素 (0) ，但缺少逆元素。 $(\mathbb{Z}, \cdot)$ 不是群：虽然 $(\mathbb{Z}, \cdot)$ 包含中性元素 (1) ，但对于任何 $z \in \mathbb{Z}, z \neq \pm 1$ ，都缺少逆元素。 $(\mathbb{R}, \cdot)$ 不是群，因为 0 没有逆元素。 $(\mathbb{R} \setminus \{0\}, \cdot)$ 是 Abelian 的。

$(\mathbb{R}^n, +)$ 是 Abelian 的，如果 $+$ 按分量定义，即：

$$(x_1, \dots, x[n]) + (y_1, \dots, y[n]) = (x_1 + y_1, \dots, x[n] + y[n]). \quad [(2.61)]$$

那么, $-(x, \dots, x) := (-x, \dots, -x)$ 是逆元素, $1[n] 1[n] e = (0, \dots, 0)$ 是中性元素。 $([m] \times [n] \mathbb{R}, +)$, 即 $m \times n$ 矩阵的集合是 Abelian 的 (按 [(2.61)] 中定义的分量加法)。

让我们更仔细地看看 $[n] \times [n] (\mathbb{R}, \cdot)$, 即具有矩阵乘法 (按 [(2.13)] 中定义) 的 $n \times n$ 矩阵集合。

- 封闭性和结合律直接从矩阵乘法的定义得出。
- 中性元素: 单位矩阵 $I[n]$ 是 $[n] \times [n] (\mathbb{R}, \cdot)$ 中关于矩阵乘法 “ $\cdot$ ” 的中性元素。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] [<https://mml-book.com>][.] [2.4 Vector Spaces]

- - 逆元素: 如果逆元素存在 (A 是正则的), 那么 A 是 $[n] \times [n] [n] \times [n] A \in \mathbb{R}$ 的逆元素, 在这种情况下 (且仅在这种情况下) $(\mathbb{R}, \cdot)$ 是一个群, 称为 *general linear group*。

Definition 2.8 (General Linear Group). 正则 (可逆) 矩阵 $[n] \times [n] A \in \mathbb{R}$ 的集合关于 [(2.13)] 中定义的矩阵乘法构成一个群, 称为 *general linear group* $GL(n, \mathbb{R})$ 。然而, 由于矩阵乘法不满足交换律, 该群不是 Abelian 的。

2.4.2 Vector Spaces

当我们讨论群时，我们考虑的是集合 G 和 G 上的内运算，即仅对 G 中元素进行运算的映射 $G \times G \rightarrow G$ 。在下面的内容中，我们将考虑除了内运算 $+$ 之外还包含外运算 $\cdot$ 的集合，即向量 $x \in G$ 与标量 $\lambda \in \mathbb{R}$ 的乘法。我们可以将内运算视为一种加法，将外运算视为一种缩放。注意内/外运算与内/外积无关。

Definition 2.9 (Vector Space). 实值 *vector space* $V = (V, +, \cdot)$ 是一个集合 V 具有两个运算

$$+ : V \times V \rightarrow V \quad (2.62) \quad \cdot : \mathbb{R} \times V \rightarrow V \quad (2.63)$$

其中

1. $(V, +)$ 是一个 Abelian group

2. 分配律：

$$1. \forall \lambda \in \mathbb{R}, x, y \in V : \lambda \cdot (x + y) = \lambda \cdot x + \lambda \cdot y \quad 2. \forall \lambda, \psi \in \mathbb{R}, x \in V : (\lambda + \psi) \cdot x = \lambda \cdot x + \psi \cdot x$$

3. 结合律（外运算）： $\forall \lambda, \psi \in \mathbb{R}, x \in V : \lambda \cdot (\psi \cdot x) = (\lambda\psi) \cdot x$ 4. 关于外运算的中性元素： $\forall x \in V : 1 \cdot x = x$

元素 $x \in V$ 称为 *vectors*。 $(V, +)$ 的中性元素是零向量 $[T] 0 = [0, \dots, 0]$ ，内运算 $+$ 称为 *vector addition*。元素 $\lambda \in \mathbb{R}$ 称为 *scalars*，外运算 $\cdot$ 是 *multiplication by scalars*。注意标量积是不同的概念，我们将在第 [3.2] 节中讨论。

Remark. [n] “向量乘法” ab ， $a, b \in \mathbb{R}$ 是未定义的。理论上，我们可以定义分量乘法，使得 $c = ab$ 其中 $c[j] = a[j]b[j]$ 。这种“数组乘法”在许多编程语言中很常见，但使用标准矩阵乘法规则在数学上意义有限：通过将向量视为 $n \times 1$ 矩阵

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

（我们通常这样做），我们可以使用 [(2.13)] 中定义的矩阵乘法。然而，这样向量的维度不匹配。只有以下向量乘法是定义的： $[T] [n] [\times] [n] ab \in \mathbb{R}$ (*outer product*)， $[T] ab \in \mathbb{R}$ (*inner/scalar/dot product*)。◇

Example 2.11 (Vector Spaces)

让我们看一些重要的例子：

$V[n] = \mathbb{R}$, $n \in \mathbb{N}$ 是一个 vector space，运算定义如下： - 加法： $x + y = (x_1, \dots, x[n]) + (y_1, \dots, y[n]) = (x_1 + y_1, \dots, x[n] + y[n])$ 对于所有 $[n] x, y \in \mathbb{R}$

- 标量乘法： $\lambda x = \lambda(x_1, \dots, x[n]) = (\lambda x_1, \dots, \lambda x[n])$ 对于所有 $[n] \lambda \in \mathbb{R}, x \in \mathbb{R}$

$V[m] [\times] [n] = \mathbb{R}$, $m, n \in \mathbb{N}$ 是一个 vector space，具有

$a + b \cdot \dots \cdot a + b$ [11] [11] [1][n] [1][n] - 加法： $A + B = \dots$ 对所有 $A, B \in V$ 按分量定义。 $\cdot am + bm \cdot \dots \cdot a[mn] + b[mn]$

$\lambda a \cdot \cdot \cdot \lambda a$ [11] [1][n] - 标量乘法: $\lambda A = \dots$ 如第 [2.2] 节中定义。 $\cdot \lambda a m \cdot \cdot \cdot \lambda a [mn]$

记住 $[m] [\times] [n] [mn] \mathbb{R}$ 等价于 $\mathbb{R}$ 。

$V = \mathbb{C}$, 具有复数加法的标准定义。

Remark. 在下面的内容中, 当 $+$ 和 $\cdot$ 是标准向量加法和标量乘法时, 我们将用 V 来表示 vector space $(V, +, \cdot)$ 。此外, 我们将使用记号 $x \in V$ 表示 V 中的向量以简化记号。◇

Remark. $[n] [n] [\times] 1 [1] [\times] [n]$ vector spaces $\mathbb{R}, \mathbb{R}, \mathbb{R}$ 只是在向量书写方式上不同。在下面的内容中, 我们不区分 $[n] [n] [\times] 1 \mathbb{R}$ 和 $\mathbb{R}$, 这允许我们将 n 元组写成列向量

$x \ 1$

$x = \dots [(2.64)] \cdot x[n]$

这简化了关于 vector space 运算的记号。然而, 我们确实区分 $[n] [\times] 1 [1] [\times] [n] \mathbb{R}$ 和 $\mathbb{R}$ (行向量) 以避免混淆

融合与矩阵乘法。默认情况下, 我们用 x 表示列向量, 行向量用 x^T 表示, 即 x 的转置。◇

2.4.3 向量子空间

在下面的内容中, 我们将介绍向量子空间。直观地说, 它们是包含在原始向量空间中的集合, 具有这样的性质: 当我们将该子空间内的元素执行向量空间运算时, 我们永远不会离开它。从这个意义上说, 它们是”封闭的”。向量子空间是机器学习中的一个关键概念。例如, 第10章演示了如何使用向量子空间进行降维。

定义 2.10 (向量子空间)。设 $V = (V, +, \cdot)$ 是一个向量空间, $U \subseteq V$, $U \neq \emptyset$ 。则 $U = (U, +, \cdot)$ 称为 V 的向量子空间 (或线性子空间), 如果 U 是一个向量空间, 其向量空间运算 $+$ 和 $\cdot$ 限制在 $U \times U$ 和 $\mathbb{R} \times U$ 上。我们写作 $U \subseteq V$ 来表示 V 的子空间 U 。

如果 $U \subseteq V$ 且 V 是向量空间, 那么 U 自然地直接从 V 继承许多性质, 因为它们对所有 $x \in V$ 都成立, 特别是对所有 $x \in U \subseteq V$ 。这包括 Abelian 群的性质、分配律、结合律和零元素。要确定 $(U, +, \cdot)$ 是否为 V 的子空间, 我们仍需证明:

1. $U \neq \emptyset$, 特别地: $0 \in U$
2. U 的封闭性:
 - a. 关于外运算: $\forall \lambda \in \mathbb{R} \forall x \in U: \lambda x \in U$
 - b. 关于内运算: $\forall x, y \in U: x + y \in U$

例 2.12 (向量子空间)

让我们看一些例子:

对于每个向量空间 V , 平凡子空间是 V 本身和 $\{0\}$ 。

图2.6中只有例子 D 是 $\mathbb{R}^2$ 的子空间 (使用通常的内/外运算)。在 A 和 C 中, 封闭性质被违反; B 不包含 0。

齐次线性方程组 $Ax = 0$ 的解集, 其中有 n 个未知数 $x = [x_1, \dots, x_n]^T$, 是 $\mathbb{R}^n$ 的子空间。

非齐次线性方程组 $Ax = b$, $b \neq 0$ 的解不是 $\mathbb{R}^n$ 的子空间。

任意多个子空间的交集本身也是子空间。

图 2.6 并非 $\mathbb{R}^2$ 的所有子集都是子空间。在 A 和 C 中，封闭性质被违反；B 不包含 0。只有 D 是子空间。

注记。 每个子空间 $U \subseteq (\mathbb{R}^n, +, \cdot)$ 都是齐次线性方程组 $Ax = 0$ 对 $x \in \mathbb{R}^n$ 的解空间。◇

2.5 线性无关性

在下面的内容中，我们将仔细研究我们可以用向量（向量空间的元素）做什么。特别地，我们可以将向量相加并用标量乘以它们。封闭性质保证我们最终得到同一向量空间中的另一个向量。我们可以找到一组向量，通过将它们相加和缩放，我们可以表示向量空间中的每个向量。这组向量是一个基，我们将在第2.6.1节中讨论它们。在此之前，我们需要引入线性组合和线性无关性的概念。

定义 2.11（线性组合）。考虑向量空间 V 和有限个向量 $x_1, \dots, x_k \in V$ 。则每个形式为

$$v = \lambda_1 x_1 + \dots + \lambda_k x_k = \sum_{i=1}^k \lambda_i x_i \in V \quad (2.65)$$

的 $v \in V$ ，其中 $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ ，称为向量 $x_1, \dots, x_k$ 的线性组合。

零向量总是可以写成 k 个向量 $x_1, \dots, x_k$ 的线性组合，因为 $0 = \sum_{i=1}^k 0 \cdot x_i$ 总是成立的。在下面的内容中，我们对表示 0 的向量集合的非平凡线性组合感兴趣，即向量 $x_1, \dots, x_k$ 的线性组合，其中 (2.65) 中并非所有系数 λ_i 都为 0 。

定义 2.12（线性（无）关性）。让我们考虑向量空间 V ，其中 $k \in \mathbb{N}$ 且 $x_1, \dots, x_k \in V$ 。如果存在非平凡线性组合，使得 $0 = \sum_{i=1}^k \lambda_i x_i$ 且至少有一个 $\lambda_i \neq 0$ ，则向量 $x_1, \dots, x_k$ 是线性相关的。如果只有平凡解存在，即 $\lambda_1 = \dots = \lambda_k = 0$ ，则向量 $x_1, \dots, x_k$ 是线性无关的。

线性无关性是线性代数中最重要的概念之一。直观地说，一组线性无关向量由没有冗余的向量组成，即如果我们从集合中移除任何一个向量，我们将失去某些东西。在接下来的章节中，我们将更正式地阐述这种直觉。

例 2.13（线性相关向量）

一个地理例子可能有助于澄清线性无关性的概念。一个在内罗毕（肯尼亚）的人描述基加利（卢旺达）在哪里可能会说：“你可以先向西北方向走506公里到坎帕拉（乌干达），然后向西南方向走374公里到达基加利。”这是描述基加利位置的充分信息，因为地理坐标系可以被视为一个二维向量空间（忽略海拔和地球的曲面）。这个人可能会补充说：“它距离这里大约751公里向西。”尽管这最后的陈述是正确的，但它对于

在基加利给定的前述信息中找到基加利（见图[2.7]的说明）。在这个例子中，“506公里西北”向量（蓝色）和“374公里西南”向量（紫色）是线性无关的。这意味着西南向量不能用西北向量来描述，反之亦然。然而，第三个“751公里西”向量（黑色）是其他两个向量的线性组合，它使得向量集合线性相关。等价地，给定“751公里西”和“374公里西南”可以线性组合得到“506公里西北”。



坎帕拉 [图 2.7]

506 [[地理示例]]

w 北 [近似到] 西 公里 [(粗略地)]

南 [西 [[基本方向]]]

内罗毕 [线性]

公里 [[相关向量]]

374 [在一个] 751 公里 西

西 [[二维]]

w [[空间（平面）。]]

基加利

南

公里

注. 以下性质对于确定向量是否线性无关很有用:

k个向量要么线性相关, 要么线性无关。没有第三种选择。

如果向量 $x_1, \dots, x[k]$ 中至少有一个是 0, 那么它们是线性相关的。如果两个向量相同, 情况也是如此。

向量 $\{x_1, \dots, x[k] : x[i] \neq 0, i = 1, \dots, k\}, k \geq 2$, 是线性相关的当且仅当其中 (至少) 一个是其他向量的线性组合。特别地, 如果一个向量是另一个向量的倍数, 即 $x[i] = \lambda x[j], \lambda \in \mathbb{R}$, 那么集合 $\{x_1, \dots, x[k] : x[i] \neq 0, i = 1, \dots, k\}$ 是线性相关的。

检查向量 $x_1, \dots, x[k] \in V$ 是否线性无关的实用方法是使用高斯消元法: 将所有向量写成矩阵 A 的列, 并执行高斯消元直到矩阵达到行阶梯形式 (这里不需要简化行阶梯形式):

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[42] [[线性代数]]

- 主元列表示与左侧向量线性无关的向量。注意在构建矩阵时存在向量的排序。
- 非主元列可以表示为其左侧主元列的线性组合。例如, 行阶梯形式

1 3 0 (2.66) 0 0 2

告诉我们第一列和第三列是主元列。第二列是非主元列, 因为它是第一列的三倍。

所有列向量都线性无关当且仅当所有列都是主元列。如果至少存在一个非主元列, 则列 (因此相应的向量) 是线性相关的。

◇

例 2.14

考虑 $\mathbb{R}[4]$ 中的

$$\begin{pmatrix} 1 & 1 & -1 & 2 & 1 & -2 \end{pmatrix}$$

$x_1 = , x_2 = , x_3 = .$ (2.67) 3 0 1 4 2 1

为了检查它们是否线性相关, 我们遵循一般方法并求解

$$\begin{pmatrix} 1 & 1 & -1 & 2 & 1 & -2 \end{pmatrix}$$

$\lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 = \lambda_1 + \lambda_2 + \lambda_3 = 0$ (2.68) 3 0 1 4 2 1

对于 $\lambda_1, \dots, \lambda_3$ 。我们将向量 $x[i], i = 1, 2, 3$ 写成矩阵的列, 并应用基本行变换直到我们识别出主元列:

$$\begin{pmatrix} 1 & 1 & -1 & 1 & 1 & -1 \\ 2 & 1 & -2 & 0 & 1 & 0 \end{pmatrix} \rightarrow \dots \rightarrow \dots \quad (2.69)$$

3 0 1 0 0 1

4 2 1 0 0 0

这里，矩阵的每一列都是主元列。因此，不存在非平凡解，我们需要 $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0$ 来求解方程组。因此，向量 x_1, x_2, x_3 是线性无关的。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] <https://mml-book.com> [.] [2.5 线性无关] [[43]]

注. 考虑一个向量空间 V ，其中有 k 个线性无关向量 $b_1, \dots, b[k]$ 和 m 个线性组合

$$[k] \times X = \lambda b, 1 \leq i \leq [i]=1]$$

.. [(2.70) .]

$$[k] \times X = \lambda b, [m] [im] [i] [i]=1]$$

定义 $B = [b_1, \dots, b[k]]$ 为列是线性无关向量 $b_1, \dots, b[k]$ 的矩阵，我们可以写成

$$\lambda [1][i]$$

$$x[j] = B\lambda[j], \lambda[j] = \dots, j = 1, \dots, m, (2.71) \cdot \lambda [k]$$

这是一种更紧凑的形式。

我们想要测试 $x_1, \dots, x[m]$ 是否线性无关。为此，我们遵循测试 $P [m] \psi x = 0$ 的一般方法。 $[i]=1] [i] [i]$

用 [(2.71)], 我们得到

$$[m] [m] [m] X X X \psi x = \psi B \lambda = B \psi \lambda. (2.72) [i] [i] [i] [i] [i] [i]=1] [i]=1] [i]=1]$$

这意味着 $\{x_1, \dots, x[m]\}$ 线性无关当且仅当列向量 $\{\lambda_1, \dots, \lambda[m]\}$ 线性无关。

◇

注. 在向量空间 V 中，如果 $m > k$ ，那么 k 个向量 $x_1, \dots, x[k]$ 的 m 个线性组合是线性相关的。 [◇]

例 2.15

考虑一组线性无关向量 $b_1, b_2, b_3, b_4 \in \mathbb{R}[n]$ 和

$$x_1 = b_1 - 2b_2 + b_3 - b_4$$

$$x_2 = -4b_1 - 2b_2 + 4b_4, (2.73) x_3 = 2b_1 + 3b_2 - b_3 - 3b_4, x_4 = 17b_1 - 10b_2 + 11b_3 + b_4$$

向量 $x_1, \dots, x_4 \in \mathbb{R}[n]$ 是线性无关的吗？为了回答这个问题，我们研究列向量

$$\begin{pmatrix} 1 & 4 & -2 & 17 \\ -2 & -2 & 3 & -10 \\ , & , & , & (2.74) \\ 1 & 0 & -1 & 11 \\ -1 & 4 & -3 & 1 \end{pmatrix}$$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[44] [[线性代数]]

是否线性无关。对应的线性方程组的系数矩阵的简化行阶梯形式

$$1 - 4 \ 2 \ 17 \ -2 \ -2 \ 3 \ -10 \ A = (2.75) \ 1 \ 0 \ -1 \ 11 \ -1 \ 4 \ -3 \ 1$$

给出为

$$1 \ 0 \ 0 \ -7 \ 0 \ 1 \ 0 \ -15 \ . \ (2.76) \ 0 \ 0 \ 1 \ 18 \ 0 \ 0 \ 0 \ 0$$

我们看到对应的线性方程组是非平凡可解

因此, $x_1, \dots, x_4$ 线性相关, 因为 x_4 可以表示为 $x_1, \dots, x_3$ 的线性组合。

2.6 基和秩

在向量空间 V 中，我们特别关注具有以下性质的向量集合 A ： V 中的任何向量 v 都可以通过 A 中向量的线性组合得到。这些向量是特殊的向量，在下面我们将对它们进行刻画。

2.6.1 生成集和基

定义 2.13 (生成集和张成)。考虑向量空间 $V = (V, +, \cdot)$ 和向量集合 $A = \{x_1, \dots, x_k\} \subseteq V$ 。如果 V 中的每个向量 v 都可以表示为 $x_1, \dots, x_k$ 的线性组合，则称 A 为 V 的生成集。 A 中所有向量的线性组合的集合称为 A 的张成。如果 A 张成向量空间 V ，我们记作 $V = \text{span}[A]$ 或 $V = \text{span}[x_1, \dots, x_k]$ 。

生成集是张成向量（子）空间的向量集，即每个向量都可以表示为生成集中向量的线性组合。现在，我们将更加具体地刻画张成向量（子）空间的最小生成集。

定义 2.14 (基)。考虑向量空间 $V = (V, +, \cdot)$ 和 $A \subseteq V$ 。如果不存在更小的集合 $\bar{A} \subsetneq A \subseteq V$ 张成 V ，则称 V 的生成集 A 是最小的。 V 的每个线性无关的生成集都是最小的，称为 V 的基。

设 $V = (V, +, \cdot)$ 是向量空间， $B \subseteq V$ ， $B \neq \emptyset$ 。则以下陈述等价：

基是最小的生成集和最大的线性无关向量集。

- B 是 V 的基。
- B 是最小生成集。
- B 是 V 中最大的线性无关向量集，即向这个集合添加任何其他向量都会使其线性相关。
- V 中的每个向量 x 都是 B 中向量的线性组合，并且每个线性组合都是唯一的，即

$$x = \sum_{i=1}^k \lambda_i b_i = \sum_{i=1}^k \psi_i b_i \quad (2.77)$$

其中 $\lambda_i, \psi_i \in \mathbb{R}$ ， $b_i \in B$ ，则有 $\lambda_i = \psi_i$ ， $i = 1, \dots, k$ 。

例 2.16

在 $\mathbb{R}^3$ 中，标准基是

$$B = \{[1;0;0], [0;1;0], [0;0;1]\} \quad (2.78)$$

$\mathbb{R}^3$ 中的不同基有

$$B_1 = \{[1;0;0], [1;1;0], [1;1;1]\}, B_2 = \{[0.5;0.8;0.4], [1.8;0.3;0.3], [-2.2;-1.3;3.5]\} \quad (2.79)$$

集合

$$A = \{[1;3;4], [2;0;2], [1;0;-4]\} \quad (2.80)$$

是线性无关的，但不是 $\mathbb{R}^4$ 的生成集（也不是基）：例如，向量 $[1,0,0,0]^T$ 不能通过 A 中元素的线性组合得到。

注释。每个向量空间 V 都有一个基 B 。前面的例子表明，向量空间 V 可能有许多基，即基不是唯一的。然而，所有基都有相同数量的元素，即基向量。◇

我们只考虑有限维向量空间 V 。在这种情况下， V 的维数是 V 的基向量的数量，记作 $\dim(V)$ 。如果 $U \subseteq V$ 是 V 的子空间，则 $\dim(U) \leq \dim(V)$ ，且 $\dim(U) = \dim(V)$ 当且仅当 $U = V$ 。直观地，向量空间的维数可以看作该向量空间中独立方向的数量。

注释。向量空间的维数不一定等于向量中元素的数量。例如，向量空间 $V = \text{span}[[0;1]]$ 是一维的，尽管基向量有两个元素。◇

注释。子空间 $U = \text{span}[x_1, \dots, x_m] \subseteq \mathbb{R}^n$ 的基可以通过执行以下步骤找到：

1. 将张成向量写成矩阵 A 的列
2. 确定 A 的行阶梯形
3. 与主元列相关联的张成向量是 U 的基

◇

例 2.17（确定基）

对于向量空间 $U \subseteq \mathbb{R}^5$ ，由向量

$$x_1 = [1; -1; -1; -1], x_2 = [2; 1; 2; -2], x_3 = [3; 3; 5; -3], x_4 = [-1; -5; -6; 1] \in \mathbb{R}^5 \quad (2.81)$$

张成，我们想要找出哪些向量 $x_1, \dots, x_4$ 是 U 的基。

为此，我们需要检查 $x_1, \dots, x_4$ 是否线性无关。因此，我们需要求解

$$\sum_{i=1}^4 \lambda_i x_i = 0 \quad (2.82)$$

这导致齐次线性方程组，其矩阵为

$$[x_1, x_2, x_3, x_4] = \begin{bmatrix} 1 & 2 & 3 & -1 \\ -1 & 1 & 3 & -5 \\ -1 & 2 & 5 & -6 \\ -1 & -2 & -3 & 1 \end{bmatrix} \quad (2.83)$$

通过线性方程组的基本变换规则，我们得到行阶梯形

$$\begin{bmatrix} 1 & 2 & 3 & -1 \\ -1 & 1 & 3 & -5 \\ -1 & 2 & 5 & -6 \\ -1 & -2 & -3 & 1 \end{bmatrix} \Rightarrow \dots \Rightarrow \begin{bmatrix} 1 & 2 & 3 & -1 \\ 0 & 1 & 2 & -2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

由于主元列表示哪一组向量线性无关，从行阶梯形可以看出 x_1, x_2, x_4 线性无关（因为线性方程组 $\lambda_1 x_1 + \lambda_2 x_2 + \lambda_4 x_4 = 0$ 只能用 $\lambda_1 = \lambda_2 = \lambda_4 = 0$ 求解）。因此， $\{x_1, x_2, x_4\}$ 是 U 的基。

2.6.2 秩(Rank)

矩阵 $A \in \mathbb{R}^{(m \times n)}$ 的线性无关列数等于线性无关行数，称为 A 的秩，记作 $\text{rk}(A)$ 。

备注。矩阵的秩具有一些重要性质：

$\text{rk}(A^T) = \text{rk}(A)$ ，即列秩等于行秩。

$A \in \mathbb{R}^{(m \times n)}$ 的列张成子空间 $U \subseteq \mathbb{R}^n$ ，其中 $\dim(U) = \text{rk}(A)$ 。稍后我们将称此子空间为 *像或值域*。通过对 A 应用高斯消元法识别主元列，可以找到 U 的基。

$A \in \mathbb{R}^{(m \times n)}$ 的行张成子空间 $W \subseteq \mathbb{R}^m$ ，其中 $\dim(W) = \text{rk}(A)$ 。通过对 A^T 应用高斯消元法，可以找到 W 的基。

对于所有 $A \in \mathbb{R}^{(n \times n)}$ ， A 是正则的(可逆的)当且仅当 $\text{rk}(A) = n$ 。

对于所有 $A \in \mathbb{R}^{(m \times n)}$ 和所有 $b \in \mathbb{R}^m$ ，线性方程组 $Ax = b$ 有解当且仅当 $\text{rk}(A) = \text{rk}(A|b)$ ，其中 $A|b$ 表示增广矩阵。

对于 $A \in \mathbb{R}^{(m \times n)}$ ， $Ax = 0$ 的解空间的维数为 $n - \text{rk}(A)$ 。稍后，我们将称此子空间为 *核或零空间*。

如果矩阵 $A \in \mathbb{R}^{(m \times n)}$ 的秩等于相同维数矩阵可能的最大秩，则称该矩阵具有 *满秩*。这意味着满秩矩阵的秩是行数和列数中的较小者，即 $\text{rk}(A) = \min(m, n)$ 。如果矩阵不具有满秩，则称其为 *秩亏缺*的。

◇

例 2.18 (秩)

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

A 有两个线性无关的行/列，所以 $\text{rk}(A) = 2$ 。

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

我们使用高斯消元法来确定秩：

$$\begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ -2 & -3 & 1 \end{bmatrix} \rightsquigarrow \cdots \rightsquigarrow \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 3 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.84)$$

这里，我们看到线性无关行数和列数是 2，因此 $\text{rk}(A) = 2$ 。

2.7 线性映射

在接下来的内容中，我们将研究保持向量空间结构的映射，这将使我们能够定义坐标的概念。

在本章开始时，我们说向量是可以相加并与标量相乘的对象，结果对象仍然是向量。我们希望在应用映射时保持这个性质：

考虑两个实向量空间 V, W 。如果映射 $\Phi : V \rightarrow W$ 满足

$$\Phi(x + y) = \Phi(x) + \Phi(y) \quad (2.85) \quad \Phi(\lambda x) = \lambda \Phi(x) \quad (2.86)$$

对于所有 $x, y \in V$ 和 $\lambda \in \mathbb{R}$ ，则该映射保持向量空间的结构。我们可以在以下定义中总结这一点：

定义 2.15 (线性映射)。对于向量空间 V, W ，映射 $\Phi : V \rightarrow W$ 称为 *线性映射* (或 *向量空间同态* / *线性变换*)，如果

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

事实证明，我们可以将线性映射表示为矩阵 (第 2.7.1 节)。回想一下，我们也可以将向量集合作为矩阵的列来收集。在处理矩阵时，我们必须记住矩阵代表什么：线性映射还是向量的集合。我们将在第 4 章中看到更多关于线性映射的内容。在继续之前，我们将简要介绍特殊映射。

定义 2.16 (单射、满射、双射)。考虑映射 $\Phi : V \rightarrow W$ ，其中 V, W 可以是任意集合。则 Φ 称为

单射，如果 $\forall x, y \in V : \Phi(x) = \Phi(y) \implies x = y$ 。*满射*，如果 $\Phi(V) = W$ 。*双射*，如果它既是单射又是满射。

如果 Φ 是满射的，那么 W 中的每个元素都可以通过使用 Φ 从 V “到达”。双射的 Φ 可以被“撤销”，即存在映射 $\Psi : W \rightarrow V$ 使得 $\Psi \circ \Phi(x) = x$ 。这个映射 Ψ 称为 Φ 的逆，通常记作 Φ^{-1} 。

根据这些定义，我们介绍向量空间 V 和 W 之间线性映射的以下特殊情况：

同构： $\Phi : V \rightarrow W$ 线性且双射 *自同态*： $\Phi : V \rightarrow V$ 线性 *自同构*： $\Phi : V \rightarrow V$ 线性且双射

我们定义 $\text{id}_V : V \rightarrow V, x \mapsto x$ 为 V 中的 *恒等映射* 或 *恒等自同构*。

例 2.19 (同态)

映射 $\Phi : \mathbb{R}^2 \rightarrow \mathbb{C}, \Phi([x_1; x_2]) = x_1 + ix_2$ ，是一个同态：

$$\Phi([x_1; x_2] + [y_1; y_2]) = (x_1 + y_1) + i(x_2 + y_2) = x_1 + ix_2 + y_1 + iy_2 = \Phi([x_1; x_2]) + \Phi([y_1; y_2])$$

$$\Phi(\lambda[x_1; x_2]) = \lambda x_1 + i\lambda x_2 = \lambda(x_1 + ix_2) = \lambda \Phi([x_1; x_2]) \quad (2.88)$$

这也证明了为什么复数可以表示为 $\mathbb{R}^2$ 中的元组：存在一个双射线性映射，将 $\mathbb{R}^2$ 中元组的逐元素加法转换为具有相应加法的复数集合。注意我们只证明了线性性，但没有证明双射性。

定理 2.17 (定理 3.59 在 [Axler (2015)] 中)。有限维向量空间 V 和 W 同构当且仅当 $\dim(V) = \dim(W)$ 。

定理 2.17 表明，在相同维数的两个向量空间之间存在线性双射映射。直观地说，这意味着相同维数的向量空间某种意义上是相同的東西，因为它们可以相互转换而不会产生任何损失。

定理 2.17 也为我们将 $\mathbb{R}^{(m \times n)}$ ($m \times n$ 矩阵的向量空间) 和 $\mathbb{R}^{(mn)}$ (长度为 mn 的向量的向量空间) 视为相同提供了理由，因为它们的维数都是 mn ，并且存在将一个转换为另一个的线性双射映射。

备注. 考虑向量空间 V, W, X 。那么：

对于线性映射 $\Phi : V \rightarrow W$ 和 $\Psi : W \rightarrow X$ ，映射 $\Psi \circ \Phi : V \rightarrow X$ 也是线性的。

如果 $\Phi : V \rightarrow W$ 是一个同构映射，那么 $\Phi : W \rightarrow V$ 也是一个同构映射。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[50] [*Linear Algebra*]]

图 2.8 由两组基向量定义的两个不同坐标系。根据选择的坐标系不同，向量 x 具有不同的坐标表示 x_b 和 x_e 。

如果 $\Phi : V \rightarrow W, \Psi : V \rightarrow W$ 是线性的，那么 $\Phi + \Psi$ 和 $\lambda\Phi, \lambda \in \mathbb{R}$ ，也是线性的。

◇

2.7.1 线性映射的矩阵表示

任何 n 维向量空间都与 $\mathbb{R}^n$ 同构（定理 2.17）。我们考虑 n 维向量空间 V 的一组基 $\{b_1, \dots, b_n\}$ 。在下面的讨论中，基向量的顺序将很重要。因此，我们写作

$$B = (b_1, \dots, b_n) \quad (2.89)$$

并称这个 n 元组为 V 的有序基。

备注（记号）. 我们现在到了记号变得有些复杂的地方。因此，我们在这里总结一些部分。 $B = (b_1, \dots, b_n)$ 是一个有序基， $B = \{b_1, \dots, b_n\}$ 是一个（无序）基， $B = [b_1, \dots, b_n]$ 是一个以向量 $b_1, \dots, b_n$ 为列的矩阵。◇

定义 2.18（坐标）. 考虑向量空间 V 和 V 的有序基 $B = (b_1, \dots, b_n)$ 。对于任何 $x \in V$ ，我们得到唯一的表示（线性组合）

$$x = \alpha_1 b_1 + \dots + \alpha_n b_n \quad (2.90)$$

其中 $\alpha_1, \dots, \alpha_n$ 是 x 关于 B 的坐标，向量

$$\alpha = [\alpha_1; \dots; \alpha_n] \in \mathbb{R}^n \quad (2.91)$$

是 x 关于有序基 B 的坐标向量/坐标表示。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>.] 2.7 Linear Mappings [51]

基实际上定义了一个坐标系。我们熟悉二维中的 Cartesian 坐标系，它由标准基向量 e_1, e_2 张成。在这个坐标系中，向量 $x \in \mathbb{R}^2$ 有一个表示，告诉我们如何线性组合 e_1 和 e_2 来得到 x 。但是， $\mathbb{R}^2$ 的任何基都定义了一个有效的坐标系，之前的同一个向量 x 在 (b_1, b_2) 基中可能有不同的坐标表示。在图 2.8 中， x 关于标准基 (e_1, e_2) 的坐标是 $[2, 2]^T$ 。但是，关于基 (b_1, b_2) ，同一个向量 x 表示为 $[1.09, 0.72]^T$ ，即 $x = 1.09b_1 + 0.72b_2$ 。在接下来的章节中，我们将发现如何获得这种表示。

例 2.20

让我们看一个几何向量 $x \in \mathbb{R}^2$ ，其关于 $\mathbb{R}^2$ 的标准基 (e_1, e_2) 的坐标为 $[2, 3]^T$ 。这意味着，我们可以写作 $x = 2e_1 + 3e_2$ 。但是，我们不必选择标准基来表示这个向量。如果我们使用基向量 $b_1 = [1, -1]^T$, $b_2 = [1, 1]^T$ ，我们将得到坐标 $[-1/2, 5/2]^T$ 来表示关于 (b_1, b_2) 的同一个向量（见图 2.9）。

图 2.9 向量 x 的不同坐标表示，取决于基的选择。 $x = 2e_1 + 3e_2 = -1/2 b_1 + 5/2 b_2$

备注. 对于 n 维向量空间 V 和 V 的有序基 B ，映射 $\Phi: \mathbb{R}^n \rightarrow V$, $\Phi(e_i) = b_i, i = 1, \dots, n$ 是线性的（并且由于定理 2.17 是一个同构映射），其中 $(e_1, \dots, e_n)$ 是 $\mathbb{R}^n$ 的标准基。◇

现在我们准备好在矩阵和有限维向量空间之间的线性映射之间建立明确的联系。

定义 2.19（变换矩阵）. 考虑向量空间 V, W 及其对应的（有序）基 $B = (b_1, \dots, b_n)$ 和 $C = (c_1, \dots, c_m)$ 。此外，我们考虑线性映射 $\Phi: V \rightarrow W$ 。对于 $j \in \{1, \dots, n\}$,

$$\Phi(b_j) = \alpha_{1j}c_1 + \cdots + \alpha_{mj}c_m = \sum_{i=1}^m \alpha_{ij}c_i \quad (2.92)$$

是 $\Phi(b_j)$ 关于 C 的唯一表示。那么，我们称 $m \times n$ 矩阵 A_Φ 为 Φ 的变换矩阵（关于 V 的有序基 B 和 W 的有序基 C ），其元素由下式给出：

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

$\Phi(b_j)$ 关于 W 的有序基 C 的坐标是 A_Φ 的第 j 列。考虑（有限维）向量空间 V, W 及其有序基 B, C 和线性映射 $\Phi: V \rightarrow W$ ，其

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[52] [[*Linear Algebra*]]

变换矩阵为 A_Φ 。如果 $\hat{x}$ 是 $x \in V$ 关于 B 的坐标向量， $\hat{y}$ 是 $y = \Phi(x) \in W$ 关于 C 的坐标向量，那么

$$\hat{y} = A_\Phi \hat{x}. \quad (2.94)$$

这意味着变换矩阵可以用来将关于 V 中有序基的坐标映射到关于 W 中有序基的坐标。

例 2.21（变换矩阵）

考虑同态 $\Phi: V \rightarrow W$ 以及 V 的有序基 $B = (b_1, \dots, b_3)$ 和 W 的有序基 $C = (c_1, \dots, c_4)$ 。给定

$$\Phi(b_1) = c_1 - c_2 + 3c_3 - c_4 \quad \Phi(b_2) = 2c_1 + c_2 + 7c_3 + 2c_4 \quad \Phi(b_3) = 3c_2 + c_3 + 4c_4 \quad (2.95)$$

关于 B 和 C 的变换矩阵 A_Φ 满足 $\Phi(b_k) = \sum_{i=1}^4 \alpha_{ik}c_i$ ， $k = 1, \dots, 3$ ，并给出为

$$A_\Phi = [\alpha_1, \alpha_2, \alpha_3] = [1 \ 2 \ 0; -1 \ 1 \ 3; 3 \ 7 \ 1; -1 \ 2 \ 4], \quad (2.96)$$

其中 $\alpha[j], j = 1, 2, 3$, 是 $\Phi(b[j])$ 相对于基 C 的坐标向量。

例 2.22 (向量的线性变换)

图 2.10 线性变换的三个例子，向量以点的形式显示：(a) 原始数据；(b) 旋转 45° ；(c) 沿水平轴拉伸；(d) 一般线性映射。



图 2.10 线性变换的三个例子，向量以点的形式显示

图 2.10 线性变换的三个例子，向量以点的形式显示

我们考虑 $\mathbb{R}^2$ 中一组向量的三种线性变换，变换矩阵为：

$$A_1 = [\cos(\pi/4) \ -\sin(\pi/4); \sin(\pi/4) \ \cos(\pi/4)], A_2 = [2 \ 0; 0 \ 1], A_3 = [1 \ 3; 2 \ -1] \quad (2.97)$$

图 2.10 给出了向量集合线性变换的三个例子。图 2.10(a) 显示了 $\mathbb{R}^2$ 中的 400 个向量，每个向量由对应 (x_1, x_2) 坐标处的点表示。这些向量排列成正方形。当我们使用 (2.97) 中的矩阵 A_1 对每个向量进行线性变换时，得到图 2.10(b) 中的旋转正方形。如果我们应用矩阵 A_2 表示的线性映射，我们得到图 2.10(c) 中的矩形，其中每个 x_1 坐标被拉伸 2 倍。图 2.10(d) 显示了图 2.10(a) 中的原始正方形使用 A_3 进行线性变换后的结果，这是反射、旋转和拉伸的组合。

2.7.2 基变换

在下面的内容中，我们将仔细研究当我们改变 V 和 W 中的基时，线性映射 $\Phi : V \rightarrow W$ 的变换矩阵如何变化。考虑两个有序基：

$$B = (b_1, \dots, b_n), \tilde{B} = (\tilde{b}_1, \dots, \tilde{b}_n) \quad (2.98)$$

作为 V 的基，以及两个有序基：

$$C = (c_1, \dots, c_m), \tilde{C} = (\tilde{c}_1, \dots, \tilde{c}_m) \quad (2.99)$$

作为 W 的基。此外， $A_\Phi \in \mathbb{R}^m \times n$ 是线性映射 $\Phi : V \rightarrow W$ 相对于基 B 和 C 的变换矩阵， $\tilde{A}_\Phi \in \mathbb{R}^m \times n$ 是相对于基 $\tilde{B}$ 和 $\tilde{C}$ 的对应变换映射。

在下面的内容中，我们将研究 $\tilde{A}$ 和 A 如何相关，即当我们选择从 $\tilde{B}, \tilde{C}$ 到 B, C 进行基变换时，我们如何/是否可以将 $\tilde{A}_\Phi$ 转换为 A_Φ 。

备注. 我们有效地得到了恒等映射 id_V 的不同坐标表示。在图 2.9 的上下文中，这意味着将相对于 (e_1, e_2) 的坐标映射到相对于 (b_1, b_2) 的坐标，而不改变向量 x 。通过改变基并相应地改变向量的表示，相对于这个新基的变换矩阵可以具有特别简单的形式，从而允许直接的计算。◇

例 2.23 (基变换)

考虑变换矩阵：

$$A = [2 \ 1; 1 \ 2] \quad (2.100)$$

相对于 $\mathbb{R}^2$ 中的标准基。如果我们定义新基：

$$B = ([1; 1], [1; -1]) \quad (2.101)$$

我们得到对角变换矩阵：

$$\tilde{A} = [3 \ 0; 0 \ 1] \quad (2.102)$$

相对于基 B ，这比 A 更容易处理。

在下面的内容中，我们将研究将相对于一个基的坐标向量转换为相对于不同基的坐标向量的映射。我们将首先陈述主要结果，然后提供解释。

定理 2.20 (基变换). 对于线性映射 $\Phi : V \rightarrow W$ ，有序基

$$B = (b_1, \dots, b_n), \tilde{B} = (\tilde{b}_1, \dots, \tilde{b}_n) \quad (2.103)$$

属于 V ，以及

$$C = (c_1, \dots, c_m), \tilde{C} = (\tilde{c}_1, \dots, \tilde{c}_m) \quad (2.104)$$

属于 W ，以及 Φ 相对于 B 和 C 的变换矩阵 A_Φ ，则相对于基 $\tilde{B}$ 和 $\tilde{C}$ 的对应变换矩阵 $\tilde{A}_\Phi$ 由下式给出：

$$\tilde{A}_\Phi = T^{-1} A_\Phi S \quad (2.105)$$

这里， $S \in \mathbb{R}^n \times \mathbb{R}^n$ 是 id_V 的变换矩阵，将相对于 $\tilde{B}$ 的坐标映射到相对于 B 的坐标， $T \in \mathbb{R}^m \times \mathbb{R}^m$ 是 id_W 的变换矩阵，将相对于 $\tilde{C}$ 的坐标映射到相对于 C 的坐标。

证明 根据 Drumm and Weil (2001)，我们可以将 V 的新基 $\tilde{B}$ 的向量写成基 B 的基向量的线性组合，使得：

$$\tilde{b}_j = s_{1j} b_1 + \dots + s_{nj} b_n = \sum_{i=1}^n s_{ij} b_i, j = 1, \dots, n \quad (2.106)$$

类似地，我们将 W 的新基向量 $\tilde{C}$ 写成基 C 的基向量的线性组合，得到：

$$\tilde{c}_k = t_{1k} c_1 + \dots + t_{mk} c_m = \sum_{l=1}^m t_{lk} c_l, k = 1, \dots, m \quad (2.107)$$

我们定义 $S = ((s_{ij})) \in \mathbb{R}^n \times \mathbb{R}^n$ 为将相对于 $\tilde{B}$ 的坐标映射到相对于 B 的坐标的变换矩阵， $T = ((t_{lk})) \in \mathbb{R}^m \times \mathbb{R}^m$ 为将相对于 $\tilde{C}$ 的坐标映射到相对于 C 的坐标的变换矩阵。特别地， S 的第 j 列是 $\tilde{b}_j$ 相对于 B 的坐标表示， T 的第 k 列是 $\tilde{c}_k$ 相对于 C 的坐标表示。注意 S 和 T 都是可逆的。

我们将从两个角度来看 $\Phi(\tilde{b}_j)$ 。首先，应用映射 Φ ，对于所有 $j = 1, \dots, n$ ，我们得到：

$$\Phi(\tilde{b}_j) = \sum_{k=1}^m \tilde{a}_{kj} \tilde{c}_k = \sum_{k=1}^m \tilde{a}_{kj} \sum_{l=1}^m t_{lk} c_l = \sum_{l=1}^m \left(\sum_{k=1}^m t_{lk} \tilde{a}_{kj} \right) c_l \quad (2.108)$$

其中我们首先将新的基向量 $\tilde{c}[k] \in W$ 表示为基向量 $c[l] \in W$ 的线性组合，然后交换求和顺序。

或者，当我们将 $\tilde{b}[j] \in V$ 表示为 $b[j] \in V$ 的线性组合时，我们得到

$$[n]! [n] [n] [m]$$

$$\Phi(\tilde{b}[j]) = \sum_{k=1}^m \tilde{a}_{kj} \tilde{c}_k = \sum_{k=1}^m \tilde{a}_{kj} \sum_{l=1}^m t_{lk} c_l = \sum_{l=1}^m \left(\sum_{k=1}^m t_{lk} \tilde{a}_{kj} \right) c_l \quad (2.109a) \quad [i] [ij] [i] [ij] [i] [ij] [i]$$

$$[i][=1] [i][=1] [i][=1] [l][=1]$$

$$[m]! [n]$$

$$= \sum_{l=1}^m \sum_{k=1}^m t_{lk} \tilde{a}_{kj} c_l, j = 1, \dots, n, [(2.109b)]$$

$$[i] [ij] [i]$$

$$[l][=1] [i][=1]$$

其中我们利用了 Φ 的线性性。比较 [(2.108)] 和 [(2.109b)], 对于所有 $j = 1, \dots, n$ 和 $l = 1, \dots, m$ ，可得

$$[m] [n]$$

$$\sum_{k=1}^m \tilde{a}_{kj} t_{lk} = \sum_{k=1}^m \tilde{a}_{kj} t_{lk} \quad (2.110) \quad [lk] [kj] [li] [ij]$$

$$[k][=1] [i][=1]$$

因此,

$$T^{-1}[m][\times][n] A[\Phi] = A[\Phi] S \in R, (2.111)$$

使得

$$A^{-1}[\Phi] = TA[\Phi]S, (2.112)$$

这证明了定理 [2.20.]

定理 [2.20] 告诉我们, 当 V 中的基改变 (B 被替换为 B^{-1}) 和 W 中的基改变 (C 被替换为 C^{-1}) 时, 线性映射 $\Phi: V \rightarrow W$ 的变换矩阵 $A[\Phi]$ 被等价矩阵 $A^{-1}[\Phi]$ 替换, 其中

$$A^{-1}[\Phi] = TA[\Phi]S. (2.113)$$

图 [2.11] 说明了这种关系: 考虑同态 $\Phi: V \rightarrow W$ 以及 V 的有序基 B, B^{-1} 和 W 的有序基 C, C^{-1} 。映射 $\Phi[CB]$ 是 Φ 的一个实例, 将 B 的基向量映射到 C 的基向量的线性组合上。假设我们知道关于有序基 B, C 的 $\Phi[CB]$ 的变换矩阵 $A[\Phi]$ 。当我们在 V 中执行从 B 到 B^{-1} 的基变换, 在 W 中执行从 C 到 C^{-1} 的基变换时, 我们可以确定

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由 Cambridge University Press 出版 (2020).]

[56] [*Linear Algebra*]]

图 2.11 对于同态 $\Phi: V \rightarrow W$ 和有序基 B, B^{-1} 的 V 以及 C, C^{-1} 的 W (标记为蓝色), 我们可以等价地将关于基 B^{-1}, C^{-1} 的映射 $\Phi[C^{-1}][B^{-1}][\cdot]$ 表示为同态的组合, 下标中的基对应的变换矩阵用红色标出。

相应的变换矩阵 $A^{-1}[\Phi]$ 如下: 首先, 我们找到线性映射 $\Psi[B][B^{-1}][\cdot]: V \rightarrow V$ 的矩阵表示, 该映射将关于新基 B^{-1} 的坐标映射到关于“旧”基 B (在 V 中) 的 (唯一) 坐标。然后, 我们使用 $\Phi[CB]: V \rightarrow W$ 的变换矩阵 $A[\Phi]$ 将这些坐标映射到关于 W 中 C 的坐标。最后, 我们使用线性映射 $\Xi[C^{-1}][C][\cdot]: W \rightarrow W$ 将关于 C 的坐标映射到关于 C^{-1} 的坐标。因此, 我们可以将线性映射 $\Phi[C^{-1}][B^{-1}][\cdot]$ 表示为涉及“旧”基的线性映射的组合:

$$\Phi^{-1}[C^{-1}][B^{-1}][\cdot][B][C][\cdot][B][C^{-1}][\cdot][B][B^{-1}][\cdot][C][C^{-1}][\cdot] = \Xi \circ \Phi \circ \Psi = \Xi \Phi \Psi [C] [(2.114) \circ \circ [B][\cdot] \cdot [B]]$$

具体地, 我们使用 $\Psi[\cdot]$ 和 $[B][B] = \text{id}[V]$ $\Xi[\cdot]$, 即恒等映射, 将向量映射到自身, 但使用不同的基。 $[C][C] = \text{id}[W]$

定义 2.21 (等价)。两个矩阵 $A, A^{-1} \in R[m][\times][n]$ 是等价的, 如果存在正则矩阵 $S \in R[n][\times][n]$ 和 $T \in R[m][\times][m]$, 使得

$$A^{-1} = TAS.$$

定义 2.22 (相似)。两个矩阵 $A, A^{-1} \in R[n][\times][n]$ 是相似的, 如果存在正则矩阵 $S \in R[n][\times][n]$ 使得 $A^{-1} = SAS^{-1}$

备注。相似矩阵总是等价的。然而, 等价矩阵不一定相似。◇

备注。考虑向量空间 V, W, X 。从定理 [2.17] 后面的备注, 我们已经知道对于线性映射 $\Phi: V \rightarrow W$ 和 $\Psi: W \rightarrow X$, 映射 $\Psi \circ \Phi: V \rightarrow X$ 也是线性的。有了相应映射的变换矩阵 $A[\Phi]$ 和 $A[\Psi]$, 整体变换矩阵是 $A[\Psi][\circ][\Phi] = A[\Psi]A[\Phi]$ 。

◇

根据这个备注，我们可以从组合线性映射的角度来看基变换：

$A[\Phi]$ 是关于基 B, C 的线性映射 $\Phi[CB]: V \rightarrow W$ 的变换矩阵。 $A^{-1}[\Phi]$ 是关于基 B^{-1}, C^{-1} 的线性映射 $\Phi[C^{-1}][B^{-1}]: V \rightarrow W$ 的变换矩阵。 S 是线性映射 $\Psi[B][B^{-1}]: V \rightarrow V$ （自同态）的变换矩阵，用 B 来表示 B^{-1} 。通常， $\Psi = \text{id}[V]$ 是 V 中的恒等映射。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] [<https://mml-book.com>][.] [2.7 Linear Mappings] [[57]]

T 是线性映射 $\Xi[C][C^{-1}]: W \rightarrow W$ （自同态）的变换矩阵，用 C 来表示 C^{-1} 。通常， $\Xi = \text{id}[W]$ 是 W 中的恒等映射。

如果我们（非正式地）仅用基来写下变换，

那么 $A[\Phi]: B \rightarrow C^{-1}, A^{-1}[\Phi]: B^{-1} \rightarrow C^{-1}, S: B^{-1} \rightarrow B, T: C^{-1} \rightarrow C$ 和 $T^{-1}: C \rightarrow C^{-1}$ ，以及

$$B^{-1} \rightarrow C^{-1} = B^{-1} \rightarrow B \rightarrow C \rightarrow C^{-1} \quad (2.115)$$

$$A^{-1}[\Phi] = TA[\Phi]S. \quad (2.116)$$

注意 [(2.116)] 中的执行顺序是从右到左，因为向量在右侧相乘，所以 $x \mapsto Sx \mapsto A\Phi \mapsto T^{-1}A(Sx) = A^{-1}x$ 。 $[\Phi]$

例 2.24 (基变换)

考虑线性映射 $\Phi: \mathbb{R}^3 \rightarrow \mathbb{R}^4$ ，其变换矩阵是

$$\begin{pmatrix} 1 & 2 & 0 \\ -1 & 1 & 3 \end{pmatrix}$$

$$A[\Phi] = \begin{pmatrix} 1 & 2 & 0 \\ -1 & 1 & 3 \end{pmatrix} \quad (2.117)$$

$$\begin{pmatrix} 3 & 7 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 & 2 & 4 \end{pmatrix}$$

关于标准基

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$B = (0, 1, 0), C = (, , ,). \quad (2.118)$$

$$\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}$$

0 0 1

0 0 0 1

我们寻求 Φ 在新基下的变换矩阵 $\tilde{A}[\Phi]$:

1 0 1 1

$\tilde{1} 1 0 1$

$B = (1, 1, 0) \in \mathbb{R}^3, C = (, , ,) . (2.119) 0 3 \tilde{0} 1 0 1 1 0$

0 1 1

0 0 0 1

那么,

1 1 0 1

1 0 1

1 0 1 0

$S = 1 1 0, T = , (2.120)$

0 1 1 0

0 1 1

0 0 0 1

其中 S 的第 $\tilde{i}$ 列是 $b[i]$ 用 B 的基向量表示的坐标表示。由于 B 是标准基, 坐标表示很容易找到。对于一般的基 B , 我们需要求解线性方程组来找到 $\lambda[i]$, 使得

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[58] *[[Linear Algebra]]*

$P 3 \lambda b, j, \dots, .$ 类似地, T 的第 j 列是 $\tilde{c}[j]$ 用 C 的基向量表示的坐标表示。 $= \tilde{b} = 1 3 j T$

$[i]=[1] [i] [i] [i]$

因此, 我们得到

1 1 - 1 - 1 3 2 1

$A \tilde{-} 1 1 -1 1 -1 0 4 2 [\Phi] = T A [\Phi] S = (2.121a) 2 - 1 1 1 1 0 8 4$

0 0 0 2 1 6 3

- 4 - 4 - 2

6 0 0

= . (2.121b)

4 8 4

1 6 3

在第 4 章中，我们将能够利用基变换的概念找到一个基，在该基下自同态的变换矩阵具有特别简单的（对角）形式。在第 10 章中，我们将研究数据压缩问题，并找到一个方便的基，我们可以将数据投影到该基上，同时最小化压缩损失。

2.7.3 像和核

线性映射的像和核是具有某些重要性质的向量空间。接下来，我们将更仔细地描述它们的特征。

定义 2.23（像和核）。对于 $\Phi : V \rightarrow W$ ，我们定义核/零空间

$$[\text{零空间}] - 0 \ker(\Phi) := \Phi^{-1}(0[W]) = \{v \in V : \Phi(v) = 0[W]\} \quad (2.122)$$

[像] 和像/值域

[值域]

$$\text{Im}(\Phi) := \Phi(V) = \{w \in W \mid \exists v \in V : \Phi(v) = w\} \quad [(2.123)]$$

[定义域] 我们也称 V 和 W 分别为 Φ 的定义域和余定义域。[余定义域]

直观上，核是 Φ 映射到中性元 $0[W] \in W$ 的向量 $v \in V$ 的集合。像是可以从 V 中的任何向量通过 Φ “到达”的向量 $w \in W$ 的集合。图 2.12 给出了示意图。

注记。 考虑线性映射 $\Phi : V \rightarrow W$ ，其中 V 、 W 是向量空间。

总是有 $\Phi(0[V]) = 0[W]$ ，因此 $0[V] \in \ker(\Phi)$ 。特别地，零空间永远不为空。

$\text{Im}(\Phi) \subseteq W$ 是 W 的子空间， $\ker(\Phi) \subseteq V$ 是 V 的子空间。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>][.] [2.7 Linear Mappings] [59]

V [$\Phi : V \rightarrow W$ 图 2.12 线性映射] W [$\Phi : V \rightarrow W$ 的核和像。]

$[\ker(\Phi)]$ $[\text{Im}(\Phi)]$

$[0[V]]$ $[0[W]]$

Φ 是单射的（一对一的）当且仅当 $\ker(\Phi) = \{0\}$ 。

◇

记 (零空间和列空间)。设 $A \in \mathbb{R}^{(m \times n)}$, 考虑线性映射 $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m, x \mapsto Ax$ 。

对于 $A = [a_1, \dots, a_n]$, 其中 a_i 是 A 的列, 我们得到

(n)

$$\text{Im}(\Phi) = \{ Ax : x \in \mathbb{R}^n \} = \left\{ \sum_{i=1}^n x[i] a[i] : x[1], \dots, x[n] \in \mathbb{R} \right\} \quad (2.124a)$$

$[i=1]$

$$= \text{span}[a_1, \dots, a_n] \subseteq \mathbb{R}^m, \quad (2.124b)$$

即, 像是 A 的列的生成空间, 也称为 *列空间*。因此, 列空间 (像) 是 $\mathbb{R}^m$ 的一个子空间, 其中 m 是矩阵的 “高度”。

$$\text{rk}(A) = \dim(\text{Im}(\Phi)).$$

核/零空间 $\ker(\Phi)$ 是齐次线性方程组 $Ax = 0$ 的通解, 它捕获了 $\mathbb{R}^n$ 中产生 $0 \in \mathbb{R}^m$ 的元素的所有可能线性组合。

核是 $\mathbb{R}^n$ 的一个子空间, 其中 n 是矩阵的 “宽度”。

核关注列之间的关系, 我们可以用它来确定是否/如何将一列表示为其他列的线性组合。

◇

例 2.25 (线性映射的像和核)

映射

$$x[1] \quad x[1]$$

$$x[2] \quad x[2]$$

$$\Phi : \mathbb{R}^4 \rightarrow \mathbb{R}^2, [x] \mapsto [x_1 + 2x_2 - x_3] = [1 \ 2 \ -1 \ 0] [x] \quad (2.125a)$$

$$x[4] \quad x_1 + x[4] \quad [1 \ 0 \ 0 \ 1] [x[4]]$$

$$1 \ 2 \ -1 \ 0$$

$$= x_1 + x[2] + x_3 + x[4] \quad (2.125b)$$

$$1 \ 0 \ 0 \ 1$$

是线性的。为了确定 $\text{Im}(\Phi)$, 我们可以取变换矩阵的列的生成空间, 得到

$$1 \ 2 \ -1 \ 0$$

$$\text{Im}(\Phi) = \text{span}\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \}. \quad (2.126)$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}$$

为了计算 Φ 的核（零空间），我们需要求解 $Ax = 0$ ，即我们需要求解齐次方程组。为此，我们使用 Gaussian 消元将 A 变换为简化行阶梯形式：

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

$$\rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \dots \quad (2.127)$$

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

这个矩阵是简化行阶梯形式，我们可以使用 Minus-1 技巧来计算核的基（见第 2.3.3 节）。或者，我们可以将非主元列（第 3 列和第 4 列）表示为主元列（第 1 列和第 2 列）的线性组合。第三列 a_3 等于第二列 a_2 的 $-1/2$ 倍。因此， $0 = a_3 + (1/2)a_2$ 。同样地，我们看到 $a_4 = a_1 - a_2$ ，因此 $0 = a_1 - a_2 - a_4$ 。

总的来说，这给出了核（零空间）：

$$\begin{bmatrix} 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1/2 & 1 \end{bmatrix}$$

$$\ker(\Phi) = \text{span}\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \}. \quad (2.128)$$

$$\begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 \end{bmatrix}$$

定理 2.24（秩-零化度定理）。对于向量空间 V 、 W 和线性映射 $\Phi : V \rightarrow W$ ，有

$$\dim(\ker(\Phi)) + \dim(\text{Im}(\Phi)) = \dim(V). \quad (2.129)$$

秩-零化度定理也被称为 *基本定理*

线性映射定理

线性映射定理 (Axler, 2015, 定理3.22)。以下是定理2.24的直接推论：

如果 $\dim(\text{Im}(\Phi)) < \dim(V)$ ，那么 $\ker(\Phi)$ 是非平凡的，即 $\ker(\Phi)$ 包含的不仅仅是 0_V ，并且 $\dim(\ker(\Phi)) \geq 1$ 。如果 A_Φ 是 Φ 相对于有序基的变换矩阵，且 $\dim(\text{Im}(\Phi)) < \dim(V)$ ，那么线性方程组 $A_\Phi \mathbf{x} = 0$ 有无穷多个解。如果 $\dim(V) = \dim(W)$ ，那么以下三向等价关系成立： - Φ 是单射的 - Φ 是满射的 - Φ 是双射的

因为 $\text{Im}(\Phi) \subseteq W$ 。

2.8 仿射空间

接下来，我们将仔细研究从原点偏移的空间，即不再是向量空间的空间。此外，我们将简要讨论这些仿射空间之间映射的性质，这些映射类似于线性映射。

备注。在机器学习文献中，线性和仿射之间的区别有时不够清晰，因此我们可以找到将仿射空间/映射称为线性空间/映射的参考文献。◇

2.8.1 仿射子空间

定义2.25（仿射子空间）。设 V 是向量空间， $x_0 \in V$ ， $U \subseteq V$ 是子空间。那么子集

$$L = x_0 + U := \{x_0 + u : u \in U\} \quad (2.130a) = \{v \in V \mid \exists u \in U : v = x_0 + u\} \subseteq V \quad (2.130b)$$

称为 V 的 *仿射子空间* 或 *线性流形*。 U 称为 *方向* 或 *方向空间*， x_0 称为 *支撑点*。在第12章中，我们将这样的子空间称为 *超平面*。

注意，仿射子空间的定义在 $x_0 \notin U$ 时排除了 0 。因此，对于 $x_0 \notin U$ ，仿射子空间不是 V 的（线性）子空间（向量子空间）。

仿射子空间的例子是 $\mathbb{R}^3$ 中的点、直线和平面，它们不（必须）通过原点。

备注。考虑向量空间 V 的两个仿射子空间 $L = x_0 + U$ 和 $\tilde{L} = \tilde{x}_0 + \tilde{U}$ 。那么， $L \subseteq \tilde{L}$ 当且仅当 $U \subseteq \tilde{U}$ 且 $x_0 - \tilde{x}_0 \in \tilde{U}$ 。

仿射子空间通常用参数来描述：考虑 V 的 k 维仿射空间 $L = x_0 + U$ 。如果 $(b_1, \dots, b_k)$ 是 U 的有序基，那么 L 中的每个元素 x 都可以唯一地表示为

$$x = x_0 + \lambda_1 b_1 + \dots + \lambda_k b_k, \quad (2.131)$$

其中 $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ 。这种表示称为 L 的 *参数方程*，具有方向向量 $b_1, \dots, b_k$ 和参数 $\lambda_1, \dots, \lambda_k$ 。◇

例2.26（仿射子空间）

一维仿射子空间称为 *直线*，可以写成 $y = x_0 + \lambda_1 b_1$ ，其中 $\lambda \in \mathbb{R}$ ， $U = \text{span}[b_1] \subseteq \mathbb{R}^n$ 是 $\mathbb{R}^n$ 的一维子空间。这意味着直线由支撑点 x_0 和定义方向的向量 b_1 确定。参见图2.13的说明。

$\mathbb{R}^n$ 的二维仿射子空间称为 *平面*。平面的参数方程是 $y = x_0 + \lambda_1 b_1 + \lambda_2 b_2$ ，其中 $\lambda_1, \lambda_2 \in \mathbb{R}$ ， $U = \text{span}[b_1, b_2] \subseteq \mathbb{R}^n$ 。这意味着平面由支撑点 x_0 和两个线性无关的向量 b_1, b_2 确定，它们张成方向空间。

在 $\mathbb{R}^n$ 中， $(n-1)$ 维仿射子空间称为 *超平面*，相应的参数方程是 $y = x_0 + \sum_{i=1}^{n-1} \lambda_i b_i$ ，其中 $b_1, \dots, b_{n-1}$ 构成 $\mathbb{R}^n$ 的 $(n-1)$ 维子空间 U 的基。这意味着超平面由支撑点 x_0 和 $(n-1)$ 个线性无关的向量 $b_1, \dots, b_{n-1}$ 确定，它们张成方向空间。在 $\mathbb{R}^2$ 中，直线也是超平面。在 $\mathbb{R}^3$ 中，平面也是超平面。

图2.13

直线是仿射子空间。直线上的向量 $y = x_0 + \lambda b_1$ 位于具有支撑点 x_0 和方向 b_1 的仿射子空间 L 中。

备注（非齐次线性方程组和仿射子空间）。对于 $A \in \mathbb{R}^{(m \times n)}$ 和 $x \in \mathbb{R}^m$ ，线性方程组 $A\lambda = x$ 的解要么是空集，要么是 $\mathbb{R}^n$ 的维数为 $n - \text{rk}(A)$ 的仿射子空间。特别地，线性方程 $\lambda_1 b_1 + \dots + \lambda_n b_n = x$ 的解，其中 $(\lambda_1, \dots, \lambda_n) \neq (0, \dots, 0)$ ，是 $\mathbb{R}^n$ 中的超平面。

在 $\mathbb{R}^n$ 中，每个 k 维仿射子空间都是非齐次线性方程组 $Ax = b$ 的解，其中 $A \in \mathbb{R}^{(m \times n)}$ ， $b \in \mathbb{R}^m$ ，且 $\text{rk}(A) = n - k$ 。回忆一下，对于齐次方程组 $Ax = 0$ ，解是向量子空间，我们也可以将其视为支撑点 $x_0 = 0$ 的特殊仿射空间。◇

2.8.2 仿射映射

类似于向量空间之间的线性映射（我们在第2.7节中讨论过），我们可以定义两个仿射空间之间的仿射映射。线性映射和仿射映射密切相关。因此，我们已经从线性映射中了解的许多性质，例如线性映射的复合是线性映射，也适用于仿射映射。

定义2.26（仿射映射）。对于两个向量空间 V, W ，线性映射 $\Phi : V \rightarrow W$ ，以及 $a \in W$ ，映射

$$\phi : V \rightarrow W \quad (2.132) \quad x \mapsto a + \Phi(x) \quad (2.133)$$

是从 V 到 W 的仿射映射。向量 a 称为 ϕ 的平移向量。

每个仿射映射 $\phi : V \rightarrow W$ 也是线性映射 $\Phi : V \rightarrow W$ 和 W 中的平移 $\tau : W \rightarrow W$ 的组合，使得

$$\phi = \tau \circ \Phi. \text{ 映射 } \Phi \text{ 和 } \tau \text{ 是唯一确定的。}$$

仿射映射 $\phi : V \rightarrow W$ ， $\phi' : W \rightarrow X$ 的组合 $\phi' \circ \phi$ 是仿射的。

仿射映射保持几何结构不变。它们也保持维度和平行性。

2.9 延伸阅读

学习线性代数有很多资源，包括 [Strang (2003),] [Golan (2007),] [Axler (2015),] 和 [Liesen and Mehrmann (2015)] 的教科书。还有几个在线资源，我们在本章的介绍中提到过。我们这里只涵盖了高斯消元法，但有许多其他方法可以求解线性方程组，我们推荐 [Stoer and Burlirsch (2002),] [Golub and Van Loan (2012),] 和 [Horn and Johnson (2013)] 的数值线性代数教科书进行深入讨论。

在本书中，我们区分线性代数的主题（如向量、矩阵、线性无关、基）和与向量空间几何相关的主题。在第3章中，我们将介绍内积，它诱发一个范数。这些概念允许我们定义角度、长度和距离，我们将用于正交投影。投影在许多机器学习算法中起关键作用，如线性回归和主成分分析，我们将分别在第9章和第[10]章中涵盖。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[64] [[Linear Algebra]]

练习

[2.1] [我们考虑] $(\mathbb{Q}[\mathbb{R}\{-1\}], \cdot)$ [，其中]

$$[a \cdot b] := [ab] + [a] + [b], [a, b] \in \mathbb{R}\{-1\} \quad [(2.134)]$$

[a. 证明] $(\mathbb{Q}[\mathbb{R}\{-1\}], \cdot)$ [是一个阿贝尔群。] [b. 求解]

$$3[\cdot x \cdot x] = 15$$

[在阿贝尔群] $(\mathbb{Q}[\mathbb{R}\{-1\}], \cdot)$ [中，其中] $[\cdot]$ [定义在 (2.134) 中。]

[2.2] [设] $[n]$ [属于] $\mathbb{N} \setminus \{0\}$ [。设] $[k, x]$ [属于] $\mathbb{Z}$ [。我们定义整数] $[k]$ [的同余类] $[k]$ [为集合]

$$[k] := \{[x] \in \mathbb{Z} \mid x \equiv k \pmod{n}\} = \{[x] \in \mathbb{Z} \mid \exists [a] \in \mathbb{Z} : ([x] - k) = [n] \cdot [a])\}$$

[我们现在定义] $\mathbb{Z}/n\mathbb{Z}$ [(有时写作] $\mathbb{Z}[n]$ [为所有模] $[n]$ [的同余类的集合。欧几里得除法意味着这个集合是包含] $[n]$ [个元素的有限集合：]

$$[\mathbb{Z}[n]] = \{[0], [1], \dots, [n-1]\}$$

[对于所有] $[a, b] \in \mathbb{Z}[n]$ [，我们定义] $[a] \oplus [b] := [a] + [b]$

[a. 证明] $(\mathbb{Z}[n], \oplus)$ [是一个群。它是阿贝尔的吗？] [b. 我们现在为] $\mathbb{Z}[n]$ [中的所有] $[a]$ [和] $[b]$ [定义另一个运算] $[\otimes]$ [：]

$$[a] \otimes [b] := [a] \times [b] \quad [(2.135)]$$

[其中] $[a] \times [b]$ [表示] $\mathbb{Z}$ [中的通常乘法。] [设] $[n] = 5$ [。绘制] $\mathbb{Z}[5]$ [在] $[\otimes]$ [下的乘法表，即计算] $\mathbb{Z}[5]$ [中所有] $[a]$ [和] $[b]$ [的乘积] $[a] \otimes [b]$ [。] [因此，证明] $\mathbb{Z}[5]$ [在] $[\otimes]$ [下封闭并且对于] $[\otimes]$ [拥有中性元素。显示] $\mathbb{Z}[5]$ [中所有元素在] $[\otimes]$ [下的逆。] [得出结论] $(\mathbb{Z}[5], \otimes)$ [是一个阿贝尔群。]

[c. 证明] $(\mathbb{Z}[8], \otimes)$ [不是一个群。] [d. 我们回顾贝祖定理(Bézout theorem)陈述两个整数] $[a]$ [和] $[b]$ [互质（即] $\gcd([a], [b]) = 1$ [）当且仅当存在两个整数] $[u]$ [和] $[v]$ [使得] $[au] + [bv] = 1$ [。证明] $(\mathbb{Z}[n], \otimes)$ [是一个群当且仅当] $[n] \in \mathbb{N} \setminus \{0\}$ [是质数。]

[2.3] [考虑如下定义的] 3×3 [矩阵集合] $[G]$ [：]

$$G = \{[1 \ x \ z], [0 \ 1 \ y], [0 \ 0 \ 1] \mid x, y, z \in \mathbb{R}\}$$

[我们定义] $[\cdot]$ [为标准矩阵乘法。] $(G, \cdot)$ [是一个群吗？如果是，它是阿贝尔的吗？证明你的答案。]

[2.4] [如果可能，计算以下矩阵乘积：]

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] <https://mml-book.com>.] [Exercises] [[65]]

$$[a.] [[1 \ 2], [7 \ 8]] [[1 \ 0], [4 \ 5 0], [1 \ 1 \ 1]]$$

$$[b.] [[1 \ 2 \ 3], [4 \ 5 \ 6], [7 \ 8 \ 9]] [[1 \ 1 \ 0], [0 \ 1 \ 1], [1 \ 0 \ 1]]$$

$$[c.] [[1 \ 1 \ 0], [0 \ 1 \ 1], [1 \ 0 \ 1]] [[1 \ 2 \ 3], [4 \ 5 \ 6], [7 \ 8 \ 9]]$$

$$[d.] \begin{bmatrix} 1 & 2 & 1 \\ 4 & 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 3 \\ 1 & -1 \\ -4 & 2 \end{bmatrix}$$

$$[e.] \begin{bmatrix} 0 & 3 \\ 1 & -1 \\ -4 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 4 & 1 & -1 \end{bmatrix}$$

[2.5] [求下列非齐次线性系统] $[Ax] = [b]$ [中] x [的所有解的集合] $[S]$ [, 其中] $[A]$ [和] $[b]$ [定义如下:]

$$[a.] A = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 2 & 5 & -7 & -5 \\ 2 & -1 & 1 & 3 \\ 5 & 2 & -4 & 2 \end{bmatrix}, b = [1, [-2], [4], 6]$$

$$[b.] A = \begin{bmatrix} 1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -3 & 0 \\ 2 & -1 & 0 & 1 & -1 \\ -1 & 2 & 0 & -2 & -1 \end{bmatrix}, b = [3, 6, [5], [-1]]$$

[2.6] [使用高斯消元法, 求非齐次方程组] $[Ax] = [b]$ [的所有解, 其中]

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, b = [[0], [2], [-1], 1]$$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[66] [*Linear Algebra*]

x_1

2.7 求解方程组在 $x \in \mathbb{R}$ 中的所有解:

$$2\Lambda x = 12x$$

其中

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

且 $\sum_{i=1 \text{ to } 3} P_{-i} x_{-i} = 1$ 。

2.8 如果可能, 求以下矩阵的逆矩阵:

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

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

2.9 以下哪些集合是 $\mathbb{R}^3$ 的子空间?

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

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

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

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

2.10 以下向量集合是否线性无关?

$$a. x_1 = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} x_2 = \begin{bmatrix} -1 \\ 1 \\ -3 \end{bmatrix} x_3 = \begin{bmatrix} 3 \\ -2 \\ 8 \end{bmatrix}$$

$$b. x_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} x_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x_3 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

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

写成以下向量的线性组合：

$$x_1 = 1 \quad x_2 = 1 \quad x_3 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

草稿 (2023-10-18) 《机器学习数学》。反馈：<https://mml-book.com>。练习 [67]

2.12 考虑 $\mathbb{R}^4$ 的两个子空间：

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

求 $U_1 \cap U_2$ 的基。

2.13 考虑两个子空间 U_1 和 U_2 ，其中 U_1 是齐次方程组 $A_1 x = 0$ 的解空间， U_2 是齐次方程组 $A_2 x = 0$ 的解空间，其中：

$$A_1 = \begin{bmatrix} 1 & 0 & 1 & 3 & -3 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 1 & -2 & -1 & 1 & 2 & 3 \\ 2 & 1 & 3 & 7 & -5 & 2 \\ 1 & 0 & 1 & 3 & -1 & 2 \end{bmatrix}$$

a. 求 U_1, U_2 的维数。

b. 求 U_1 和 U_2 的基。

c. 求 $U_1 \cap U_2$ 的基。

2.14 考虑两个子空间 U_1 和 U_2 ，其中 U_1 由 A_1 的列张成， U_2 由 A_2 的列张成，其中：

$$A_1 = \begin{bmatrix} 1 & 0 & 1 & 3 & -3 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 1 & -2 & -1 & 1 & 2 & 3 \\ 2 & 1 & 3 & 7 & -5 & 2 \\ 1 & 0 & 1 & 3 & -1 & 2 \end{bmatrix}$$

a. 求 U_1, U_2 的维数

b. 求 U_1 和 U_2 的基

c. 求 $U_1 \cap U_2$ 的基

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

a. 证明 F 和 G 是 $\mathbb{R}^3$ 的子空间。

b. 不使用任何基向量计算 $F \cap G$ 。

c. 分别为 F 和 G 找一组基，使用之前找到的基向量计算 $F \cap G$ ，并与前一题的结果进行核对。

2.16 以下映射是否为线性映射？

a. 设 $a, b \in \mathbb{R}$ 。

$$\Phi : L^1([a, b]) \rightarrow \mathbb{R} \quad f \mapsto \Phi(f) = \int_a^b f(x) dx$$

其中 $L^1([a, b])$ 表示在 $[a, b]$ 上可积函数的集合。

b. $\Phi : C^1 \rightarrow C^0 \quad f \mapsto \Phi(f) = f'$

其中对于 $k \geq 1$, C^k 表示 k 次连续可微函数的集合, C^0 表示连续函数的集合。

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由 Cambridge University Press 出版 (2020)。

68 [线性代数]

c. $\Phi: \mathbb{R} \rightarrow \mathbb{R} \times \mathbb{R} \mapsto \Phi(x) = (\cos(x), \sin(x))$

d. $\Phi: \mathbb{R}^3 \rightarrow \mathbb{R}^2 \times \mathbb{R} \mapsto [1 \ 2 \ 3] \times [1 \ 4 \ 3]$

e. 设 $\theta \in [0, 2\pi]$ 且

$$\Phi: \mathbb{R}^2 \rightarrow \mathbb{R}^2 \times \mathbb{R} \mapsto [\cos(\theta) \ \sin(\theta)] \times [-\sin(\theta) \ \cos(\theta)]$$

2.17 考虑线性映射

$$\Phi: \mathbb{R}^3 \rightarrow \mathbb{R}^4$$

$$\Phi([x_1]) = [3x_1 + 2x_2 + x_3] [x_2] [x_1 + x_2 + x_3] [x_3] [x_1 + 3x_2 - x_3] [2x_1 + 3x_2 + x_3]$$

求变换矩阵 A_Φ 。求 $\text{rk}(A_\Phi)$ 。计算 Φ 的核(kernel)和像(image)。 $\dim(\ker(\Phi))$ 和 $\dim(\text{Im}(\Phi))$ 是什么?

2.18 设 E 是向量空间。设 f 和 g 是 E 上的两个自同构(automorphism), 使得 $f \circ g = \text{id}_E$ (即, $f \circ g$ 是恒等映射 id_E)。证明 $\ker(f) = \ker(g \circ f)$, $\text{Im}(g) = \text{Im}(g \circ f)$ 且 $\ker(f) \cap \text{Im}(g) = \{0_E\}$ 。

2.19 考虑自同态 $\Phi: \mathbb{R}^3 \rightarrow \mathbb{R}^3$, 其变换矩阵 (相对于 $\mathbb{R}^3$ 中的标准基) 是

$$A_\Phi = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

a. 求 $\ker(\Phi)$ 和 $\text{Im}(\Phi)$ 。

b. 求相对于基

$$B = (1, 1, 1) \ 1 \ 2 \ 0$$

的变换矩阵 $\tilde{A}_\Phi$ 。

即执行向基变换向新基 B 。

2.20 让我们考虑 $\mathbb{R}^2$ 的向量 b_1', b_2', b_1, b_2 , 在 $\mathbb{R}^2$ 的标准基中表示为:

$$b_1' = [2, 1]^T, b_2' = [-1, 2]^T, b_1 = [2, -1]^T, b_2 = [1, 1]^T$$

并且让我们定义两个有序基 $B' = (b_1', b_2')$ 和 $B = (b_1, b_2)$, 它们都是 $\mathbb{R}^2$ 的基。

a. 证明 B' 和 B 都是 $\mathbb{R}^2$ 的基, 并画出这些基向量。

b. 计算矩阵 $P_1 B'$, 该矩阵执行从 B' 到 B 的基变换。

c. 我们考虑 $\mathbb{R}^3$ 中的三个向量 c_1, c_2, c_3 , 在 $\mathbb{R}^3$ 的标准基中定义为:

$$c_1 = [1, 2, -1]^T, c_2 = [0, -1, 2]^T, c_3 = [1, 0, -1]^T$$

并且我们定义 $C = (c_1, c_2, c_3)$ 。

- i. 证明 C 是 $\mathbb{R}^3$ 的基，例如通过使用行列式（见第4.1节）。
- ii. 让我们称 $C' = (c_1', c_2', c_3')$ 为 $\mathbb{R}^3$ 的标准基。确定矩阵 $P_2 C'$ ，该矩阵执行从 C' 到 C 的基变换。
- d. 我们考虑一个同态 $\Phi: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ ，使得

$$\Phi(b_1 + b_2) = c_2 + c_3 \quad \Phi(b_1 - b_2) = 2c_1 - c_2 + 3c_3$$

其中 $B = (b_1, b_2)$ 和 $C = (c_1, c_2, c_3)$ 分别是 $\mathbb{R}^2$ 和 $\mathbb{R}^3$ 的有序基。

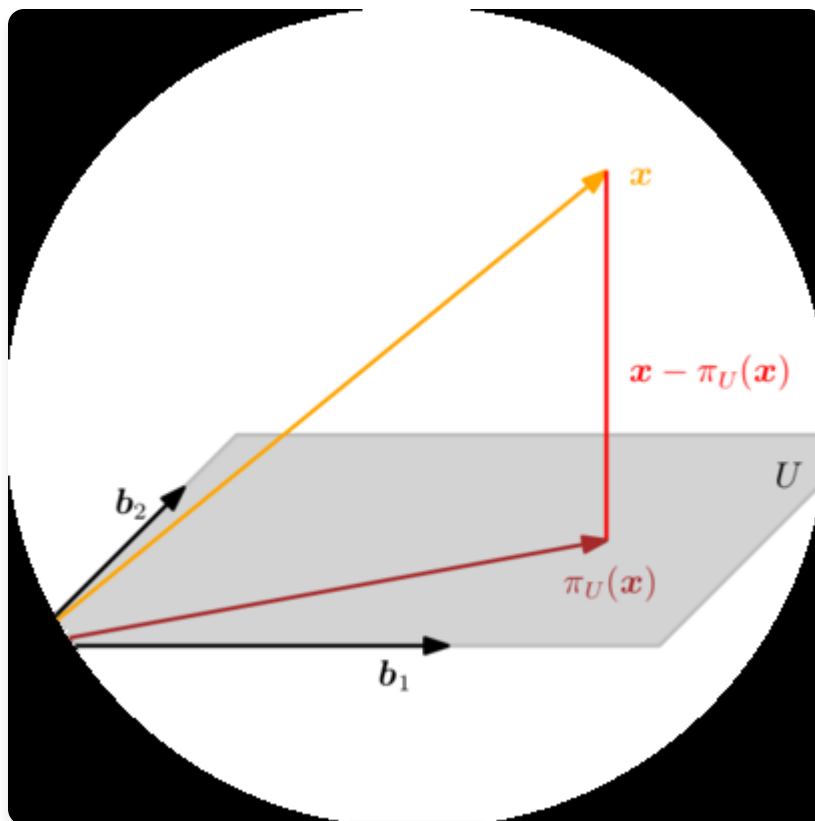
确定 Φ 关于有序基 B 和 C 的变换矩阵 A_Φ 。

- e. 确定 A' ，即 Φ 关于基 B' 和 C' 的变换矩阵。
- f. 让我们考虑向量 $x \in \mathbb{R}^2$ ，其在 B' 中的坐标为 $[2, 3]^T$ 。

换句话说， $x = 2b_1' + 3b_2'$ 。

- i. 计算 x 在 B 中的坐标。
- ii. 基于此，计算 $\Phi(x)$ 在 C 中表示的坐标。
- iii. 然后，用 c_1', c_2', c_3' 表示 $\Phi(x)$ 。
- iv. 使用 x 在 B' 中的表示和矩阵 A' 直接找到这个结果。

解析几何



在第2章中，我们在一般但抽象的层面研究了向量、向量空间和线性映射。在本章中，我们将为所有这些概念添加一些几何解释和直觉。特别地，我们将研究几何向量并计算它们的长度和距离，或两个向量之间的角度。为了能够做到这一点，我们为向量空间配备一个内积(inner product)，该内积诱导向量空间的几何结构。内积及其对应的范数(norms)和度量(metrics)捕获了相似性和距离的直观概念，我们在第12章中使用这些概念来开发支持向量机(support vector machine)。然后，我们将使用向量长度和向量间角度的概念来讨论正交投影(orthogonal projections)，这在我们讨论第10章的主成分分析(principal component analysis)和第9章通过最大似然估计的回归时将起核心作用。图3.1概述了本章概念之间的关系以及它们与本书其他章节的联系。

图3.1 本章介绍的概念的思维导图，以及它们在本书其他部分中的使用时机。

内积(Inner product)

诱导

范数(Norm) → 分类(Classification) → 第12章

长度(Lengths) $\rightarrow$ 正交投影(Orthogonal projection) $\rightarrow$ 角度(Angles) $\rightarrow$ 旋转(Rotations)

第9章 $\rightarrow$ 第4章 $\rightarrow$ 第10章 回归(Regression) $\rightarrow$ 矩阵分解(Matrix decomposition) $\rightarrow$ 降维(Dimensionality reduction)

3.1 范数

当我们考虑几何向量，即从原点开始的有向线段时，直觉上向量的长度就是这个有向线段的”端点”到原点的距离。在下面的内容中，我们将使用范数(norm)的概念来讨论向量长度的概念。

定义3.1 (范数)。向量空间 V 上的范数是一个函数

$$\|\cdot\| : V \rightarrow \mathbb{R}, (3.1) \quad x \mapsto \|x\|, (3.2)$$

它为每个向量 x 分配其长度 $\|x\| \in \mathbb{R}$ ，使得对于所有 $\lambda \in \mathbb{R}$ 和 $x, y \in V$ ，以下条件成立：

绝对齐次性: $\|\lambda x\| = |\lambda| \|x\|$

三角不等式: $\|x + y\| \leq \|x\| + \|y\|$

正定性: $\|x\| \geq 0$ 且 $\|x\| = 0 \Leftrightarrow x = 0$

在几何术语中，三角不等式表明对于任何三角形，任意两边长度之和必须大于或等于

图3.2 三角不等式。

图3.3 对于不同的范数，红线表示范数为1的向量集合。左：Manhattan范数；右：欧几里得距离。

$$\|x\|_1 = 1 \quad \|x\|_2 = 1$$

到剩余边的长度；参见图[3.2]的说明。

定义[3.1]是基于一般向量空间 V （第[2.4]节），但在本书中我们只考虑 n 维向量空间 $\mathbb{R}^n$ 。回忆对于向量 $x \in \mathbb{R}^n$ ，我们使用下标表示向量的元素，即 x_i 是向量 x 的第 i 个元素。

例3.1 (曼哈顿范数)

$\mathbb{R}^n$ 上的曼哈顿范数定义为对于 $x \in \mathbb{R}^n$ ：

$$\|x\|_1 := \sum |x_i|, (3.3)$$

其中 $|\cdot|$ 是绝对值。图[3.3]的左面板显示了所有满足 $\|x\|_1 = 1$ 的向量 $x \in \mathbb{R}^2$ 。曼哈顿范数也称为 ℓ_1 范数。

例3.2 (欧几里得范数)

$x \in \mathbb{R}^n$ 的欧几里得范数定义为：

$$\|x\|_2 := \sqrt{(\sum x_i^2)} = \sqrt{(x^T x)} (3.4)$$

并计算 x 到原点的欧几里得距离。图[3.3]的右面板显示了所有满足 $\|x\|_2 = 1$ 的向量 $x \in \mathbb{R}^2$ 。欧几里得范数也称为 ℓ_2 范数。

注意。在本书中，如果没有特别说明，我们将默认使用欧几里得范数[(3.4)]。◇

3.2 内积

内积允许引入直观的几何概念，如向量的长度以及两个向量之间的角度或距离。内积的主要目的是确定向量之间是否正交。

3.2.1 点积

我们可能已经熟悉一种特殊类型的内积，即 $\mathbb{R}^n$ 中的标量积/点积，其定义为：

$$\mathbf{x}^T \mathbf{y} = \sum x_i y_i. \quad (3.5)$$

在本书中我们将此特定内积称为点积。然而，内积是具有特定性质的更一般概念，我们现在将介绍这些概念。

3.2.2 一般内积

回忆第[2.7]节的线性映射，我们可以重新安排关于加法和标量乘法的映射。双线性映射 Ω 是具有两个参数的映射，并且在每个参数中都是线性的，即当我们考虑向量空间 V 时，对于所有 $x, y, z \in V, \lambda, \psi \in \mathbb{R}$ 成立：

$$\Omega(\lambda x + \psi y, z) = \lambda \Omega(x, z) + \psi \Omega(y, z) \quad [(3.6)] \quad \Omega(x, \lambda y + \psi z) = \lambda \Omega(x, y) + \psi \Omega(x, z). \quad [(3.7)]$$

这里，[(3.6)]断言 Ω 在第一个参数中是线性的，[(3.7)]断言 Ω 在第二个参数中是线性的（另见[(2.87)]）。

定义3.2. 设 V 是向量空间， $\Omega : V \times V \rightarrow \mathbb{R}$ 是将两个向量映射到实数的双线性映射。那么

Ω 称为*对称的*，如果对于所有 $x, y \in V$ ， $\Omega(x, y) = \Omega(y, x)$ ，即参数的顺序无关紧要。

Ω 称为*正定的*，如果

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

定义3.3. 设 V 是向量空间， $\Omega : V \times V \rightarrow \mathbb{R}$ 是将两个向量映射到实数的双线性映射。那么

正定的对称双线性映射 $\Omega : V \times V \rightarrow \mathbb{R}$ 称为 V 上的*内积*。我们通常写作 $\langle x, y \rangle$ 而不是 $\Omega(x, y)$ 。

对 $(V, \langle \cdot, \cdot \rangle)$ 称为*内积空间*或（实）*带内积的向量空间*。如果我们使用[(3.5)]中定义了点积，我们称 $(V, \langle \cdot, \cdot \rangle)$ 为欧几里得向量空间。

在本书中我们将这些空间称为内积空间。

例3.3（非点积的内积） 考虑 $V = \mathbb{R}^2$ 。如果我们定义

$$\langle x, y \rangle := x_1 y_1 - (x_1 y_2 + x_2 y_1) + 2x_2 y_2 \quad (3.9)$$

那么 $\langle \cdot, \cdot \rangle$ 是内积但不同于点积。证明将作为练习。

3.2.3 对称正定矩阵

对称正定矩阵在机器学习中起重要作用，它们通过内积定义。在第[4.3]节中，我们将在矩阵分解的背景下回到对称正定矩阵。对称正半定矩阵的概念是核函数定义的关键（第[12.4]节）。

考虑具有内积 $\langle \cdot, \cdot \rangle: V \times V \rightarrow \mathbb{R}$ （见定义[3.3]）的 n 维向量空间 V 和 V 的有序基 $B = (b_1, \dots, b_n)$ 。回忆第[2.6.1]节，任何向量 $x, y \in V$ 都可以写成基向量的线性组合，使得 $x = \sum \psi_i b_i \in V$ 和 $y = \sum \lambda_j b_j \in V$ ，其中 $\psi_i, \lambda_j \in \mathbb{R}$ 为合适的参数。由于内积的双线性性，对于所有 $x, y \in V$ 成立：

$$\langle x, y \rangle = \langle \sum \psi_i b_i, \sum \lambda_j b_j \rangle = \sum \sum \psi_i \langle b_i, b_j \rangle \lambda_j = \hat{x}^T A \hat{y}, \quad (3.10)$$

其中 $A_{ij} := \langle b_i, b_j \rangle$ ， $\hat{x}, \hat{y}$ 是 x 和 y 相对于基 B 的坐标。这意味着内积 $\langle \cdot, \cdot \rangle$ 通过 A 唯一确定。内积的对称性也意味着 A 是对称的。此外，内积的正定性意味着

$$\forall x \in V \setminus \{0\} : \hat{x}^T A \hat{x} > 0. \quad (3.11)$$

定义 3.4（对称正定矩阵）。满足[(3.11)]的对称矩阵 $A \in \mathbb{R}^{(n \times n)}$ 称为 *对称正定*，或

[正定] 只是 *正定的*。如果在 [(3.11)] 中只有 $\geq$ 成立，那么 A 被称为 *对称*，[正定] *正半定*。

[对称，正]

[半定]

例 3.4 (对称正定矩阵)

考虑矩阵

$$A_1 = \begin{pmatrix} 9 & 6 \\ 6 & 5 \end{pmatrix}, A_2 = \begin{pmatrix} 9 & 6 \\ 6 & 3 \end{pmatrix}. \quad (3.12)$$

A_1 是正定的，因为它是对称的且

$$x^T A_1 x = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{pmatrix} 9 & 6 \\ 6 & 5 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (3.13a)$$

$$= 9x_1^2 + 12x_1x_2 + 5x_2^2 = (3x_1 + 2x_2)^2 + x_2^2 > 0 \quad (3.13b)$$

对于所有 $x \in V \setminus \{0\}$ 。相比之下， A_2 是对称的但不是正定的，因为 $x^T A_2 x = 9x_1^2 + 12x_1x_2 + 3x_2^2 = (3x_1 + 2x_2)^2 - x_2^2$ ，可能小于 0，例如，对于 $x = [2, -3]^T$ 。

如果 $A \in \mathbb{R}^{(n \times n)}$ 是对称正定的，那么

$$\langle x, y \rangle = \hat{x}^T A \hat{y} \quad (3.14)$$

定义了关于有序基 B 的内积，其中 $\hat{x}$ 和 $\hat{y}$ 是 $x, y \in V$ 关于 B 的坐标表示。

定理 3.5. 对于实值有限维向量空间 V 和 V 的有序基 B ， $\langle \cdot, \cdot \rangle: V \times V \rightarrow \mathbb{R}$ 是内积当且仅当存在对称正定矩阵 $A \in \mathbb{R}^{(n \times n)}$ 使得

$$\langle x, y \rangle = \hat{x}^T A \hat{y}. \quad (3.15)$$

如果 $A \in \mathbb{R}^{(n \times n)}$ 是对称正定的, 则有以下性质:

A 的零空间(kernel)只包含 0, 因为对所有 $x \neq 0$ 有 $x^T A x > 0$ 。这意味着如果 $x \neq 0$ 则 $Ax \neq 0$ 。 A 的对角元素 a_{ii} 都是正的, 因为 $a_{ii} = e_i^T A e_i > 0$, 其中 e_i 是 $\mathbb{R}^n$ 中标准基的第 i 个向量。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>.] 3.3 长度和距离 75

3.3 长度和距离

在第 3.1 节中，我们已经讨论了可以用来计算向量长度的范数。内积和范数密切相关，任何内积都自然地诱导出一个范数 [内积诱导范数。]

$$\|x\| := \sqrt{\langle x, x \rangle} \quad (3.16)$$

这样我们可以使用内积来计算向量的长度。然而，并非每个范数都由内积诱导。Manhattan 范数 [(3.3)] 是一个没有对应内积的范数的例子。在下面，我们将重点关注由内积诱导的范数，并引入几何概念，如长度、距离和角度。

备注 (Cauchy-Schwarz 不等式)。对于内积向量空间 $(V, \langle \cdot, \cdot \rangle)$ ，诱导范数 $\|\cdot\|$ 满足 Cauchy-Schwarz 不等式 [Cauchy-Schwarz 不等式]

$$|\langle x, y \rangle| \leq \|x\| \|y\|. \quad (3.17)$$

◇

例 3.5 (使用内积计算向量长度)

在几何学中，我们经常关心向量的长度。现在我们可以使用内积通过 [(3.16)] 来计算它们。让我们取 $x = [1, 1]^T \in \mathbb{R}^2$ 。如果我们使用点积作为内积，通过 [(3.16)] 我们得到

$$\|x\| = \sqrt{\langle x, x \rangle} = \sqrt{1^2 + 1^2} = \sqrt{2} \quad (3.18)$$

作为 x 的长度。现在让我们选择一个不同的内积：

$$\langle x, y \rangle := x^T [2 \ -1; -1 \ 1] y = 2x_1y_1 - (x_1y_2 + x_2y_1) + x_2y_2. \quad (3.19)$$

如果我们计算向量的范数，那么当 x_1 和 x_2 符号相同 (且 $x_1x_2 > 0$) 时，这个内积返回比点积更小的值；否则，它返回比点积更大的值。使用这个内积，我们得到

$$\langle x, x \rangle = 2x_1^2 - 2x_1x_2 + x_2^2 = 2 \cdot 1 - 2 \cdot 1 \cdot 1 + 1 = 1 \Rightarrow \|x\| = 1 \quad (3.20)$$

所以使用这个内积时 x 比使用点积时更“短”。

定义 3.6 (距离和度量)。考虑内积空间 $(V, \langle \cdot, \cdot \rangle)$ 。那么

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

称为 $x, y \in V$ 之间的 *距离*。如果我们使用点积作为内积，那么距离称为 *欧几里得距离*。[距离] [欧几里得距离]

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).] 76 [解析几何]

映射

$$d: V \times V \rightarrow \mathbb{R} \quad (x, y) \mapsto d(x, y) \quad (3.23)$$

称为度量。[度量]

备注。与向量的长度类似，向量之间的距离不需要内积：范数就足够了。如果我们有一个由内积诱导的范数，距离可能会根据内积的选择而变化。◇

度量 d 满足以下性质：

1. d 是正定的，即对所有 $x, y \in V$ 有 $d(x, y) \geq 0$ ，且 $d(x, y) = 0 \Leftrightarrow x = y$ 。[正定]
2. d 是对称的，即对所有 $x, y \in V$ 有 $d(x, y) = d(y, x)$ 。[对称]
3. 三角不等式：对所有 $x, y, z \in V$ 有 $d(x, z) \leq d(x, y) + d(y, z)$ 。[三角不等式]

备注。乍一看，内积和度量的性质列表非常相似。然而，通过比较定义 3.3 和定义 3.6，我们观察到 $\langle x, y \rangle$ 和 $d(x, y)$ 的行为方向相反。非常相似的 x 和 y 会导致内积的大值和度量的小值。◇

3.4 角度和正交性

图 3.4 [当限制到 $[0, \pi]$ 时, $f(\omega) = \cos(\omega)$ 返回区间 $[-1, 1]$ 中的唯一数。] 除了能够定义向量的长度以及两个向量之间的距离外, 内积还通过定义两个向量之间的角度 ω 来捕获向量空间的几何性质。我们使用 Cauchy-Schwarz 不等式 [(3.17)] 在内积空间中定义向量 x, y 之间的角度 ω , 这个概念与我们在 $\mathbb{R}^1$ 、 $\mathbb{R}^2$ 和 $\mathbb{R}^3$ 中的直觉一致。假设 $x \neq 0, y \neq 0$ 。那么

$$-1 \leq \langle x, y \rangle / (\|x\| \|y\|) \leq 1. \quad (3.24)$$

$[\cos(\omega)]$

$[-1] \quad [0] \quad [\pi/2] \quad [\pi] \quad [0] \quad 1$

因此, 存在唯一的 $\omega \in [0, \pi]$, 如图3.4所示, 其中

$$\cos \omega = \langle x, y \rangle / (\|x\| \|y\|) \quad (3.25)$$

数值 ω 是向量 x 和 y 之间的角度。直观地说, 两个向量之间的角度告诉我们它们的方向有多相似。例如, 使用点积, x 和 $y = 4x$ 之间的角度 (即 y 是 x 的缩放版本) 为0: 它们的方向相同。

3.4 角度和正交性

例3.6 (向量间的角度)

让我们计算 $\mathbf{x} = [1, 1]^T \in \mathbb{R}^2$ 和 $\mathbf{y} = [1, 2]^T \in \mathbb{R}^2$ 之间的角度；参见图3.5，其中我们使用点积作为内积。然后我们得到

$$\cos \omega = \langle \mathbf{x}, \mathbf{y} \rangle / \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle \langle \mathbf{y}, \mathbf{y} \rangle} = (\mathbf{x}^T \mathbf{y}) / \sqrt{(\mathbf{x}^T \mathbf{x}) (\mathbf{y}^T \mathbf{y})} = 3/\sqrt{10} \quad (3.26)$$

两个向量之间的角度是 $\arccos(3/\sqrt{10}) \approx 0.32 \text{ rad}$ ，对应约 18° 。

内积的一个关键特征是它还允许我们描述正交向量的特征。

定义3.7 (正交性)。如果且仅当 $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ 时，两个向量 $\mathbf{x}$ 和 $\mathbf{y}$ 是正交的，我们记作 $\mathbf{x} \perp \mathbf{y}$ 。如果另外 $\|\mathbf{x}\| = 1 = \|\mathbf{y}\|$ ，即这些向量是单位向量，那么 $\mathbf{x}$ 和 $\mathbf{y}$ 是标准正交的。

这个定义的一个含义是零向量与向量空间中的每个向量都正交。

备注。正交性是垂直概念对不一定是点积的双线性形式的推广。在我们的上下文中，从几何角度来看，我们可以将正交向量视为相对于特定内积成直角。◇

例3.7 (正交向量)

考虑两个向量 $\mathbf{x} = [1, 1]^T$, $\mathbf{y} = [-1, 1]^T \in \mathbb{R}^2$ ；参见图3.6。我们有兴趣使用两种不同的内积确定它们之间的角度 ω 。使用点积作为内积得到 $\mathbf{x}$ 和 $\mathbf{y}$ 之间的角度 ω 为 90° ，使得 $\mathbf{x} \perp \mathbf{y}$ 。但是，如果我们选择内积

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T [2 \ 0; 0 \ 1] \mathbf{y} \quad (3.27)$$

我们得到 $\mathbf{x}$ 和 $\mathbf{y}$ 之间的角度 ω 由下式给出

$$\cos \omega = \langle \mathbf{x}, \mathbf{y} \rangle / (\|\mathbf{x}\| \|\mathbf{y}\|) = -1/3 \Rightarrow \omega \approx 1.91 \text{ rad} \approx 109.5^\circ \quad (3.28)$$

并且 $\mathbf{x}$ 和 $\mathbf{y}$ 不正交。因此，相对于一个内积正交的向量不一定相对于不同的内积正交。

定义3.8 (正交矩阵)。方阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 是正交矩阵当且仅当其列是标准正交的，使得

$$\mathbf{A} \mathbf{A}^T = \mathbf{I} = \mathbf{A}^T \mathbf{A} \quad (3.29)$$

这意味着

$$\mathbf{A}^{-1} = \mathbf{A}^T \quad (3.30)$$

即逆矩阵通过简单地转置矩阵获得。

正交矩阵的变换是特殊的，因为当使用正交矩阵 $\mathbf{A}$ 变换向量 $\mathbf{x}$ 时，向量 $\mathbf{x}$ 的长度不会改变。对于点积，我们得到

$$\|\mathbf{A}\mathbf{x}\|^2 = (\mathbf{A}\mathbf{x})^T (\mathbf{A}\mathbf{x}) = \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{x}^T \mathbf{I} \mathbf{x} = \mathbf{x}^T \mathbf{x} = \|\mathbf{x}\|^2 \quad (3.31)$$

此外，任意两个向量 x 、 y 之间的角度（由它们的内积测量）在使用正交矩阵 A 变换它们时也保持不变。假设点积作为内积，图像 Ax 和 Ay 的角度由下式给出

$$\cos \omega = (Ax)^T(Ay) / (\|Ax\| \|Ay\|) = (x^T A^T A y) / \sqrt{(x^T A^T A x)(y^T A^T A y)} = (x^T y) / (\|x\| \|y\|) \quad (3.32)$$

这给出了 x 和 y 之间的确切角度。这意味着具有 $A^T = A^{-1}$ 的正交矩阵保持角度和距离。事实上，正交矩阵定义了旋转变换（可能带有翻转）。在第3.9节中，我们将讨论关于旋转的更多细节。

3.5 标准正交基

在第2.6.1节中，我们描述了基向量的性质，发现在 n 维向量空间中，我们需要 n 个基向量，即 n 个线性无关的向量。在第3.3节和第3.4节中，我们使用内积计算向量的长度和向量之间的角度。在下面，我们将讨论基向量彼此正交且每个基向量的长度为1的特殊情况。我们将这样的基称为标准正交基。

让我们更正式地介绍这一点。

定义3.9 (标准正交基)。考虑 n 维向量空间 V 和 V 的基 $\{b_1, \dots, b_n\}$ 。如果

$$\langle b_i, b_j \rangle = 0 \text{ 对于 } i \neq j \quad (3.33) \quad \langle b_i, b_i \rangle = 1 \quad (3.34)$$

对于所有 $i, j = 1, \dots, n$ 成立，那么该基被称为标准正交基 (ONB)。

如果只满足(3.33)，那么该基被称为正交基。注意(3.34)意味着每个基向量的长度/范数为1。

回顾第[2.6.1]节，我们可以使用高斯消元法来找到由一组向量张成的向量空间的基。假设我们给定一组非正交且未规范化的基向量 $\{\tilde{b}_1, \dots, \tilde{b}_n\}$ 。我们将它们连接成矩阵 $\tilde{B} = [\tilde{b}_1, \dots, \tilde{b}_n]$ ，并对增广矩阵（第[2.3.2]节） $[\tilde{B}^T \tilde{B} \mid \tilde{B}]$ 应用高斯消元法来获得标准正交基。这种迭代构建标准正交基 $\{b_1, \dots, b_n\}$ 的构造性方法称为 *Gram-Schmidt过程* [(Strang, 2003).]

例3.8 (标准正交基)

欧几里得向量空间 $\mathbb{R}^n$ 的标准基是一个标准正交基，其中内积是向量的点积。

在 $\mathbb{R}^2$ 中，向量

$$b_1 = \sqrt{1/2}[1, 1]^T, b_2 = \sqrt{1/2}[1, -1]^T \quad (3.35)$$

构成标准正交基，因为 $b_1^T b_2 = 0$ 且 $\|b_1\| = 1 = \|b_2\|$ 。

我们将在第[12]章和第[10]章中利用标准正交基的概念，当我们讨论支持向量机和主成分分析时。{#calibre_link-531.calibre3}

3.6 正交补

定义了正交性之后，我们现在来看彼此正交的向量空间。这在第[10]章中将起到重要作用，当我们从几何角度讨论线性降维时。

考虑一个 D 维向量空间 V 和一个 M 维子空间 $U \subseteq V$ 。那么它的正交补 $U^\perp$ 是 V 的一个 $(D-M)$ 维子空间，包含 V 中与 U 中每个向量都正交的所有向量。此外， $U \cap U^\perp = \{0\}$ ，因此 V 中的任何向量 x 都可以

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

解析几何

[图3.7] [三维向量空间中的平面 U 可以用其法向量来描述，该法向量张成其正交补 $U^\perp$ 。]

被唯一分解为

$$x = \sum_{m=1}^M \lambda_m b_m + \sum_{j=1}^{D-M} \psi_j b_{\perp j}, \lambda_m, \psi_j \in \mathbb{R}, (3.36)$$

其中 $(b_1, \dots, b_M)$ 是 U 的基， $(b_{\perp 1}, \dots, b_{\perp D-M})$ 是 $U^\perp$ 的基。

因此，正交补也可以用来描述三维向量空间中的平面 U （二维子空间）。更具体地说，与平面 U 正交且 $\|w\| = 1$ 的向量 w 是 $U^\perp$ 的基向量。图[3.7]说明了这种设置。所有与 w 正交的向量必须（根据构造）位于平面 U 中。向量 w 称为 U 的法向量。

一般来说，正交补可以用来描述 n 维向量空间和仿射空间中的超平面。

3.7 函数的内积

到目前为止，我们研究了内积的性质来计算长度、角度和距离。我们专注于有限维向量的内积。接下来，我们将看一个不同类型向量内积的例子：函数的内积。

我们迄今讨论的内积是为具有有限条目数的向量定义的。我们可以将向量 $x \in \mathbb{R}^n$ 看作具有 n 个函数值的函数。内积的概念可以推广到具有无限条目数的向量（可数无限）以及连续值函数（不可数无限）。然后，对向量各个分量的求和（例如见方程[(3.5)]）转变为积分。

两个函数 $u: \mathbb{R} \rightarrow \mathbb{R}$ 和 $v: \mathbb{R} \rightarrow \mathbb{R}$ 的内积可以定义为定积分

$$\langle u, v \rangle := \int [a \text{ to } b] u(x)v(x)dx \quad (3.37)$$

对于下限和上限分别为 $a, b < \infty$ 。与我们通常的内积一样，我们可以通过内积来定义范数和正交性。如果[(3.37)]计算结果为0，则函数 u 和 v 是正交的。为了使前述内积在数学上精确，我们需要处理测度和积分的定义，这导致Hilbert空间的定义。此外，与有限维向量上的内积不同，函数上的内积可能发散（具有无限值）。所有这些都需要深入实分析和函数分析的一些更复杂的细节，这些我们在本书中不涉及。

例3.9 (函数的内积)

如果我们选择 $u = \sin(x)$ 和 $v = \cos(x)$ ，则[(3.37)]的被积函数 $f(x) = u(x)v(x)$ 如图[3.8]所示。我们看到这个函数是奇函数，即 $f(-x) = -f(x)$ 。因此，在限制 $a = -\pi, b = \pi$ 下该乘积的积分计算结果为0。因此， $\sin$ 和 $\cos$ 是正交函数。

[图3.8] $[f(x) = \sin(x)\cos(x).]$

注释。还成立的是函数集合

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

如果我们从 $-\pi$ 积分到 π ，是正交的，即任何一对函数彼此正交。[(3.38)]中的函数集合张成在 $[-\pi, \pi]$ 上偶函数和周期函数的一个大子空间，将函数投影到这个子空间是Fourier级数背后的基本思想。◇

在第[6.4.6]节中，我们将看一个第二种非常规内积类型：随机变量的内积。

3.8 正交投影

投影是线性变换的一个重要类别（除了旋转和反射），在图形学、编码理论、统计学和机器学习中发挥重要作用。在机器学习中，我们经常处理高维数据。高维数据通常很难分析或可视化。然而，高维数据经常具有这样的性质：只有少数几个维度包含大部分信息，而大多数其他维度对于描述数据的关键属性并不重要。当我们压缩或可视化高维数据时，我们将丢失信息。为了最小化这种压缩损失，我们理想情况下要找到数据中最具信息性的维度。正如第1章中讨论的，数据可以表示为向量，在本章中，我们将讨论数据压缩的一些基本工具。更具体地说，我们可以将原始高维数据投影到低维特征空间，并在这个低维空间中工作，以更多地了解数据集并提取相关模式。例如，机器学习算法，如Pearson (1901)和Hotelling (1933)的主成分分析(PCA)以及深度神经网络（例如，深度自编码器(Deng et al., 2010)），大量利用了降维的思想。在接下来的内容中，我们将专注于正交投影，我们将在第10章中用于线性降维，在第12章中用于分类。甚至我们在第9章中讨论的线性回归，也可以使用正交投影来解释。对于给定的低维子空间，高维数据的正交投影保留尽可能多的信息，并最小化原始数据与相应投影之间的差异/误差。图3.9给出了这种正交投影的示例。在我们详细说明如何获得这些投影之前，让我们定义什么是投影。

定义3.10（投影）。设 V 是一个向量空间， $U \subseteq V$ 是 V 的子空间。线性映射 $\pi: V \rightarrow U$ 被称为*投影*，如果 $\pi^2 = \pi \circ \pi = \pi$ 。

由于线性映射可以用变换矩阵表示（参见第2.7节），前面的定义同样适用于一种特殊的变换矩阵，即*投影矩阵* P_π ，它具有 $P_\pi^2 = P_\pi$ 的性质。

接下来，我们将推导内积空间 $(\mathbb{R}^n, \langle \cdot, \cdot \rangle)$ 中向量到子空间的正交投影。我们将从一维子空间开始，这也被称为*直线*。如果没有另外提及，我们假设点积 $\langle x, y \rangle = x^T y$ 作为内积。

3.8.1 投影到一维子空间（直线）

假设我们给定一条通过原点的直线（一维子空间），其基向量为 $\mathbf{b} \in \mathbb{R}^n$ 。直线是由 $\mathbf{b}$ 张成的一维子空间 $U \subseteq \mathbb{R}^n$ 。当我们将 $\mathbf{x} \in \mathbb{R}^n$ 投影到 U 时，我们寻找最接近 $\mathbf{x}$ 的向量 $\pi_U(\mathbf{x}) \in U$ 。使用几何论证，让我们描述投影 $\pi_U(\mathbf{x})$ 的一些性质（图3.10(a)作为说明）：

投影 $\pi_U(\mathbf{x})$ 最接近 $\mathbf{x}$ ，其中“最接近”意味着距离 $\|\mathbf{x} - \pi_U(\mathbf{x})\|$ 是最小的。由此可知，从 $\pi_U(\mathbf{x})$ 到 $\mathbf{x}$ 的线段 $\pi_U(\mathbf{x}) - \mathbf{x}$ 垂直于 U ，因此也垂直于 U 的基向量 $\mathbf{b}$ 。正交性条件产生 $\langle \pi_U(\mathbf{x}) - \mathbf{x}, \mathbf{b} \rangle = 0$ ，因为向量之间的角度通过内积定义。

$\mathbf{x}$ 到 U 的投影 $\pi_U(\mathbf{x})$ 必须是 U 的元素，因此必须是张成 U 的基向量 $\mathbf{b}$ 的倍数。因此， $\pi_U(\mathbf{x}) = \lambda \mathbf{b}$ ，对于某个 $\lambda \in \mathbb{R}$ 。

在接下来的三个步骤中，我们确定坐标 λ 、投影 $\pi_U(\mathbf{x}) \in U$ ，以及将任何 $\mathbf{x} \in \mathbb{R}^n$ 映射到 U 的投影矩阵 P_π ：

1. 寻找坐标 λ 。正交性条件产生 $\langle \mathbf{x} - \pi_U(\mathbf{x}), \mathbf{b} \rangle = 0 \Leftrightarrow \langle \mathbf{x} - \lambda \mathbf{b}, \mathbf{b} \rangle = 0$ (3.39)

我们现在可以利用内积的双线性性，得到 $\langle \mathbf{x}, \mathbf{b} \rangle - \lambda \langle \mathbf{b}, \mathbf{b} \rangle = 0 \Leftrightarrow \lambda = \langle \mathbf{x}, \mathbf{b} \rangle / \langle \mathbf{b}, \mathbf{b} \rangle = \langle \mathbf{b}, \mathbf{x} \rangle / \|\mathbf{b}\|^2$ (3.40)

在最后一步中，我们利用了内积是对称的这一事实。如果我们选择 $\langle \cdot, \cdot \rangle$ 为点积，我们得到 $\lambda = \mathbf{b}^T \mathbf{x} / \mathbf{b}^T \mathbf{b} = \mathbf{b}^T \mathbf{x} / \|\mathbf{b}\|^2$ (3.41)

如果 $\|\mathbf{b}\| = 1$ ，那么投影的坐标 λ 由 $\mathbf{b}^T \mathbf{x}$ 给出。



图3.9：数据集（蓝点）到一维子空间（直线）的正交投影（橙点）

图3.9：数据集（蓝点）到一维子空间（直线）的正交投影（橙点）



图3.10：投影到一维子空间的示例

图3.10：投影到一维子空间的示例

2. 寻找投影点 $\pi_U(\mathbf{x}) \in U$ 。由于 $\pi_U(\mathbf{x}) = \lambda \mathbf{b}$ ，我们立即从(3.40)得到

$$\pi_U(\mathbf{x}) = \lambda \mathbf{b} = \frac{\langle \mathbf{b}, \mathbf{x} \rangle}{\|\mathbf{b}\|^2} \mathbf{b} \quad (3.42)$$

其中最后的等式仅对点积成立。我们也可以通过定义3.1计算 $\pi_U(\mathbf{x})$ 的长度：

$$\|\pi_U(\mathbf{x})\| = \|\lambda \mathbf{b}\| = |\lambda| \|\mathbf{b}\| \quad (3.43)$$

因此，我们的投影长度是 $|\lambda|$ 倍的 $\mathbf{b}$ 的长度。这也提供了直观理解： λ 是 $\pi_U(\mathbf{x})$ 相对于张成一维子空间 U 的基向量 $\mathbf{b}$ 的坐标。

如果我们使用点积作为内积，得到

$$\|\pi_U(\mathbf{x})\| = \|\mathbf{b}\| |\cos \omega| = \|\mathbf{x}\| \|\mathbf{b}\| |\cos \omega| = \|\mathbf{x}\| \|\mathbf{b}\| \cos \omega$$

这里， ω 是 x 和 b 之间的夹角。这个方程在三角学中应该很熟悉：如果 $\|x\| = 1$ ，那么 x 位于单位圆上。水平轴是一维子空间，因此投影到由 b 张成的水平轴上正好是 $\cos \omega$ ，对应向量 $\pi[U](x) = |\cos \omega|$ 的长度。图3.10(b)给出了说明。

3. 寻找投影矩阵 $P[\pi]$ 。我们知道投影是线性映射（见定义3.10）。因此，存在投影矩阵 $P[\pi]$ ，使得 $\pi[U](x) = P[\pi]x$ 。以点积作为内积，有

$$\pi[U](x) = \lambda b = b\lambda = b \cdot x, \quad (3.45) \quad \|b\|^2 = \|b\|^2$$

我们立即看到

$$P[\pi] = bb^T / \|b\|^2. \quad (3.46)$$

投影矩阵总是对称的。注意 bb^T （因此 $P[\pi]$ ）是对称矩阵（秩为1），而 $\|b\|^2 = \langle b, b \rangle$ 是标量。

投影矩阵 $P[\pi]$ 将任何向量 $x \in \mathbb{R}^n$ 投影到通过原点且方向为 b 的直线上（等价地，由 b 张成的子空间 U ）。

备注。投影 $\pi U \in \mathbb{R}^n$ 仍然是 n 维向量而不是标量。然而，如果我们想用张成子空间 U 的基向量 b 来表示投影，我们不再需要 n 个坐标来表示投影，只需要一个： λ 。◇

图3.11 投影到二维子空间 U ，基为 b_1, b_2 。 $x \in \mathbb{R}^3$ 到 U 的投影 πU 可以表示为 b_1, b_2 的线性组合，位移向量 $x - \pi U$ 与 b_1 和 b_2 都正交。

例3.10 (投影到直线)

找到投影矩阵 $P[\pi]$ ，投影到通过原点由 $b = [1 \ 2 \ 2]^T$ 张成的直线。 b 是一维子空间（通过原点的直线）的方向和基。

利用(3.46)，我们得到

$$P[\pi] = bb^T / b^T b = (1/9)[1 \ 2 \ 2][1 \ 2 \ 2] = (1/9)[1 \ 2 \ 2; 2 \ 4 \ 4; 2 \ 4 \ 4] \quad (3.47)$$

现在选择特定的 x ，看它是否位于由 b 张成的子空间中。对于 $x = [1 \ 1 \ 1]^T$ ，投影是

$$\pi U = P[\pi]x = (1/9)[1 \ 2 \ 2; 2 \ 4 \ 4; 2 \ 4 \ 4][1; 1; 1] = (1/9)[5; 10; 10] \in \text{span}[2]. \quad (3.48)$$

注意 $P[\pi]$ 应用于 πU 不会改变任何东西，即 $P[\pi]\pi U = \pi U$ 。这是预期的，因为根据定义3.10，我们知道投影矩阵 $P[\pi]$ 满足 $P^2[\pi]x = P[\pi]x$ 对所有 x 成立。

备注。利用第4章的结果，我们可以证明 πU 是 $P[\pi]$ 的特征向量，对应的特征值是1。◇

3.8.2 投影到一般子空间

如果U由一组张成向量给出，在不是基的情况下，确保在继续之前确定基 $b_1, \dots, b[m]$ 。

接下来，我们考虑向量 $x \in R^n$ 到低维子空间 $U \subseteq R^n$ 的正交投影，其中 $\dim(U) = m \geq 1$ 。图3.11给出了说明。

假设 $(b_1, \dots, b[m])$ 是U的有序基。任何到U的投影 πU 必然是U的元素。因此，它们可以表示为U的基向量 $b_1, \dots, b[m]$ 的线性组合，使得 $\pi U = \sum_{i=1}^m \lambda_i b_i$ 。

基向量构成 $B \in R^{(n \times m)}$ 的列，其中 $B = [b_1, \dots, b[m]]$ 。

如同1D情况，我们遵循三步程序来找到投影 πU 和投影矩阵 $P[\pi]$:

1. 找到投影的坐标 $\lambda_1, \dots, \lambda[m]$ （相对于U的基），使得线性组合

$$\pi U = \sum_{i=1}^m \lambda_i b_i = B\lambda, \quad (3.49)$$

$$B = [b_1, \dots, b[m]] \in R^{(n \times m)}, \lambda = [\lambda_1, \dots, \lambda[m]]^T \in R^m, \quad (3.50)$$

最接近 $x \in R^n$ 。如同1D情况，“最接近”意味着“最小距离”，这意味着连接 $\pi U \in U$ 和 $x \in R^n$ 的向量必须与U的所有基向量正交。因此，我们得到m个同时条件（假设点积作为内积）

$$\langle b_1, x - \pi U \rangle = b_1^T (x - \pi U) = 0 \quad (3.51)$$

...

$$\langle b[m], x - \pi U \rangle = b[m]^T (x - \pi U) = 0 \quad (3.52)$$

利用 $\pi U = B\lambda$ ，可写成

$$b_1^T (x - B\lambda) = 0 \quad (3.53)$$

...

$$b[m]^T (x - B\lambda) = 0 \quad (3.54)$$

从而得到齐次线性方程组

$$b_1^T; \dots; b[m]^T = B^T (x - B\lambda) = 0 \quad (3.55)$$

$$\Leftrightarrow B^T B \lambda = B^T x. \quad (3.56)$$

最后的表达式称为正规方程(normal equation)。由于 $b_1, \dots, b[m]$ 是U的基，因此线性无关， $B^T B \in R^{(m \times m)}$ 是正则的且可逆。这允许我们求解系数/坐标

$$\lambda = (B^T B)^{-1} B^T x. \quad (3.57)$$

伪逆矩阵 矩阵 $(B^T B)^{-1} B^T$ 也被称为 B 的 *伪逆矩阵* (pseudo-inverse)，它可以为非方形矩阵 B 计算。它只要求 $B^T B$ 是正定的，当 B 是满秩的时候就是这种情况。在实际应用中（例如，线性回归），我们经常向 $B^T B$ 添加一个“扰动项” ϵI 来保证增强的数值稳定性和正定性。这个“岭”可以通过贝叶斯推理严格推导得出。详见第9章。

2. 寻找投影 $\pi_U(x) \in U$ 。我们已经确定 $\pi_U(x) = B\lambda$ 。因此，由(3.57)

$$\pi_U(x) = B(B^T B)^{-1} B^T x. \quad (3.58)$$

3. 寻找投影矩阵 P_π 。从(3.58)，我们可以立即看出解决 $P_\pi x = \pi_U(x)$ 的投影矩阵必须是

$$P_\pi = B(B^T B)^{-1} B^T. \quad (3.59)$$

注释。 投影到一般子空间的解包括1D情况作为特例：如果 $\dim(U) = 1$ ，那么 $B^T B \in \mathbb{R}$ 是一个标量，我们可以将(3.59)中的投影矩阵 $P_\pi = B(B^T B)^{-1} B^T$ 重写为

$$P_\pi = B B^T / (B^T B), \text{ 这正是(3.46)中的投影矩阵。} \diamond$$

示例3.11（投影到二维子空间）

对于子空间 $U = \text{span}\left\{\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}\right\} \subseteq \mathbb{R}^3$ 和 $x = \begin{bmatrix} 6 \\ 1 \\ 3 \end{bmatrix} \in \mathbb{R}^3$ ，找出 x 关于子空间 U 的坐标 λ 、投影点 $\pi_U(x)$ 和投影矩阵 P_π 。

首先，我们看到 U 的生成集是一个基（线性无关）并将 U 的基向量写入矩阵 $B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 2 \end{bmatrix}$

其次，我们计算矩阵 $B^T B$ 和向量 $B^T x$ 如下

$$B^T B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix}, B^T x = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 6 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ 6 \end{bmatrix} \quad (3.60)$$

第三，我们求解正规方程 $B^T B \lambda = B^T x$ 来找到 λ ：

$$\begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 6 \end{bmatrix} \iff \lambda = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \quad (3.61)$$

第四， x 到 U 的投影 $\pi_U(x)$ ，也就是到 B 的列空间的投影，可以直接通过以下方式计算

$$\pi_U(x) = B \lambda = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \\ 10 \end{bmatrix} \quad (3.62)$$

投影误差 相应的投影误差 (projection error) 是原向量与其在 U 上投影之间差向量的范数，即

$$\|x - \pi_U(x)\| = \left\| \begin{bmatrix} 6 \\ 1 \\ 3 \end{bmatrix} - \begin{bmatrix} 5 \\ 10 \\ 10 \end{bmatrix} \right\| = \sqrt{6} \quad (3.63)$$

投影误差也被称为重构误差 (reconstruction error)。

第五，投影矩阵（对于任何 $x \in \mathbb{R}^3$ ）由以下给出

$$P_\pi = B(B^T B)^{-1} B^T = 1/6 \begin{bmatrix} 5 & 2 & -1 \\ 2 & 2 & 2 \\ -1 & 2 & 5 \end{bmatrix} \quad (3.64)$$

为了验证结果，我们可以 (a) 检查位移向量 $\pi_U(x) - x$ 是否与 U 的所有基向量正交，以及 (b) 验证 $P_\pi^2 = P_\pi$ （见定义 3.10）。

注释。 投影 $\pi_U(x)$ 仍然是 $\mathbb{R}^n$ 中的向量，尽管它们位于 m 维子空间 $U \subseteq \mathbb{R}^n$ 中。然而，为了表示一个投影向量，我们只需要关于 U 的基向量 $b_1, \dots, b_m$ 的 m 个坐标 $\lambda_1, \dots, \lambda_m$ 。◇

注释。 在带有一般内积的向量空间中，我们在计算角度和距离时必须注意，这些是通过内积定义的。◇

我们可以找到无解线性方程组的近似解 投影允许我们处理线性系统 $Ax = b$ 没有解的情况。回想这意味着 b 不在 A 的张成空间中，也就是说，向量 b 不在由 A 的列张成的子空间中。**使用投影。** 给定线性方程不能精确求解，我们可以找到一个近似解。思路是在由 A 的列张成的子空间中找到最接近 b 的向量，也就是说，我们计算 b 到由 A 的列张成的子空间的正交投影。这个问题在实践中经常出现，其解被称为**最小二乘解** (least-squares solution)（假设点积作为内积）的超定系统。这在第 9.4 节中进一步讨论。使用重构误差 (3.63) 是推导主成分分析的一种可能方法（第 10.3 节）。

注释。 我们刚刚看了向量 x 到具有基向量 $\{b_1, \dots, b_k\}$ 的子空间 U 的投影。如果这个基是标准正交基，也就是说，满足 (3.33) 和 (3.34)，投影方程 (3.58) 大大简化为

$$\pi_U(x) = BB^T x \quad (3.65)$$

因为 $B^T B = I$ ，坐标为

$$\lambda = B^T x. \quad (3.66)$$

这意味着我们不再需要计算(3.58)中的逆，这节省了计算时间。◇

3.8.3 Gram-Schmidt正交化

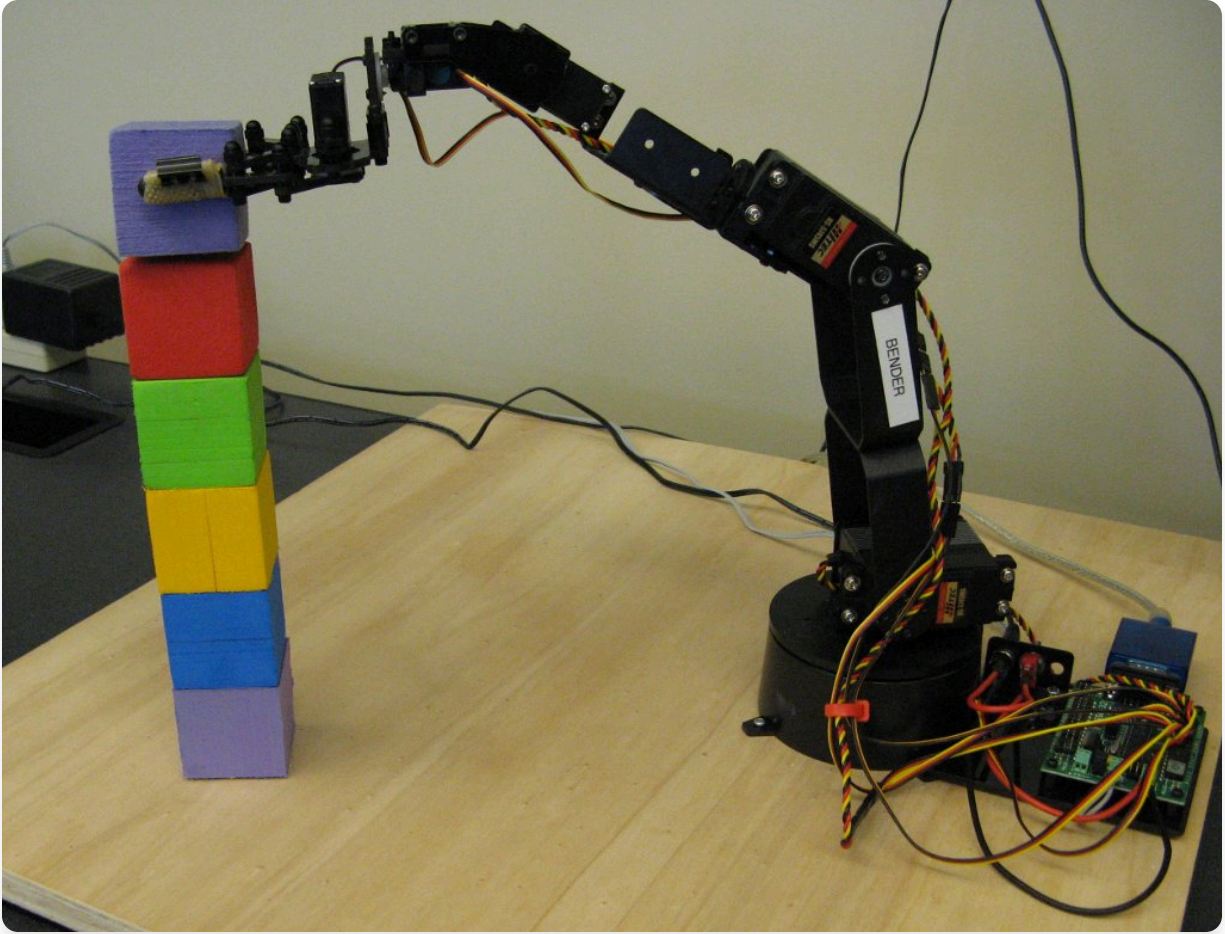
投影是Gram-Schmidt方法的核心，该方法允许我们构造性地将 n 维向量空间 V 的任何基 $(b_1, \dots, b_n)$ 转换为 V 的正交/标准正交基 $(u_1, \dots, u_n)$ 。这个基总是存在的 (Liesen and Mehrmann, 2015)，并且 $\text{span}[b_1, \dots, b_n] = \text{span}[u_1, \dots, u_n]$ 。**Gram-Schmidt正交化**方法迭代地从 V 的任何基 $(b_1, \dots, b_n)$ 构造正交基 $(u_1, \dots, u_n)$ ，如下所示：

$$u_1 := b_1 \quad (3.67) \quad u_k := b_k - \pi_{\{\text{span}[u_1, \dots, u_{k-1}]\}}(b_k), \quad k = 2, \dots, n. \quad (3.68)$$

在公式(3.68)中，第 k 个基向量 $b[k]$ 被投影到由前 $k-1$ 个已构造的正交向量 $u_1, \dots, u_{k-1}$ 张成的子空间上；参见第3.8.2节。然后从 $b[k]$ 中减去这个投影，得到一个与由 $u_1, \dots, u_{k-1}$ 张成的 $(k-1)$ 维子空间正交的向量 $u[k]$ 。对所有 n 个基向量 $b_1, \dots, b_n$ 重复这个过程，得到 V 的正交基 $(u_1, \dots, u_n)$ 。如果我们将 $u[k]$ 标准化，就得到一个ONB(正交标准基)，其中 $\|u[k]\| = 1, \quad k = 1, \dots, n$ 。

示例3.12 (Gram-Schmidt正交化)

图3.12



Gram-Schmidt正交化。(a) $\mathbb{R}^2$ 的非正交基 (b_1, b_2) ；(b)第一个构造的基向量 u_1 和 b_2 在 $\text{span}[u_1]$ 上的投影 $\pi_{\text{span}u_1}$ ；(c)正交基向量 u_1 和 $u_2 = b_2 - \pi_{\text{span}u_1}$ 考虑 $\mathbb{R}^2$ 的基 $(b_1, b[2])$ ，其中

$$b_1 = [2; 0], \quad b[2] = [1; 1]; \quad (3.69)$$

另见图3.12(a)。使用Gram-Schmidt方法，我们按以下方式构造 $\mathbb{R}^2$ 的正交基 $(u_1, u[2])$ （假设点积作为内积）：

$$u_1 := b_1 = [2; 0], \quad (3.70)$$

$$u[2] := b[2] - \pi_{\text{span}u_1} = b[2] - (b[2]^T u_1 / \|u_1\|^2) u_1 = [1; 1] - (2/4)[2; 0] = [0; 1]。 \quad (3.71)$$

这些步骤在图3.12(b)和(c)中进行了说明。我们立即看到 u_1 和 $u[2]$ 是正交的，即 $u_1^T u[2] = 0$ 。

3.8.4 仿射子空间上的投影

到目前为止，我们讨论了如何将向量投影到低维子空间 U 上。接下来，我们提供将向量投影到仿射子空间的解决方案。

考虑图3.13(a)中的设置。给定仿射空间 $L = x[0] + U$ ，其中 b_1, b_2 是 U 的基向量。为了确定 x 在 L 上的正交投影 π_L ，我们将问题转换为我们知道如何解决的问题：向量子空间上的投影。为了达到这个目的，我们从 x 和 L 中减去支撑点 $x[0]$ ，使得 $L - x[0] = U$ 正好是向量子空间 U 。现在我们可以使用第3.8.2节中讨论的子空间上的正交投影，得到投影 π_U ，如图3.13(b)所示。现在可以通过加上 $x[0]$ 将这个投影翻译回 L ，从而得到仿射空间 L 上的正交投影为

$$\pi_L = x[0] + \pi_U, \quad (3.72)$$

其中 π_U 是到子空间 U 上的正交投影，即 L 的方向空间；见图3.13(c)。

图3.13 仿射空间上的投影。(a)原始设置；(b)通过减去 x_0 进行平移，使得可以投影到方向空间 $U = L - x_0$ 上；(c)投影被翻译回 $x_0 + \pi_U(x - x_0)$ ，这给出了最终的正交投影 $\pi_L(x)$ 。

从图3.13还可以明显看出， x 到仿射空间 L 的距离与 $x - x[0]$ 到 U 的距离相同，即

$$d(x, L) = \|x - \pi_L\| = \|x - (x[0] + \pi_U)\| \quad (3.73a) = d(x - x[0], \pi_U) = d(x - x[0], U). \quad (3.73b)$$

我们将使用仿射子空间上的投影来推导第12.1节中分离超平面的概念。

3.9 旋转

长度和角度保持，如第3.4节所讨论的，是具有正交变换矩阵的线性映射的两个特征。接下来，我们将更仔细地看看特定的正交变换矩阵，它们描述旋转。

旋转是一种线性映射（更具体地说，是欧几里得向量空间的自同构），它将平面绕原点旋转角度 θ ，即原点是一个不动点。对于正角度 $\theta > 0$ ，按照通常的约定，我们按逆时针方向旋转。图3.14中显示了一个例子，其中变换矩阵为

$$R = \begin{bmatrix} -0.38 & -0.92 \\ 0.92 & -0.38 \end{bmatrix}. \quad (3.74)$$

图3.14 旋转使平面中的物体绕原点旋转。如果旋转角度为正，我们按逆时针方向旋转。

图3.15 机器人手臂需要旋转其关节以拾取物体或正确放置它们。图片取自(Deisenroth等人，2015)。

旋转的重要应用领域包括计算机图形学和机器人学。例如，在机器人学中，了解如何旋转机器人手臂的关节以拾取或放置物体通常很重要，见图3.15。

图3.16 $\Phi(e_2) = [-\sin \theta, \cos \theta]$ 标准基在 $\mathbb{R}^2$ 中按角度 θ 的旋转。

$$\Phi(e_1) = [\cos \theta, \sin \theta]$$



图3.16：标准基在 $\mathbb{R}^2$ 中按角度 θ 的旋转

图3.16：标准基在 $\mathbb{R}^2$ 中按角度 θ 的旋转

3.9.1 $\mathbb{R}^2$ 中的旋转

考虑 $\mathbb{R}^2$ 的标准基 $\mathbf{e}_1 = [1, 0]$, $\mathbf{e}_2 = [0, 1]$, 它定义了 $\mathbb{R}^2$ 中的标准坐标系。我们的目标是将此坐标系按角度 θ 旋转, 如图3.16所示。注意旋转后的向量仍然线性无关, 因此是 $\mathbb{R}^2$ 的一组基。这意味着旋转执行基变换。

旋转 Φ 是线性映射, 因此我们可以用**旋转矩阵** $R(\theta)$ 来表示它们。三角学 (见图3.16) 允许我们确定旋转轴 (Φ 的像) 相对于 $\mathbb{R}^2$ 中标准基的坐标。我们得到

$$\Phi(\mathbf{e}_1) = [\cos \theta, \sin \theta], \Phi(\mathbf{e}_2) = [-\sin \theta, \cos \theta]. \quad (3.75)$$

因此, 执行基变换到旋转坐标的旋转矩阵 $R(\theta)$ 为

$$R(\theta) = [\Phi(\mathbf{e}_1) \ \Phi(\mathbf{e}_2)] = [\cos \theta \ -\sin \theta; \sin \theta \ \cos \theta]. \quad (3.76)$$

3.9.2 $\mathbb{R}^3$ 中的旋转

与 $\mathbb{R}^2$ 情况不同，在 $\mathbb{R}^3$ 中我们可以绕一维轴旋转任何二维平面。指定一般旋转矩阵的最简单方法是指定标准基 $\mathbf{e}_1$ 、 $\mathbf{e}_2$ 、 $\mathbf{e}_3$ 的像应该如何旋转，并确保这些像 $\mathbf{Re}_1$ 、 $\mathbf{Re}_2$ 、 $\mathbf{Re}_3$ 彼此正交规范。然后我们可以通过组合标准基的像来获得一般旋转矩阵 $\mathbf{R}$ 。

为了有一个有意义的旋转角，当我们在超过二维的空间中操作时，我们必须定义”逆时针”的含义。我们采用的约定是，绕轴的”逆时针”（平面）旋转是指当我们”正面”看轴，从末端向原点看时绕轴的旋转。在 $\mathbb{R}^3$ 中，因此有三个（平面）绕三个标准基向量的旋转（见图3.17）：

图3.17 在 $\mathbb{R}^3$ 中将向量（灰色）绕 $\mathbf{e}_3$ 轴按角度 θ 旋转。旋转后的向量以蓝色显示。



图3.17：在 $\mathbb{R}^3$ 中将向量绕 $\mathbf{e}_3$ 轴旋转

图3.17：在 $\mathbb{R}^3$ 中将向量绕 $\mathbf{e}_3$ 轴旋转
绕 $\mathbf{e}_1$ 轴的旋转

$$\mathbf{R}_1(\theta) = [\Phi(\mathbf{e}_1) \ \Phi(\mathbf{e}_2) \ \Phi(\mathbf{e}_3)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}. \quad (3.77)$$

这里， $\mathbf{e}_1$ 坐标是固定的，逆时针旋转在 $\mathbf{e}_2\mathbf{e}_3$ 平面中执行。

绕 $\mathbf{e}_2$ 轴的旋转

$$\mathbf{R}_2(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}. \quad (3.78)$$

如果我们绕 $\mathbf{e}_2$ 轴旋转 $\mathbf{e}_1\mathbf{e}_3$ 平面，我们需要从 $\mathbf{e}_2$ 轴的”顶端”向原点看。

绕 $\mathbf{e}_3$ 轴的旋转

$$\mathbf{R}_3(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.79)$$

图3.17说明了这一点。

3.9.3 n 维中的旋转

从2D和3D到 n 维欧几里得向量空间的旋转推广可以直观地描述为固定 $n-2$ 个维度，并将旋转限制在 n 维空间中的二维平面上。与三维情况一样，我们可以旋转任何平面（ $\mathbb{R}^n$ 的二维子空间）。

定义3.11（Givens旋转）。设 V 是 n 维欧几里得向量空间， $\Phi: V \rightarrow V$ 是具有变换矩阵的自同态

$$R_{ij}(\theta) := [\text{Identity matrix with } \cos \theta \text{ and } \sin \theta \text{ terms}] \in \mathbb{R}^{n \times n}, \quad (3.80)$$

对于 $1 \leq i < j \leq n$ 和 $\theta \in \mathbb{R}$ 。那么 $R_{ij}(\theta)$ 称为**Givens旋转**。

本质上， $R_{ij}(\theta)$ 是单位矩阵 I_n ，其中

$$r_{ii} = \cos \theta, r_{ij} = -\sin \theta, r_{ji} = \sin \theta, r_{jj} = \cos \theta. \quad (3.81)$$

在二维中（即 $n=2$ ），我们获得(3.76)作为特殊情况。

3.9.4 旋转的性质

旋转表现出许多有用的性质，这些性质可以通过将它们视为正交矩阵来导出（定义3.8）：

- 旋转保持距离，即 $\|x - y\| = \|R\theta(x) - R\theta(y)\|$ 。换句话说，旋转在变换后保持任意两点之间的距离不变。
- 旋转保持角度，即 $R\theta x$ 和 $R\theta y$ 之间的角度等于 x 和 y 之间的角度。
- 三维（或更多维）中的旋转通常不是可交换的。因此，应用旋转的顺序很重要，即使它们绕同一点旋转。只有在二维中向量旋转是可交换的，使得对于所有 $\varphi, \theta \in [0, 2\pi)$ ， $R(\varphi)R(\theta) = R(\theta)R(\varphi)$ 。只有当它们绕同一点（例如原点）旋转时，它们才形成一个阿贝尔群（在乘法下）。

3.10 进一步阅读

在本章中，我们简要概述了解析几何的一些重要概念，这些概念将在本书的后续章节中使用。

对于我们所介绍概念的更广泛和深入的概述，我们推荐以下优秀书籍：Axler (2015)和Boyd and Vandenberghe (2018)。

内积允许我们确定向量（子）空间的特定基，其中每个向量与所有其他向量正交（正交基），使用Gram-Schmidt方法。这些基在优化和求解线性方程组的数值算法中很重要。例如，Krylov子空间方法，如共轭梯度或广义最小残差方法(GMRES)，最小化彼此正交的残差误差(Stoer and Burlirsch, 2002)。

在机器学习中，内积在以下上下文中很重要

3.10 进一步阅读

kernel methods (Schölkopf and Smola, 2002)。Kernel方法利用了许多线性算法可以纯粹通过内积计算来表达的特点。然后，“kernel trick”允许我们在一个（潜在的无限维）特征空间中隐式地计算这些内积，甚至不需要明确知道这个特征空间。这使得机器学习中使用的许多算法能够“非线性化”，例如

用于降维的kernel-PCA (Schölkopf et al., 1997)。Gaussian processes (Rasmussen and Williams, 2006) 也属于kernel方法的范畴，是概率回归（拟合曲线到数据点）的当前最先进技术。kernel的思想在第12章中进一步探讨。

投影经常用于计算机图形学中，例如生成阴影。在优化中，正交投影经常用于（迭代地）最小化残差误差。这在机器学习中的应用，例如在线性回归中，我们想要找到一个（线性）函数来最小化残差误差，即数据到线性函数的正交投影的长度 (Bishop, 2006)。我们将在第9章中进一步研究这个问题。PCA (Pearson, 1901; Hotelling, 1933) 也使用投影来降低高维数据的维度。我们将在第10章中更详细地讨论这个问题。

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

练习题

3.1 证明对所有 $x = [x_1, x_2]^T \in \mathbb{R}^2$ 和 $y = [y_1, y_2]^T \in \mathbb{R}^2$ 定义的 $\langle \cdot, \cdot \rangle$

$$\langle x, y \rangle := x_1 y_1 - (x_1 y_2 + x_2 y_1) + 2(x_2 y_2)$$

是一个内积。

3.2 考虑 $\mathbb{R}^2$ ，其中对所有 x 和 y 在 $\mathbb{R}^2$ 中的 $\langle \cdot, \cdot \rangle$ 定义为

$$\langle x, y \rangle := x^T A y, \text{ 其中 } A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$\langle \cdot, \cdot \rangle$ 是内积吗？

3.3 计算距离，其中

$$x = [1; 2; 3], \quad y = [-1; -1; 0]$$

使用

a. $\langle x, y \rangle := x^T y$

b. $\langle x, y \rangle := x^T A y, \quad A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}$

3.4 计算角度，其中

$$x = [1; 2], \quad y = [-1; -1]$$

使用

a. $\langle x, y \rangle := x^T y$

b. $\langle x, y \rangle := x^T B y, \quad B = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$

3.5 考虑具有点积的欧几里得向量空间 $\mathbb{R}^5$ 。子空间 $U \subseteq \mathbb{R}^5$ 和 $x \in \mathbb{R}^5$ 由下式给出

$$U = \text{span}[[0; 2; -1; 2], [1; 1; 2; 2], [-3; 1; 0; 1], [-1; 5; 4; 7], [-1; -3; -9; 1]], \quad x = [-1; -1; 0; 1]$$

a. 确定 x 到 U 的正交投影 $\pi_U(x)$

b. 确定距离 $d(x, U)$

3.6 考虑 $\mathbb{R}^3$ ，其内积为

$$\langle x, y \rangle := x^T \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} y$$

此外，我们将 e_1, e_2, e_3 定义为 $\mathbb{R}^3$ 中的标准/规范基。

a. 确定 e_2 到 $U = \text{span}\{e_1, e_3\}$ 的正交投影 $\pi_U(e_2)$ 。提示：正交性通过内积定义。

b. 计算距离 $d(e_2, U)$ 。

c. 画出场景：标准基向量和 $\pi_U(e_2)$

3.7 设 V 是向量空间， π 是 V 的自同态映射。

a. 证明 π 是投影当且仅当 $\text{id}_V - \pi$ 是投影，其中 id_V 是 V 上的恒等自同态映射。

b. 现在假设 π 是投影。计算 $\text{Im}(\text{id}_V - \pi)$ 和 $\ker(\text{id}_V - \pi)$ 作为 $\text{Im}(\pi)$ 和 $\ker(\pi)$ 的函数。

3.8 使用 Gram-Schmidt 方法，将三维子空间 $U \subseteq \mathbb{R}^3$ 的基 $B = (b_1, b_2)$ 转换为 U 的 ONB $C = (c_1, c_2)$ ，其中

$$b_1 := [1; 1; 1], \quad b_2 := [-1; 2; 0]$$

3.9 设 $n \in \mathbb{N}$ ，设 $x_1, \dots, x_n > 0$ 是 n 个正实数，使得 $x_1 + \dots + x_n = 1$ 。使用 Cauchy-Schwarz 不等式并证明

a. $\sum_{i=1}^n 1/x_i \geq n^2$

b. $\sum_{i=1}^n x_i^2 \geq 1/n$

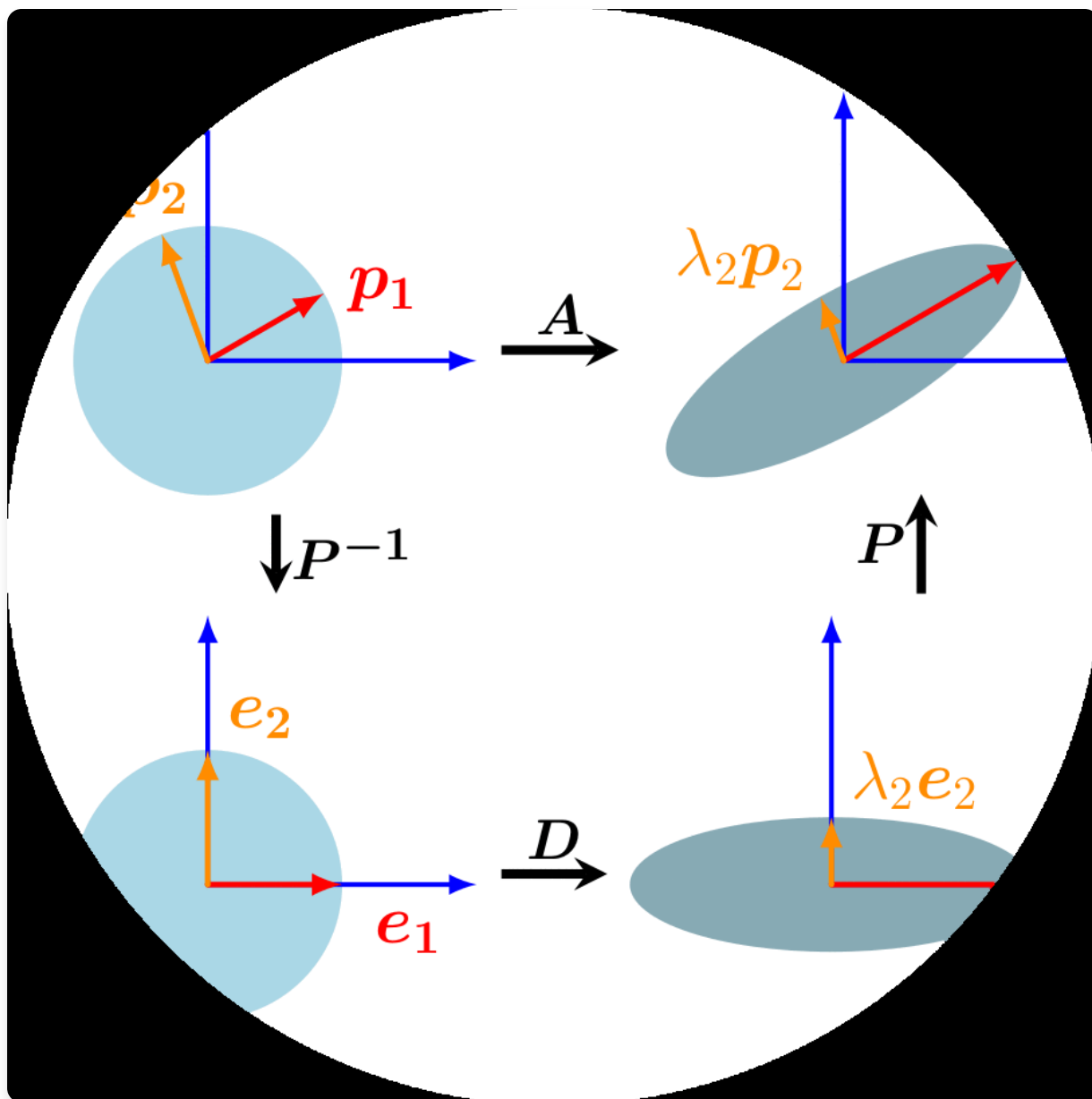
提示：考虑 $\mathbb{R}^n$ 上的点积。然后选择特定的向量 $x, y \in \mathbb{R}^n$ 并应用 Cauchy-Schwarz 不等式。

3.10 将向量旋转 30°

$$x_1 := [2; 3], \quad x_2 := [0; -1]$$

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

4 Matrix分解



在第2章和第3章中，我们研究了操作和测量向量、向量投影和线性映射的方法。向量的映射和变换可以方便地描述为由矩阵执行的操作。此外，数据通常也以矩阵形式表示，例如，矩阵的行表示不同的人，列

描述人们不同特征，例如体重、身高和社会经济地位。在本章中，我们将介绍矩阵的三个方面：如何总结矩阵、矩阵如何分解，以及如何使用这些分解进行矩阵近似。

我们首先考虑能够用少数几个数字来描述矩阵整体性质特征的方法。我们将在行列式(Section [4.1])和特征值(Section [4.2])章节中针对方阵这一重要特殊情况进行讨论。这些特征数具有重要的数学意义，使我们能够快速把握矩阵的有

用性质。接下来我们将讨论矩阵分解方法：矩阵分解的类比是数字的因式分解，例如将21分解为质数 $7 \cdot 3$ 。因此矩阵分解也常被称为*矩阵因式分解*。矩阵分解用于通过可解释矩阵的因子的不同表示来描述矩阵。

我们首先介绍对称正定矩阵的类似平方根运算，即Cholesky分解(Section [4.3])。然后我们将研究两种将矩阵因式分解为标准形式的相关方法。第一种称为矩阵对角化(Section [4.4])，它使我们能够在选择适当基的情况下使用对角变换矩阵来表示线性映射。第二种方法是奇异值分解(Section [4.5])，它将这种因式分解扩展到非方阵，被认为是线性代数的基本概念之一。这些分解很有用，因为表示数值数据的矩阵通常非常大且难以分析。我们通过系统地概述矩阵类型及其特征性质，以矩阵分类法的形式结束本章(Section [4.7])。

本章介绍的方法在后续的数学章节（如第6章）以及应用章节（如第10章的降维或第11章的密度估计）中都将变得重要。本章的整体结构如图4.1的思维导图所示。

4.1 行列式和迹

行列式是线性代数中的重要概念。行列式是分析和求解线性方程组的数学对象。行列式仅对方阵 $A \in \mathbb{R}^{(n \times n)}$ 定义，即行数和列数相同的矩阵。在本书中，我们将行列式写作 $\det(A)$ 或有时写作 $|A|$ ，使得

$$\det(A) = |a_{11} \ a_{12} \ \dots \ a_{1n}| \ |a_{21} \ a_{22} \ \dots \ a_{2n}| \ \dots \ \dots \ \dots \ |a_{n1} \ a_{n2} \ \dots \ a_{nn}| \quad (4.1)$$

$n \times n$ 方阵 $A \in \mathbb{R}^{n \times n}$ 的行列式是一个将 A 映射到实数的函数。在为一般 $n \times n$ 矩阵提供行列式定义之前，让我们先看一些激发性的例子，并为一些特殊矩阵定义行列式。

例4.1 (矩阵可逆性测试)

让我们从探索方阵 A 是否可逆开始(见Section [2.2.2])。对于最简单的情况，我们已经知道矩阵何时可逆。如果 A 是 1×1 矩阵，即它是一个标量，那么 $A^{-1} = 1/a \Rightarrow A = 1$ 。因此 $a = 1$ 成立，当且仅当 $a \neq 0$ 。

对于 2×2 矩阵，根据逆矩阵的定义(Definition [2.3])，我们知道 $AA^{-1} = I$ 。然后，通过(2.24)， A 的逆矩阵是

$$A^{-1} = 1/(a_{11}a_{22} - a_{12}a_{21}) \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix} \quad (4.2)$$

因此， A 可逆当且仅当

$$a_{11}a_{22} - a_{12}a_{21} \neq 0 \quad (4.3)$$

这个量就是 2×2 矩阵 $A \in \mathbb{R}^{2 \times 2}$ 的行列式，即

$$\det(|a_{11} \ a_{12}|) = a_{11}a_{22} - a_{12}a_{21} \quad (4.4)$$

例4.1已经指出了行列式与逆矩阵存在性之间的关系。下面的定理对 $n \times n$ 矩阵陈述了相同的结果。

定理4.1. 对于任何方阵 $A \in \mathbb{R}^{n \times n}$ ， A 可逆当且仅当 $\det(A) \neq 0$ 。

我们有用矩阵元素表示的小矩阵行列式的显式（闭式）表达式。对于 $n = 1$ ，

$$\det(A) = \det(a_{11}) = a_{11} \quad (4.5)$$

对于 $n = 2$ ，

$$\det(|a_{11} \ a_{12}|) = a_{11}a_{22} - a_{12}a_{21} \quad (4.6)$$

这在前面的例子中我们已经观察到。

对于 $n = 3$ （称为Sarrus法则），

$$|a_{11} \ a_{12} \ a_{13}| \ |a_{21} \ a_{22} \ a_{23}| \ |a_{31} \ a_{32} \ a_{33}|$$

$$aaa = aaa + aaa + aaa \quad (4.7) \quad [21] \ [22] \ [23] \ [11] \ [22] \ [33] \ [21] \ [32] \ [13] \ [31] \ [12] \ [23] \ aaa$$

[31] [32] [33]

$$-a[31]a[22]a[13] - a[11]a[32]a[23] - a[21]a[12]a[33] .$$

作为Sarrus规则中乘积项的记忆辅助，可以尝试在矩阵中追踪三重乘积的元素。

我们称方阵 T 为**上三角矩阵**，如果对于 $i > j$ ，有 $T[ij] = 0$ ，即矩阵在对角线以下为零。类似地，我们定义**下三角矩阵**为对角线以上为零的矩阵。对于 $n \times n$ 三角矩阵 $T \in \mathbb{R}$ ，行列式是对角元素的乘积，即

$$\det(T) = \prod_{i=1}^n T[ii] . \quad (4.8)$$

例4.2 (行列式作为体积的度量)

当我们将行列式视为从 n 个向量张成 $\mathbb{R}^n$ 中对象的集合的映射时，行列式的概念是自然的。事实证明，行列式 $\det(A)$ 是由矩阵 A 的列向量形成的 n 维平行六面体的有向体积。

对于 $n = 2$ ，矩阵的列形成一个平行四边形；见图4.2。当向量之间的夹角变小时，平行四边形的面积也会缩小。考虑两个向量 b 、 g 构成矩阵 $A = [b, g]$ 的列。那么， A 的行列式的绝对值就是顶点为 0 、 b 、 g 、 $b + g$ 的平行四边形的面积。特别地，如果 b 、 g 线性相关，使得对某个 $\lambda \in \mathbb{R}$ 有 $b = \lambda g$ ，它们不再形成二维平行四边形。因此，相应的面积为 0 。相反，如果 b 、 g 线性无关且是标准基向量 e_1 、 e_2 的倍数，那么它们可以写成 $b = [b_1, 0]^T$ ， $g = [0, g_2]^T$ ，行列式为 $b_1 g_2 - 0 = b_1 g_2$ 。

行列式的符号表示张成向量 b 、 g 相对于标准基 (e_1, e_2) 的方向。在我们的图中，将顺序翻转为 g 、 b 会交换 A 的列并逆转阴影区域的方向。这成为熟悉的公式：面积 = 高度 $\times$ 长度。这种直觉扩展到更高维度。在 $\mathbb{R}^3$ 中，我们考虑三个向量 r 、 b 、 $g \in \mathbb{R}^3$ 张成平行六面体的边，即具有平行四边形面的实体（见图4.3）。 3×3 矩阵 $[r, b, g]$ 的行列式的绝对值是该实体的体积。因此，行列式作为一个函数，测量由矩阵中组成的列向量形成的有向体积。

考虑三个线性无关的向量 r 、 g 、 $b \in \mathbb{R}^3$ ，给定为

$$r = [2, 0, -8]^T, \quad g = [6, 1, 0]^T, \quad b = [1, 4, -1]^T . \quad (4.9)$$

将这些向量写成矩阵的列

$$A = [r, g, b] = \begin{bmatrix} 2 & 6 & 1 \\ 0 & 1 & 4 \\ -8 & 0 & -1 \end{bmatrix} \quad (4.10)$$

允许我们计算所需的体积为

$$V = |\det(A)| = 186 . \quad (4.11)$$

计算 $n \times n$ 矩阵的行列式需要一个通用算法来解决 $n > 3$ 的情况，我们将在下面探讨。下面的定理4.2将计算 $n \times n$ 矩阵行列式的问题简化为计算 $(n-1) \times (n-1)$ 矩阵的行列式。通过递归应用Laplace展开（定理4.2），我们因此可以通过最终计算 2×2 矩阵的行列式来计算 $n \times n$ 矩阵的行列式。

定理4.2 (Laplace展开)。考虑矩阵 $A \in \mathbb{R}^{(n \times n)}$ 。那么，对于所有 $j = 1, \dots, n$:

1. 沿列 j 展开

$$\det(A) = \sum_{k=1}^n (-1)^{(k+j)} a[kj] \det(A[k,j]) . \quad (4.12)$$

2. 沿行 j 展开

$$\det(A) = \sum_{k=1}^n (-1)^{k+j} a_{jk} \det(A_{j,k}). \quad (4.13)$$

这里 $A_{j,k} \in \mathbb{R}^{(n-1) \times (n-1)}$ 是我们删除第 k 行和第 j 列后从 A 得到的子矩阵。

例4.3 (Laplace展开)

让我们使用沿第一行的Laplace展开来计算行列式

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

应用(4.13)得出

$$\det(A) = (-1)^{(1+1)} \cdot 1 \cdot \det(\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}) + (-1)^{(1+2)} \cdot 2 \cdot \det(\begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix}) + (-1)^{(1+3)} \cdot 3 \cdot \det(\begin{bmatrix} 3 & 1 \\ 0 & 0 \end{bmatrix}) \quad (4.15)$$

我们使用(4.6)计算所有 2×2 矩阵的行列式，得到

$$\det(A) = 1(1 - 0) - 2(3 - 0) + 3(0 - 0) = -5. \quad (4.16)$$

为了完整性，我们可以将此结果与使用Sarrus规则(4.7)计算行列式进行比较：

$$\det(A) = 1 \cdot 1 \cdot 1 + 3 \cdot 0 \cdot 3 + 0 \cdot 2 \cdot 2 - 0 \cdot 1 \cdot 3 - 1 \cdot 0 \cdot 2 - 3 \cdot 2 \cdot 1 = 1 - 6 = -5. \quad (4.17)$$

对于 $A \in \mathbb{R}^{(n \times n)}$ ，行列式具有以下性质：

矩阵乘积的行列式是相应行列式的乘积， $\det(AB) = \det(A)\det(B)$ 。

行列式对转置不变，即 $\det(A^T) = \det(A)$ 。

如果 A 是正则的（可逆的），那么 $\det(A^{-1}) = 1/\det(A)$ 。

相似矩阵（定义[2.22]）具有相同的行列式。因此，对于线性映射 $\Phi: V \rightarrow V$ ， Φ 的所有变换矩阵 $A[\Phi]$ 都具有相同的行列式。因此，行列式对于线性映射的基选择是不变的。

将一行/列的倍数加到另一行/列上不会改变 $\det(A)$ 。

用 $\lambda \in \mathbb{R}$ 乘一行/列会将 $\det(A)$ 缩放 λ 倍。特别地， $\det(\lambda A) = \lambda^n \det(A)$ 。

交换两行/列会改变 $\det(A)$ 的符号。

由于这后三个性质，我们可以使用高斯消元法（见第[2.1]节）通过将 A 化为行阶梯形式来计算 $\det(A)$ 。当我们将 A 化为三角形（对角线下方的元素全为0）时，就可以停止高斯消元法。回忆公式(4.8)，三角矩阵的行列式等于对角元素的乘积。

定理4.3. 方阵 $A \in \mathbb{R}^{n \times n}$ 满足 $\det(A) \neq 0$ 当且仅当 $\text{rk}(A) = n$ 。换句话说， A 可逆当且仅当它是满秩的。

当数学主要靠手工计算时，行列式计算被认为是分析矩阵可逆性的重要方法。然而，机器学习中的现代方法使用直接数值方法，这些方法已经取代了行列式的显式计算。例如，在第[2]章中，我们学到逆矩阵可以通过高斯消元法计算。因此高斯消元法也可以用来计算矩阵的行列式。

行列式在接下来的章节中将发挥重要的理论作用，特别是当我们通过特征多项式学习特征值和特征向量（第[4.2]节）时。

定义4.4. 方阵 $A \in \mathbb{R}^{n \times n}$ 的迹定义为

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

矩阵分解

$$\text{tr}(A) := \sum_{i=1}^n a_{ii}, \quad (4.18)$$

即，迹是 A 的对角元素之和。

迹满足以下性质：

$$\begin{aligned} \text{tr}(A + B) &= \text{tr}(A) + \text{tr}(B) \text{ 对于 } A, B \in \mathbb{R}^{n \times n} \\ \text{tr}(\alpha A) &= \alpha \text{tr}(A), \alpha \in \mathbb{R} \text{ 对于 } A \in \mathbb{R}^{n \times n} \\ \text{tr}(I_n) &= n \\ \text{tr}(AB) &= \text{tr}(BA) \text{ 对于 } A \in \mathbb{R}^{n \times k}, B \in \mathbb{R}^{k \times n} \end{aligned}$$

可以证明只有一个函数同时满足这四个性质——迹[Gohberg et al., 2012]。

矩阵乘积的迹的性质更加一般。具体地，迹在循环置换下不变，即

$$\text{tr}(AKL) = \text{tr}(KLA) \quad (4.19)$$

对于矩阵 $A \in \mathbb{R}^{n \times k}$, $K \in \mathbb{R}^{k \times l}$, $L \in \mathbb{R}^{l \times n}$ 。这个性质推广到任意个矩阵的乘积。作为(4.19)的特例，对于两个向量 $x, y \in \mathbb{R}^n$ 有

$$\text{tr}(xy^T) = \text{tr}(y^T x) = y^T x \in \mathbb{R}. \quad (4.20)$$

给定线性映射 $\Phi: V \rightarrow V$ ，其中 V 是向量空间，我们通过使用 Φ 的矩阵表示的迹来定义这个映射的迹。对于 V 的给定基，我们可以用变换矩阵 A 来描述 Φ 。那么 Φ 的迹就是 A 的迹。对于 V 的不同基， Φ 对应的变换矩阵 B 可以通过形如 $S^{-1}AS$ 的基变换得到，其中 S 合适（见第[2.7.2]节）。对于 Φ 对应的迹，这意味着

$$\text{tr}(B) = \text{tr}(S^{-1}AS) = \text{tr}(ASS^{-1}) = \text{tr}(A). \quad (4.21)$$

因此，虽然线性映射的矩阵表示依赖于基，但线性映射 Φ 的迹与基无关。

在本节中，我们涵盖了行列式和迹作为刻画方阵的函数。结合我们对行列式和迹的理解，我们现在可以定义一个重要的方程，用多项式来描述矩阵 A ，我们将在接下来的章节中广泛使用它。

定义4.5（特征多项式）。对于 $\lambda \in \mathbb{R}$ 和方阵 $A \in \mathbb{R}^{n \times n}$

$$p_A(\lambda) := \det(A - \lambda I) \quad (4.22a) = c_0 + c_1\lambda + c_2\lambda^2 + \cdots + c_{n-1}\lambda^{n-1} + (-1)^n\lambda^n, \quad (4.22b)$$

其中 $c_0, \dots, c_{n-1} \in \mathbb{R}$, 这是 A 的特征多项式。特别地,

$$c_0 = \det(A), \quad c_{n-1} = (-1)^{n-1} \operatorname{tr}(A). \quad (4.24)$$

特征多项式(4.22a)将允许我们计算特征值和特征向量, 这将在下一节中讨论。

4.2 特征值和特征向量

我们现在将了解刻画矩阵及其相关线性映射的新方法。回忆第[2.7.1]节，给定有序基，每个线性映射都有唯一的变换矩阵。我们可以通过执行“eigen”分析来解释线性映射及其相关的变换矩阵。正如我们将看到的，线性映射的特征值将告诉我们一组特殊向量——特征向量，是如何被线性映射变换的。

定义 4.6. 设 $A \in \mathbb{R}^{n \times n}$ 是方阵。那么 $\lambda \in \mathbb{R}$ 是 A 的特征值， $x \in \mathbb{R}^n \setminus \{0\}$ 是 A 对应的特征向量，如果

$$Ax = \lambda x. \quad (4.25)$$

我们称(4.25)为特征值方程。

备注。在线性代数文献和软件中，通常会按降序排列特征值，使得最大特征值和关联的特征向量被称为第一特征值及其关联的特征向量，第二大的被称为第二特征值及其关联的特征向量，以此类推。然而，教科书和出版物可能有不同的或没有排序概念。如果没有明确说明，我们不想在本书中假设排序。◇

以下陈述是等价的：

λ 是 $A \in \mathbb{R}^{(n \times n)}$ 的特征值。

存在 $x \in \mathbb{R}^n \setminus \{0\}$ 使得 $Ax = \lambda x$ ，或等价地， $(A - \lambda I_n)x = 0$ 可以非平凡求解，即 $x \neq 0$ 。

$\text{rk}(A - \lambda I_n) < n$ 。

$\det(A - \lambda I_n) = 0$ 。

定义 4.7（共线性和同向性）。指向同一方向的两个向量称为*同向的*。如果两个向量指向相同或相反方向，则它们是*共线的*。

备注（特征向量的非唯一性）。如果 x 是与特征值 λ 关联的 A 的特征向量，那么对于任何 $c \in \mathbb{R} \setminus \{0\}$ ，都有 cx 是具有相同特征值的 A 的特征向量，因为

$$A(cx) = cAx = c\lambda x = \lambda(cx). \quad (4.26)$$

因此，所有与 x 共线的向量也是 A 的特征向量。◇

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[106] [Matrix Decompositions]

定理 4.8。 $\lambda \in \mathbb{R}$ 是 $A \in \mathbb{R}^{(n \times n)}$ 的特征值当且仅当 λ 是 A 的特征多项式 $p_A(\lambda)$ 的根。

定义 4.9。设方阵 A 有特征值 λ_i 。 λ_i 的代数重数是该根在特征多项式中出现的次数。

定义 4.10（特征空间和特征谱）。对于 $A \in \mathbb{R}^{(n \times n)}$ ，与特征值 λ 关联的所有特征向量的集合张成 $\mathbb{R}^n$ 的一个子空间，这个子空间称为 A 关于 λ 的*特征空间*，记为 E_λ 。 A 的所有特征值的集合称为*特征谱*，或简称*谱*。

如果 λ 是 $A \in \mathbb{R}^{(n \times n)}$ 的特征值，那么相应的特征空间 E_λ 是齐次线性方程组 $(A - \lambda I)x = 0$ 的解空间。几何上，对应于非零特征值的特征向量指向被线性映射拉伸的方向。特征值是拉伸的倍数。如果特征值为负，拉伸的方向会翻转。

例 4.4（单位矩阵的情况）

单位矩阵 $I \in \mathbb{R}^{(n \times n)}$ 的特征多项式为 $p_I(\lambda) = \det(I_n - \lambda I_n) = (1 - \lambda)^n = 0$ ，只有一个特征值 $\lambda = 1$ ，出现 n 次。此外，对于所有向量 $x \in \mathbb{R}^n \setminus \{0\}$ ，都有 $Ix = \lambda x = 1x$ 。因此，单位矩阵的唯一特征空间 E_1 张成 n 维， $\mathbb{R}^n$ 的所有 n 个标准基向量都是 I 的特征向量。

关于特征值和特征向量的有用性质包括：

矩阵 A 及其转置 A^T 具有相同的特征值，但不一定有相同的特征向量。

特征空间 E_λ 是 $A - \lambda I$ 的零空间，因为

$$Ax = \lambda x \iff Ax - \lambda x = 0 \text{ (4.27a)} \iff (A - \lambda I)x = 0 \iff x \in \ker(A - \lambda I). \text{ (4.27b)}$$

相似矩阵（见定义 2.22）具有相同的特征值。因此，线性映射 Φ 的特征值独立于其变换矩阵的基的选择。这使得特征值与行列式和迹一起成为线性映射的关键特征参数，因为它们在基变换下都是不变的。对称正定矩阵总是有正的实特征值。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>. 4.2 Eigenvalues and Eigenvectors] [107]

例 4.5（计算特征值、特征向量和特征空间）

让我们找到 2×2 矩阵的特征值和特征向量

$$A = \begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix}. \text{ (4.28)}$$

步骤 1：特征多项式。 根据我们对 A 的特征向量 $x \neq 0$ 和特征值 λ 的定义，存在一个向量使得 $Ax = \lambda x$ ，即 $(A - \lambda I)x = 0$ 。由于 $x \neq 0$ ，这要求 $A - \lambda I$ 的 kernel（零空间）包含除 0 之外的更多元素。这意味着 $A - \lambda I$ 不可逆，因此 $\det(A - \lambda I) = 0$ 。因此，我们需要计算特征多项式 (4.22a) 的根来找到特征值。

步骤 2：特征值。 特征多项式为

$$p_A(\lambda) = \det(A - \lambda I) \text{ (4.29a)} = \det\begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \det\begin{bmatrix} 4-\lambda & 2 \\ 1 & 3-\lambda \end{bmatrix} \text{ (4.29b)} = \begin{vmatrix} 4-\lambda & 2 \\ 1 & 3-\lambda \end{vmatrix} = (4-\lambda)(3-\lambda) - 2 \cdot 1. \text{ (4.29c)}$$

我们分解特征多项式，得到

$$p_A(\lambda) = (4-\lambda)(3-\lambda) - 2 \cdot 1 = 10 - 7\lambda + \lambda^2 = (\lambda-2)(\lambda-5) \text{ (4.30)}$$

给出根 $\lambda_1 = 2$ 和 $\lambda_2 = 5$ 。

步骤 3：特征向量和特征空间。 我们通过寻找向量 x 使得

$$\begin{bmatrix} 4-\lambda & 2 \\ 1 & 3-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ (4.31)}$$

对于 $\lambda = 5$ ，我们得到

$$\begin{bmatrix} 4 & -5 & 2 \end{bmatrix} \begin{bmatrix} x_1 \end{bmatrix} = \begin{bmatrix} -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \begin{bmatrix} 1 & 3 & -5 \end{bmatrix} \begin{bmatrix} x_2 \end{bmatrix} \begin{bmatrix} 1 & -2 \end{bmatrix} \begin{bmatrix} x_2 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix}. \quad (4.32)$$

我们求解这个齐次系统，得到解空间

$$E_5 = \text{span} \left\{ \begin{bmatrix} 2 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} \right\}. \quad (4.33)$$

这个特征空间是一维的，因为它只有一个基向量。

类似地，我们通过求解齐次方程组来找到 $\lambda = 2$ 的特征向量

$$\begin{bmatrix} 4 & -2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \end{bmatrix} = \begin{bmatrix} 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \begin{bmatrix} 1 & 3 & -2 \end{bmatrix} \begin{bmatrix} x_2 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_2 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix}. \quad (4.34)$$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[108] *[Matrix Decompositions]*

这意味着任何向量 $x = \begin{bmatrix} x_1 \end{bmatrix}$ ，其中 $x_2 = -x_1$ ，例如 $\begin{bmatrix} 1 \end{bmatrix}$ ，都是特征值为 2 的特征向量。 $\begin{bmatrix} x_2 \end{bmatrix} \begin{bmatrix} -1 \end{bmatrix}$ 相应的特征空间为

$$E_2 = \text{span} \left\{ \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} -1 \end{bmatrix} \right\}. \quad (4.35)$$

在示例 4.5 中，两个特征空间 $E[5]$ 和 $E[2]$ 都是一维的，因为它们都由单个向量张成。然而，在其他情况下，我们可能有多个相同的特征值(见定义 4.9)，特征空间可能具有多于一个维度。

定义 4.11. 设 $\lambda[i]$ 是方阵 A 的特征值。那么 $\lambda[i]$ 的几何重数是与 $\lambda[i]$ 相关联的线性无关特征向量的个数。换句话说，它是由与 $\lambda[i]$ 相关联的特征向量张成的特征空间的维数。

注释. 特定特征值的几何重数必须至少是 1，因为每个特征值至少有一个相关联的特征向量。特征值的几何重数不能超过其代数重数，但可能更低。◇

示例 4.6

矩阵 $A = \begin{bmatrix} 2 & 1 \end{bmatrix}$ 有两个重复的特征值 $\lambda_1 = \lambda_2 = 2$ ，代数重数为 2。然而，

该特征值只有一个不同的单位特征向量 $x_1 = \begin{bmatrix} 1 \end{bmatrix}$ ，因此几何重数为 1。

二维中的图形直觉

让我们通过不同的线性映射来获得对行列式、特征向量和特征值的一些直觉。图4.4描绘了五个变换矩阵 $A_1, \dots, A_5$ 及其对以原点为中心的正方形网格点的影响：

$A_1 = 2 \ 0$ 。两个特征向量的方向对应于 $\mathbb{R}^2$ 中的标准基向量，即两个基 $0 \ 1/2$

本坐标轴。垂直轴被扩展了2倍(特征值 $\lambda_1 = 2$)，水平轴被压缩了1/2倍(特征值 $\lambda_2 = 1/2$)。该映射是面积保持的($\det(A_1) = 1 = 2 \cdot 1/2$)。

$A_2 = 1 \ 1$ 对应于剪切映射，即它将点沿水平轴向右剪切(如果它们在正 $0 \ 1$

图4.4 行列式与特征空间。

五个线性映射及其相关变换矩阵 $A[i] \in \mathbb{R}^{2 \times 2}$ 的概述，将400个颜色编码的点 $x \in \mathbb{R}^2$ (左列)投影到目标点 $A[i]x$ (右列)。中央列描绘了第一个特征向量被其相关特征值 λ_1 拉伸，以及第二个特征向量被其特征值 λ_2 拉伸。每一行描绘了五个变换矩阵 $A[i]$ 中的一个相对于标准基的效果。

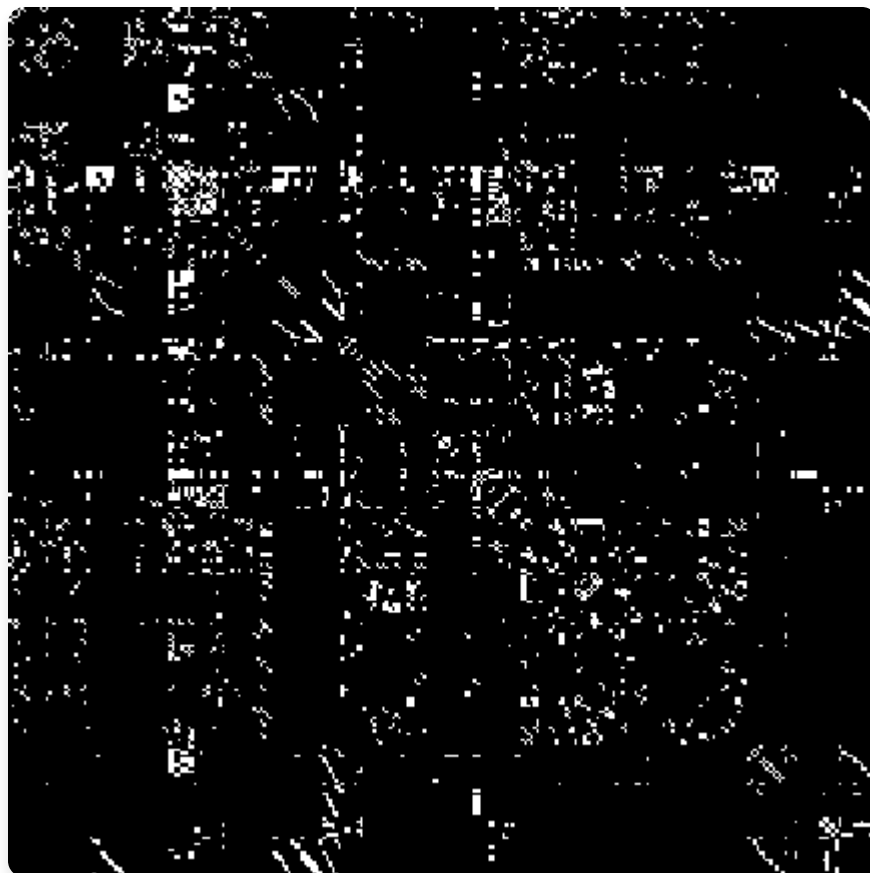


图4.4：行列式与特征空间概述

垂直轴的一半), 反之向左剪切。该映射是面积保持的($\det(A_2) = 1$)。特征值 $\lambda_1 = 1 = \lambda_2$ 是重复的, 特征向量是共线的(这里为了强调在两个相反方向上绘制)。这表明映射只沿一个方向(水平轴)作用。

矩阵 $A_3 = \sqrt{3}/2 \ -1/2 = \cos(\pi/6) \ -\sin(\pi/6)$ 将点逆时针旋转 $\pi/6$ 弧度 $1/2 \ \sqrt{3}/2 \ \sin(\pi/6) \ \cos(\pi/6)$

$= 30^\circ$, 并且只有复特征值, 反映该映射是旋转(因此, 没有绘制特征向量)。旋转必须是体积保持的, 所以行列式为1。有关旋转的更多细节, 我们参考第3.9节。

$A_4 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ 表示标准基中的映射, 将二维域坍塌到一维。由于一个特征 -1

值为0, 沿对应于 $\lambda_1 = 0$ 的(蓝色)特征向量方向的空间坍塌, 而正交的(红色)特征向量将空间拉伸 $\lambda_2 = 2$ 倍。因此, 像的面积为0。

$A_5 = \begin{pmatrix} 2/3 & 1/3 \\ 1/3 & 1/2 \end{pmatrix}$ 是剪切拉伸映射, 将空间缩放75%, 因为 $|\det(A_5)| = 3/4$ 。 $1/3 \ 1/2$

它沿 λ_2 的(红色)特征向量将空间拉伸1.5倍, 沿正交的(蓝色)特征向量将其压缩0.5倍。

示例4.7(生物神经网络的特征谱)

图4.5 秀丽隐杆线虫神经网络

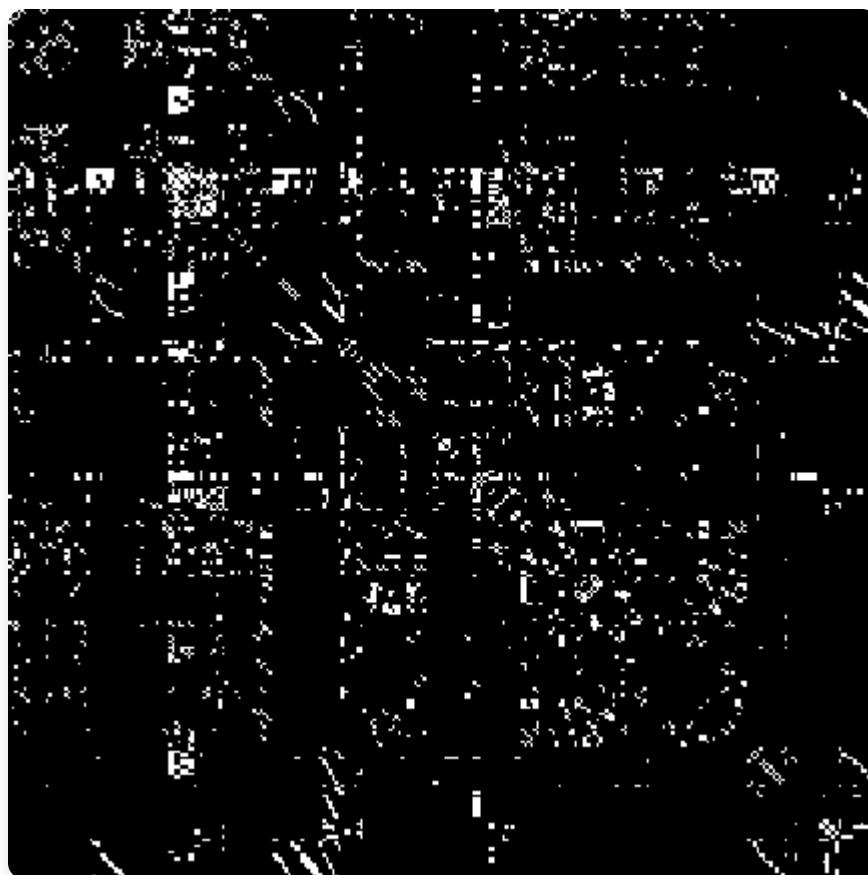


图4.5: 秀丽隐杆线虫神经网络

a. 对称化连接矩阵; (b) 特征谱。

分析和从网络数据中学习的方法是机器学习方法的重要组成部分。理解网络的关键是网络节点之间的连接性，特别是两个节点是否相互连接。在数据科学应用中，研究捕获这种连接数据的矩阵通常是有用的。

我们构建了秀丽隐杆线虫完整神经网络的连接/邻接矩阵 $A \in \mathbb{R}^{277 \times 277}$ 。每行/列代表这个蠕虫大脑的277个神经元中的一个。如果神经元*i*通过突触与神经元*j*交流，连接矩阵*A*的值为 $a[ij] = 1$ ，

$a[ij] = 0$ 否则。连接矩阵不是对称的，这意味着特征值可能不是实数。因此，我们计算连接矩阵的对称化版本为 $A[\text{sym}] := A + A^T$ 。这个新矩阵 $A[\text{sym}]$ 如图4.5(a)所示，当且仅当两个神经元连接时（白色像素），不论连接方向如何，都有非零值 $a[ij][\text{sym}]$ 。在图4.5(b)中，我们展示了 $A[\text{sym}]$ 对应的特征谱。横轴显示特征值的索引，按降序排列。纵轴显示对应的特征值。这种特征谱的S型形状在许多生物神经网络中都很典型。导致这种现象的潜在机制是神经科学研究的活跃领域。

[《机器学习数学》草稿 (2023-10-18)。反馈: <https://mml-book.com>。4.2 特征值和特征向量 111]

定理 4.12

矩阵 $A \in \mathbb{R}^{(n \times n)}$ 的特征向量 $x_1, \dots, x_n$ 具有 n 个不同的特征值 $\lambda_1, \dots, \lambda_n$ ，它们是线性无关的。

这个定理说明具有 n 个不同特征值的矩阵的特征向量构成 $\mathbb{R}^n$ 的一个基。

定义 4.13

方矩阵 $A \in \mathbb{R}^{(n \times n)}$ 是缺陷的(defective), 如果它拥有少于 n 个线性无关的特征向量。

非缺陷矩阵 $A \in \mathbb{R}^{(n \times n)}$ 不一定需要 n 个不同的特征值, 但确实需要特征向量构成 $\mathbb{R}^n$ 的一个基。观察缺陷矩阵的特征空间, 可以得出特征空间维数之和小于 n 。具体地, 缺陷矩阵至少有一个特征值 λ_i , 其代数重数 $m > 1$, 几何重数小于 m 。

注记。 缺陷矩阵不能有 n 个不同的特征值, 因为不同的特征值有线性无关的特征向量 (定理4.12)。 $\diamond$

定理 4.14

给定矩阵 $A \in \mathbb{R}^{(m \times n)}$ ，我们总能通过定义 $S := A^T A$ 得到一个对称的、半正定矩阵 $S \in \mathbb{R}^{(n \times n)}$ 。(4.36)

注记。 如果 $\text{rk}(A) = n$ ，那么 $S := A^T A$ 是对称的、正定的。 $\diamond$

理解定理4.14成立的原因有助于我们了解如何使用对称化矩阵：对称性要求 $S = S^T$ ，通过代入(4.36)我们得到 $S^T = (A^T A)^T = A^T (A^T)^T = A^T A = S$ 。此外，半正定性（第3.2.3节）要求 $x^T S x \geq 0$ ，代入(4.36)我们得到 $x^T S x = x^T A^T A x = (x^T A^T)(A x) = (A x)^T (A x) \geq 0$ ，因为点积计算的是平方和（本身非负）。

定理 4.15（谱定理）

如果 $A \in \mathbb{R}^{(n \times n)}$ 是对称的，那么存在对应向量空间 V 的一个标准正交基，该基由 A 的特征向量组成，且每个特征值都是实数。

谱定理的直接含义是对称矩阵 A 的特征分解存在（具有实特征值），并且我们可以找到特征向量的标准正交基 (ONB)，使得 $A = PDP^T$ ，其中 D 是对角矩阵， P 的列包含特征向量。

例 4.8

考虑矩阵

$$A = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{bmatrix} \quad (4.37)$$

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由 Cambridge University Press 出版 (2020)。

112 矩阵分解

A 的特征多项式是

$$p_A(\lambda) = -(\lambda-1)^2(\lambda-7), \quad (4.38)$$

因此我们得到特征值 $\lambda_1 = 1$ 和 $\lambda_2 = 7$, 其中 λ_1 是重根特征值。按照计算特征向量的标准程序, 我们得到特征空间

$$E_1 = \text{span}[\begin{bmatrix} -1 \\ -1 \end{bmatrix}], E_7 = \text{span}[\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}] \quad (4.39)$$
$$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} =: \mathbf{x}_1, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} =: \mathbf{x}_2, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} =: \mathbf{x}_3$$

我们看到 $\mathbf{x}_3$ 与 $\mathbf{x}_1$ 和 $\mathbf{x}_2$ 都正交。然而, 由于 $\mathbf{x}_1^T \mathbf{x}_2 = 1 \neq 0$, 它们不正交。谱定理 (定理4.15) 说明存在一个正交基, 但我们得到的不是正交的。不过, 我们可以构造一个。

为了构造这样的基, 我们利用 $\mathbf{x}_1, \mathbf{x}_2$ 是与同一特征值 λ 关联的特征向量这一事实。因此, 对于任何 $\alpha, \beta \in \mathbb{R}$ 都有

$$A(\alpha \mathbf{x}_1 + \beta \mathbf{x}_2) = A\mathbf{x}_1\alpha + A\mathbf{x}_2\beta = \lambda(\alpha \mathbf{x}_1 + \beta \mathbf{x}_2), \quad (4.40)$$

即, $\mathbf{x}_1$ 和 $\mathbf{x}_2$ 的任何线性组合也是 A 与 λ 关联的特征向量。Gram-Schmidt 算法 (第3.8.3节) 是一种使用这种线性组合从一组基向量迭代构造正交/标准正交基的方法。因此, 即使 $\mathbf{x}_1$ 和 $\mathbf{x}_2$ 不正交, 我们也可以应用 Gram-Schmidt 算法找到与 $\lambda_1 = 1$ 关联的特征向量, 它们彼此正交 (也与 $\mathbf{x}_3$ 正交)。在我们的例子中, 我们将得到

$$\mathbf{x}_1' = [1/\sqrt{2}], \mathbf{x}_2' = [-1/\sqrt{6}] \quad (4.41)$$
$$\begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{6} \\ 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{6} \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2/\sqrt{6} \end{bmatrix}$$

它们彼此正交, 与 $\mathbf{x}_3$ 正交, 且是 A 与 $\lambda_1 = 1$ 关联的特征向量。

在结束对特征值和特征向量的考虑之前, 将这些矩阵特征与行列式和迹的概念联系起来是有用的。

定理 4.16

矩阵 $A \in \mathbb{R}^{(n \times n)}$ 的行列式是其特征值的乘积，即

$$\det(A) = \prod_{i=1}^n \lambda_i, \quad (4.42)$$

其中 $\lambda_i \in \mathbb{C}$ 是 A 的（可能重复的）特征值。

[《机器学习数学》草稿 (2023-10-18)。反馈：<https://mml-book.com>。4.2 特征值和特征向量 113]

图 4.6

特征值的几何解释。 A 的特征向量 v_1 被对应的特征值 λ_1 拉伸， v_2 被对应的特征值 λ_2 拉伸。

定理 4.17. 矩阵 $A \in \mathbb{R}^{n \times n}$ 的迹是其特征值之和，即

$$\text{tr}(A) = \sum_{i=1}^n \lambda_i, \quad (4.43)$$

其中 $\lambda_i \in \mathbb{C}$ 是 A 的（可能重复的）特征值。

让我们提供这两个定理的几何直观。考虑一个矩阵 $A \in \mathbb{R}^{2 \times 2}$ ，它拥有两个线性无关的特征向量 x_1, x_2 。在这个例子中，我们假设 (x_1, x_2) 是 $\mathbb{R}^2$ 的 ONB(正交归一基)，因此它们是正交的，并且它们张成的正方形面积为 1；见图 4.6。

从第 4.1 节我们知道，行列式计算单位正方形在变换 A 下的面积变化。在这个例子中，我们可以显式计算面积变化：使用 A 映射特征向量给出向量 $v_1 = Ax_1 = \lambda_1 x_1$ 和 $v_2 = Ax_2 = \lambda_2 x_2$ ，即新向量 v_i 是特征向量 x_i 的缩放版本，缩放因子是相应的特征值 λ_i 。 v_1, v_2 仍然正交，它们张成的矩形面积是 $|\lambda_1 \lambda_2|$ 。

给定在我们的例子中 x_1, x_2 是正交归一的，我们可以直接计算单位正方形的周长为 $2(1 + 1)$ 。使用 A 映射特征向量创建一个矩形，其周长是 $2(|\lambda_1| + |\lambda_2|)$ 。因此，特征值绝对值的和告诉我们单位正方形的周长在变换矩阵 A 下如何变化。

例 4.9 (Google 的 PageRank 算法——网页作为特征向量) Google 使用对应于矩阵 A 最大特征值的特征向量来确定页面在搜索中的排名。PageRank 算法的想法由 Larry Page 和 Sergey Brin 于 1996 年在斯坦福大学开发，即任何网页的重要性都可以通过链接到它的页面的重要性来近似。为此，他们将所有网站写成一个巨大的有向图，显示哪个页面链接到哪个。PageRank 通过计算指向 a_i 的页面数量来计算网站 a_i 的权重（重要性） $x_i \geq 0$ 。此外，PageRank 考虑链接到 a_i 的网站的重要性。用户的导航行为然后通过该图的转移矩阵 A 建模，该矩阵告诉我们某人以什么（点击）概率最终到达不同的网站。矩阵 A 具有以下性质：对于网站的任何初始排名/重要性向量 x ，序列 $x, Ax, A^2x, \dots$ 收敛到向量 x^* 。该向量称为 PageRank，满足

$Ax^* = x^*$ ，即它是 A 的特征向量（对应特征值为 1）。在归一化 x 使得 $\|x^*\| = 1$ 后，我们可以将条目解释为概率。关于 PageRank 的更多细节和不同观点可以在原始技术报告 (Page et al., 1999) 中找到。

4.3 Cholesky分解

有许多方法可以分解我们在机器学习中经常遇到的特殊类型矩阵。在正实数中，我们有平方根运算，它给我们数字分解为相同组件，例如 $9 = 3 \cdot 3$ 。对于矩阵，我们需要小心在正数量上计算类似平方根的运算。对于对称正定矩阵（见第3.2.3节），我们可以从许多等价的平方根运算中选择。Cholesky分解/Cholesky因式分解在对称正定矩阵上提供等价的平方根运算，在实践中很有用。

定理 4.18 (Cholesky分解). 对称正定矩阵 A 可以因式分解为乘积 $A = LL^T$ ，其中 L 是具有正对角元素的下三角矩阵：

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & \cdots & l_{n1} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & l_{nn} \end{bmatrix} \quad (4.44)$$

L 称为 A 的Cholesky因子，且 L 是唯一的。

例 4.10 (Cholesky因式分解)

考虑对称正定矩阵 $A \in \mathbb{R}^{\{3 \times 3\}}$ 。我们感兴趣找到其Cholesky因式分解 $A = LL^T$ ，即

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix} \quad (4.45)$$

$$A = a_{21} \ a_{22} \ a_{32} = LL^T = l_{21} \ l_{22} \ 0 \cdot \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix} \quad (4.45)$$

展开右侧得到

$$\begin{bmatrix} l_{11}^2 & l_{21}l_{11} & l_{31}l_{11} \\ l_{21}l_{11} & l_{22}^2 & l_{32}l_{22} \\ l_{31}l_{11} & l_{32}l_{22} & l_{31}^2 + l_{32}^2 + l_{33}^2 \end{bmatrix} \quad (4.46)$$

$$A = l_{21}l_{11} \ l_{21}^2 + l_{22}^2 \ l_{31}l_{21} + l_{32}l_{22} \ (4.46) \ l_{31}l_{11} \ l_{31}l_{21} + l_{32}l_{22} \ l_{31}^2 + l_{32}^2 + l_{33}^2$$

比较(4.45)的左侧和(4.46)的右侧表明，对角元素 l_{ii} 有一个简单的模式：

$$l_{11} = \sqrt{a_{11}}, l_{22} = \sqrt{(a_{22} - l_{21}^2)}, l_{33} = \sqrt{(a_{33} - (l_{31}^2 + l_{32}^2))} \quad (4.47)$$

类似地，对于对角线以下的元素 (l_{ij} ，其中 $i > j$)，也有重复模式：

$$l_{21} = a_{21}/l_{11}, l_{31} = a_{31}/l_{11}, l_{32} = (a_{32} - l_{31}l_{21})/l_{22} \quad (4.48)$$

因此，我们构造了任何对称正定 3×3 矩阵的 Cholesky 分解。关键的认识是，给定矩阵 A 的分量 $a_{[ij]}$ 和先前计算的 $l_{[ij]}$ 值，我们可以反向计算 L 矩阵的分量 $l_{[ij]}$ 应该是什么。

Cholesky 分解是机器学习数值计算的重要工具。在这里，对称正定矩阵需要频繁操作，例如，多元 Gaussian 变量的协方差矩阵（见第 [6.5] 节）是对称正定的。这个协方差矩阵的 Cholesky 分解允许我们从 Gaussian 分布中生成样本。它还允许我们对随机变量进行线性变换，这在计算深度随机模型（如变分自编码器）中的梯度时被大量利用 [Jimenez Rezende et al., 2014;] [Kingma and Welling,] [2014]。Cholesky 分解还允许我们非常高效地计算行列式。给定 Cholesky 分解 $A = LL^T$ ，我们知道 $\det(A) = \det(L)\det(L^T) = \det(L)^2$ 。由于 L 是三角矩阵，行列式就是其对角元素的乘积，所以 $\det(A) = (\prod_i L_{ii})^2$ 。因此，许多数值软件包使用 Cholesky 分解来提高计算效率。

4.4 特征分解和对角化

对角矩阵是在所有非对角元素上值为零的矩阵，即它们的形式为

$$D = \begin{bmatrix} c_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & c_n \end{bmatrix} \quad (4.49)$$

它们允许快速计算行列式、幂次和逆矩阵。行列式是其对角元素的乘积，矩阵的 k 次幂 $D[k]$ 由每个对角元素的 k 次幂给出，如果所有对角元素都非零，则逆矩阵 $D[-1]$ 是其对角元素的倒数。

在本节中，我们将讨论如何将矩阵转换为对角形式。这是我们在第 2.7.2 节讨论的基变换和第 4.2 节的特征值的重要应用。

回忆两个矩阵 A 、 D 是相似的（定义 2.22），如果存在可逆矩阵 P ，使得 $D = PAP[-1]$ 。更具体地，我们将研究与对角矩阵 D 相似的矩阵 A ，其中 D 在对角线上包含 A 的特征值。

定义 4.19（可对角化）。矩阵 $A \in R[n \times n]$ 是可对角化的，如果它与对角矩阵相似，即如果存在可逆矩阵 $P \in R[n \times n]$ 使得 $D = PAP[-1]$ 。

在下面，我们将看到对角化矩阵 $A \in R[n \times n]$ 是表达同一个线性映射但在另一个基下的方法（见第 2.6.1 节），这个基将是由 A 的特征向量组成的基。

设 $A \in R[n \times n]$ ，设 $\lambda_1, \dots, \lambda[n]$ 为一组标量，设 $p_1, \dots, p[n]$ 为 $R[n]$ 中的一组向量。我们定义 $P := [p_1, \dots, p[n]]$ ，设 $D \in R[n \times n]$ 为对角元素为 $\lambda_1, \dots, \lambda[n]$ 的对角矩阵。那么我们可以证明

$$AP = PD \quad (4.50)$$

当且仅当 $\lambda_1, \dots, \lambda[n]$ 是 A 的特征值， $p_1, \dots, p[n]$ 是 A 的对应特征向量。

我们可以看到这个陈述成立，因为

$$AP = A[p_1, \dots, p[n]] = [Ap_1, \dots, Ap[n]], \quad (4.51)$$

$$PD = [p_1, \dots, p[n]] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \lambda[n] \end{bmatrix} = [\lambda_1 p_1, \dots, \lambda[n] p[n]]. \quad (4.52)$$

因此，(4.50) 意味着

$$Ap_1 = \lambda_1 p_1 \quad (4.53) \quad \vdots \quad Ap[n] = \lambda[n] p[n]. \quad (4.54)$$

因此， P 的列必须是 A 的特征向量。

我们对角化的定义要求 $P \in R[n \times n]$ 是可逆的，即 P 具有满秩（定理 4.3）。这要求我们有 n 个线性无关的特征向量 $p_1, \dots, p[n]$ ，即它们形成 $R[n]$ 的基。

定理 4.20（特征分解）。方阵 $A \in R[n \times n]$ 可以分解为

$$A = PDP[-1], \quad (4.55)$$

其中 $P \in \mathbb{R}[n \times n]$ 且 D 是对角元素为 A 的特征值的对角矩阵，当且仅当 A 的特征向量形成 $\mathbb{R}[n]$ 的基。

图 4.7 特征分解作为顺序变换的直观理解。从左上到左下： $P[-1]$ 执行基变换（这里在 $\mathbb{R}[2]$ 中绘制并描述为类似旋转的操作）从标准基到特征基。从左下到右下： D 沿重新映射的正交特征向量执行缩放，这里通过圆被拉伸成椭圆来描述。从右下到右上： P 撤销基变换（描述为反向旋转）并恢复

图 4.7 特征分解作为顺序变换的直观理解。从左上到左下： $P[-1]$ 执行基变换（这里在 $\mathbb{R}[2]$ 中绘制并描述为类似旋转的操作）从标准基到特征基。从左下到右下： D 沿重新映射的正交特征向量执行缩放，这里通过圆被拉伸成椭圆来描述。从右下到右上： P 撤销基变换（描述为反向旋转）并恢复

定理 4.20 意味着只有非亏损矩阵可以被对角化，并且 P 的列是 A 的 n 个特征向量。对于对称矩阵，我们可以为特征值分解获得更强的结果。

定理 4.21。 对称矩阵 $S \in \mathbb{R}[n \times n]$ 总是可以被对角化。

定理 4.21 直接从谱定理 4.15 得出。此外，谱定理表明我们可以找到特征向量的 ONB

矩阵 $[n] [T] [原坐标] R$ 。这使得 P 成为一个正交矩阵，从而 $D = P A P$ 。[框架。]

注释。矩阵的 Jordan 标准形提供了一个分解，适用于缺陷矩阵 [(Lang, 1987)]，但超出了本书的范围。◇

特征分解的几何直觉

我们可以按如下方式解释矩阵的特征分解（另见图 [4.7]）：] 设 A 是相对于标准基 $-e$ （蓝色箭头）的线性映射的变换矩阵。 P 执行从标准基到特征基的基变换。然后，对角矩阵 D 沿这些轴按特征值 $\lambda[i]$ 缩放向量。最后， P 将这些缩放后的向量变换回标准/规范坐标，得到 $\lambda[i] p[i]$ 。

例 4.11 (特征分解)

5 - 2

让我们计算 A 的特征分解。

[2] -2 5

步骤 1: 计算特征值和特征向量。 特征多项式

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[118] [[矩阵分解]]

A 的特征多项式为

$$\det(A - [2] \begin{bmatrix} 1 & 5 \\ 5 & -2 \end{bmatrix} - \lambda I) = \det \begin{bmatrix} 5 & -2 \\ 5 & -2 \end{bmatrix} - \lambda \det \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (4.56a)$$

$$= \det \begin{bmatrix} 5 & -2 \\ 5 & -2 \end{bmatrix} - \lambda \det \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$= (5 \cdot (-2) - (-2) \cdot 5) - \lambda (1 \cdot (-1) - 1 \cdot 1) = (-10 + 10) - \lambda (-1 - 1) = 0 - \lambda (-2) = 2\lambda \quad (4.56b)$$

$\begin{bmatrix} 2 \\ 4 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix}$

因此，A 的特征值为 $\lambda = 7$ 和 $\lambda = 3$ （特征多项式的根），相关的（标准化）特征向量

$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix}$

通过以下方式获得

$\begin{bmatrix} 7 \\ 3 \end{bmatrix}$

$A\mathbf{p}_1 = \mathbf{p}_1, A\mathbf{p}_2 = \mathbf{p}_2 \cdot [(4.57)]$

$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$

这产生

$\mathbf{p}_1 = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \mathbf{p}_2 = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 \\ -2 \end{bmatrix} \cdot [(4.58)]$

$\begin{bmatrix} 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

步骤 2: 检查存在性。 $\begin{bmatrix} 2 \end{bmatrix}$ 特征向量 $\mathbf{p}_1, \mathbf{p}_2$ 构成 $\mathbb{R}^2$ 的基。

$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$

因此，A 可以对角化。

步骤 3: 构造矩阵 P 来对角化 A。 我们将 A 的特征向量收集在 P 中，使得

$P = [\mathbf{p}_1, \mathbf{p}_2] = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 1 \\ 2 & -2 \end{bmatrix} \cdot [(4.59)]$

$\begin{bmatrix} 2 \\ - \end{bmatrix}$

然后我们得到

$P^{-1} = \begin{bmatrix} 2 & 1 \\ 7 & 0 \end{bmatrix}$

$AP = \Lambda = D \cdot [(4.60)]$

$\begin{bmatrix} 0 \\ 2 \end{bmatrix}$

[图 4.7 可视化了] 等价地，我们得到（利用 $P^{-1}AP = \Lambda$ ，因为此例中的特征向量

[特征分解] $\mathbf{p}_1$ 和 $\mathbf{p}_2$ 形成正交标准基） $\begin{bmatrix} 2 \end{bmatrix}$

[作为] $\begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 7 \\ 5 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \end{bmatrix} [A] = \begin{bmatrix} -2 & 1 & 1 & 0 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \begin{bmatrix} 5 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 \\ 3 \end{bmatrix} \frac{1}{\sqrt{2}} \cdot [(4.61)] \begin{bmatrix} 2 & -2 \\ 5 & 1 \end{bmatrix}$ [线性变换的] $\begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ [序列。]

$[A] = [P] [\Lambda] [P]^{-1}$

对角矩阵 D 可以高效地计算幂。因此，我们可以通过特征值分解（如果存在）为矩阵 $[n] \times [n]$ $A \in \mathbb{R}$ 求矩阵幂，使得

$$A^k = [k] \times [k] = (P D P^{-1})^k = P D^k P^{-1}. \quad (4.62)$$

计算 $[k] D^k$ 是高效的，因为我们对每个对角元素单独应用此操作。

假设特征分解 $A = P D P^{-1}$ 存在。那么，

$$\det(-A) = \det(P D P^{-1}) = \det(P) \det(D) \det(P^{-1}) \quad (4.63a)$$

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: [\[https://mml-book.com\]](https://mml-book.com).] [4.5 奇异值分解] [119]

$$= \det(Y D) = d \quad (4.63b)$$

[ii]

[i]

允许高效计算 A 的行列式。

特征值分解需要方阵。对一般矩阵执行分解将是有用的。在下一节中，我们介绍一种更一般的矩阵分解技术，即奇异值分解。

4.5 奇异值分解

矩阵的奇异值分解(SVD)是线性代数中的核心矩阵分解方法。它被称为”线性代数的基本定理” [(Strang, 1993)], 因为它可以应用于所有矩阵, 不仅仅是方阵, 而且总是存在。此外, 正如我们将在下面探讨的, 矩阵 A 的 SVD 表示线性映射 $\Phi: V \rightarrow W$, 量化了这两个向量空间之间底层几何的变化。我们推荐 [Kalman (1996)] 和 [Roy and Banerjee (2014)] 的工作, 以更深入地了解 SVD 的数学。

[SVD定理]

定理 4.22 $[m][\times][n]$ (SVD 定理)。 设 $A \in \mathbb{R}$ 是秩为 $r \in [0, \min(m, n)]$ 的矩形矩阵。 A 的 SVD 是形式为 [SVD]

[奇异值]

$[n][m][n]$ [分解]

$[n]$

$[m][m][m][A]=[U][\Sigma][V][T][n]$

(4.64)

的分解, 其中 $[m][\times][m] U \in \mathbb{R}$ 是正交矩阵, 列向量为 $u, i = 1, \dots, m, [i]$

并且 $[n][\times][n] V \in \mathbb{R}$ 是正交矩阵, 列向量为 $v, j = 1, \dots, n, [j]$

此外, Σ 是 $m \times n$ 矩阵, 其中 $\Sigma[ii] = \sigma[i] \geq 0$ 且 $\Sigma[ij] = 0, i \neq j$ 。

Σ 的对角元素 $\sigma[i], i = 1, \dots, r$ 称为 奇异值, [[奇异值]]

$u[i]$ 称为左奇异向量, $v[j]$ 称为右奇异 [[左奇异向量]]

向量。按照惯例, 奇异值是有序的, 即 $\sigma[1] \geq \sigma[2] \geq \dots$ [[右奇异]]

$\sigma[r+1] = \dots = \sigma[n] = 0$ 。

奇异值矩阵 Σ 是唯一的, 但需要注意一些细节。 [[奇异值]]

观察到 $[m][\times][n]$ [矩阵] $\Sigma \in \mathbb{R}$ 是矩形的。特别地, Σ 与 A 大小相同。这意味着 Σ 有一个包含奇异值的对角子矩阵, 需要额外的零填充。具体地, 如果 $m > n$, 那么矩阵 Σ 在第 n 行之前具有对角结构, 然后由

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[120] [[矩阵分解]]

[图4.8] [矩阵]

$[A] \in \mathbb{R}^{3 \times 2}$ [的SVD]

[R] [直觉]

[作为顺序] $v[\sigma][2][u][2]$ [A] [变换。]

[左上到]

[左下:] $[\tau][V][\sigma][u]$

1 1

[在] [R] [中执行基] $[V][2][\tau][U]$ [变换。左下到]

[右下:] $[\Sigma]$

[缩放并从] $[e][2]$

$[2] 3$ [R] [映射到] $[R][\circ.] [\sigma][e]$

$[2] [2]$

[椭圆位于] $[\Sigma]$ [右下角的] $e \in R$ [中。第三个]

[维度]

[垂直于]

[椭圆盘的] $[\tau]$ 从 $n+1$ 到 m 的 0 行向量, 使得 [表面。]

[从右下角到] $\sigma 0 0$

1

[右上角:] $[U] \dots$ [进行基] $0 0$

[变换在] $3 [R]$ [内。] $0 0 \sigma$

$\Sigma [n] = . (4.65)$

$0 \dots 0 \dots$

$0 \dots 0$

如果 $m < n$, 矩阵 Σ 具有对角结构直到第 m 列, 从 $m+1$ 到 n 的列

由 0 组成:

$\sigma 0 0 0 \dots 0$

1

$$\Sigma = \dots 0 \dots (4.66) 0 \dots$$

$$0 \ 0 \ \sigma[m] \ 0 \ \dots \ 0$$

备注。 对于任何矩阵 $A \in \mathbb{R}[m][\times][n]$, SVD 都存在。◇

4.5.1 SVD 的几何直观

SVD 提供了几何直观来描述变换矩阵 A 。在下文中，我们将讨论 SVD 作为在基上执行的顺序线性变换。在例子 4.12 中，我们将把 SVD 的变换矩阵应用到 $\mathbb{R}^2$ 中的一组向量上，这使我们能够更清楚地可视化每个变换的效果。

矩阵的 SVD 可以解释为相应线性映射的分解（回想第 2.7.1 节） $\Phi: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 为三个操作；见图 4.8。SVD 直观在表面上遵循与我们特征分解直观类似的结构，见图 4.7：广义上说，SVD 通过 $V[T]$ 执行基变换，然后通过奇异值

Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>. 4.5 奇异值分解 121

矩阵 Σ 进行缩放和维度扩充（或降维）。最后，它通过 U 执行第二次基变换。SVD 包含许多重要的细节和注意事项，这就是为什么我们将更详细地回顾我们的直观理解。回顾基变换是有用的

假设我们给定一个线性映射的变换矩阵 $\Phi: \mathbb{R}^n \rightarrow \mathbb{R}^m$ ，相对于 $\mathbb{R}^n$ 和 $\mathbb{R}^m$ 的标准基 B 和 C （第 2.7.2 节）。此外，假设第二个基 $\tilde{B}$ 和 $\tilde{C}$ 分别属于 $\mathbb{R}^n$ 和 $\mathbb{R}^m$ 。然后正交矩阵（定义 3.8）和正交标准基（第 3.5 节）

1. 矩阵 V 在定义域 $\mathbb{R}^n$ 中执行从 B （由图 4.8 左上角的红色和橙色向量 v_1 和 v_2 表示）到标准基 $\tilde{B}$ 的基变换。
 $V[T] = V[-1]$ 执行从 $\tilde{B}$ 到 B 的基变换。红色和橙色向量现在与图 4.8 左下角的规范基对齐。
2. 将坐标系更改为 $\tilde{B}$ 后， Σ 按奇异值 $\sigma[i]$ 缩放新坐标（并添加或删除维度），即， Σ 是 Φ 相对于 $\tilde{B}$ 和 $\tilde{C}$ 的变换矩阵，由红色和橙色向量被拉伸并位于 e_1 - e_2 平面中表示，该平面现在嵌入在图 4.8 右下角的第三维中。
3. U 在上域 $\mathbb{R}^m$ 中执行从 $\tilde{C}$ 到 $\mathbb{R}^m$ 规范基的基变换，由红色和橙色向量从 e_1 - e_2 平面的旋转表示。这在图 4.8 的右上角显示。

SVD 表达了定义域和上域中的基变换。这与在同一向量空间内操作的特征分解形成对比，其中应用相同的基变换然后撤消。使 SVD 特殊的是这两个不同的基通过奇异值矩阵 Σ 同时连接。

例子 4.12（向量和 SVD）

考虑一个向量网格 $X \in \mathbb{R}^2$ 的映射，它们适合位于原点的 2×2 大小的盒子中。使用标准基，我们使用以下方式映射这些向量

$$1 \quad 0.8$$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = U \Sigma V^T \quad (4.67a)$$

$$1 \quad 0$$

$$= \begin{bmatrix} 0.79 & 0 \\ 0 & 0.62 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.62 & 0 \\ 0 & 0.79 \end{bmatrix} \quad (4.67b)$$

$$= \begin{bmatrix} 0.38 & -0.78 \\ -0.49 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.62 & 0 \\ 0 & 0.78 \end{bmatrix} \quad (4.67c)$$

$- .48 \ -0.62 \ 0.62 \ 0 \ 0 \ 0 \ - .62 \ -0.78$

我们从一组向量 X （彩色点；见图 4.9 的左上面板）开始，排列成网格。然后我们应用 $V[T] \in \mathbb{R}[2 \times 2]$ ，它旋转 X 。旋转后的向量显示在图 4.9 的左下面板中。我们现在使用奇异值矩阵 Σ 将这些向量映射到上域 $\mathbb{R}^3$ （见图 4.9 中的右下面板）。注意所有向量都位于

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

122 矩阵分解

x_1 - x_2 平面中。第三个坐标总是 0。 x_1 - x_2 平面中的向量已被奇异值拉伸。

向量 X 通过 A 到上域 $\mathbb{R}^3$ 的直接映射等于 X 通过 $U \Sigma V[T]$ 的变换，其中 U 在上域 $\mathbb{R}^3$ 内执行旋转，使得映射的向量不再限制在 x_1 - x_2 平面内；如图 4.9 右上面板所示，它们仍在一个平面上。



SVD 和向量映射（由圆盘表示）。面板遵循图 4.8 的相同逆时针结构。

4.5.2 SVD 的构造

接下来我们将讨论为什么 SVD 存在，并详细展示如何计算它。一般矩阵的 SVD 与方阵的特征分解。

注释。比较 SPD 矩阵的特征分解

$$S^T = S = P D P^T \quad (4.68)$$

[《机器学习数学》草稿 (2023-10-18)。反馈：] [https://mml-book.com][。] [4.5 奇异值分解] [123]

与相应的 SVD

$$S = U \Sigma V^T. \quad (4.69)$$

如果我们设

$$U = P = V, \quad D = \Sigma, \quad (4.70)$$

我们可以看到 SPD 矩阵的 SVD 就是它们的特征分解。◇

接下来，我们将探讨为什么定理 4.22 成立以及 SVD 是如何构造的。计算 $m \times n$ 矩阵 $A \in \mathbb{R}^{m \times n}$ 的 SVD 等价于找到两组标准正交基 $U = (u_1, \dots, u_m)$ 和 $V = (v_1, \dots, v_n)$ ，分别属于上域 $\mathbb{R}^m$ 和定义域 $\mathbb{R}^n$ 。从这些有序基中，我们将构造矩阵 U 和 V 。

我们的计划是首先构造标准正交的右奇异向量集 $v_1, \dots, v_n \in \mathbb{R}^n$ 。然后构造标准正交的左奇异向量集 $u_1, \dots, u_m \in \mathbb{R}^m$ 。之后，我们将这两者联系起来，并要求 v_i 的正交性在 A 的变换下得以保持。这很重要，因为我们知道像 Av_i 形成一组正交向量。然后通过标量因子对这些像进行归一化，这些标量因子将是奇异值。

让我们从构造右奇异向量开始。谱定理(定理 4.15)告诉我们，对称矩阵的特征向量形成一个 ONB，这也意味着它可以被对角化。此外，从定理 4.14 我们知道，总是可以从任何矩形矩阵 $A \in \mathbb{R}^{m \times n}$ 构造一个对称的半正定矩阵 $A^T A \in \mathbb{R}^{n \times n}$ 。因此，我们总是可以对角化 $A^T A$ 并得到

$$A^T A = P D P^T = P \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} P^T, \quad (4.71)$$

其中 P 是一个正交矩阵，由标准正交特征基组成。 $\lambda_i \geq 0$ 是 $A^T A$ 的特征值。让我们假设 A 的 SVD 存在并将 (4.64) 代入 (4.71)。这产生

$$A^T A = (U \Sigma V^T)^T (U \Sigma V^T) = V \Sigma^T U^T U \Sigma V^T, \quad (4.72)$$

其中 U^T, V 是正交矩阵。因此，由于 $U U^T = I$ ，我们得到

$$A^T A = V \Sigma^2 V^T = V \begin{bmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_n^2 \end{bmatrix} V^T. \quad (4.73)$$

现在比较 (4.71) 和 (4.73)，我们识别出

$$V^T = P^T, \quad (4.74) \quad \sigma_i^2 = \lambda_i. \quad (4.75)$$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[124] [矩阵分解]

因此，构成P的 $A^T A$ 的特征向量是A的右奇异向量 V^T (见(4.74))。 $A^T A$ 的特征值是 Σ 的平方奇异值(见(4.75))。

为了获得左奇异向量U，我们遵循类似的过程。我们从计算对称矩阵 $AA^T \in \mathbb{R}^{m \times m}$ 的SVD开始(而不是之前的 $A^T A \in \mathbb{R}^{n \times n}$)。A的SVD产生

$$AA^T = (U \Sigma V^T)(U \Sigma V^T)^T = U \Sigma V^T V \Sigma^T U^T \quad (4.76a) = U [\sigma_1^2 \ 0 \ 0; \dots; 0 \ 0 \ 0 \ \sigma_m^2] U^T. \quad (4.76b)$$

谱定理告诉我们 $AA^T = SDS^T$ 可以被对角化，我们可以找到 AA^T 的特征向量的ONB，它们被收集在 S^T 中。 AA^T 的标准正交特征向量是左奇异向量U，并在SVD的上域中形成标准正交基。

这留下了矩阵 Σ 结构的问题。由于 AA^T 和 $A^T A$ 具有相同的非零特征值(见第106页)，在两种情况下SVD中 Σ 矩阵的非零条目必须相同。

最后一步是连接我们到目前为止涉及的所有部分。我们在V中有一组标准正交的右奇异向量。为了完成SVD的构造，我们将它们与标准正交向量U连接起来。为了达到这个目标，我们使用 v_i 在A下的像也必须是正交的这一事实。我们可以通过使用第3.4节的结果来证明这一点。我们要求 Av_i 和 Av_j 之间的内积对于 $i \neq j$ 必须为0。对于任何两个正交特征向量 $v_i, v_j, i \neq j$ ，有

$$(Av_i)^T (Av_j) = v_i^T (A^T A) v_j = v_i^T (\lambda_j v_j) = \lambda_j v_i^T v_j = 0. \quad (4.77)$$

对于 $m \geq r$ 的情况， $\{Av_1, \dots, Av_r\}$ 是 $\mathbb{R}^m$ 的r维子空间的基。

为了完成SVD构造，我们需要标准正交的左奇异向量：我们对右奇异向量的像 Av_i 进行归一化并得到

$$u_i := Av_i / \|Av_i\| = (1/\sqrt{\lambda_i}) Av_i = (1/\sigma_i) Av_i, \quad (4.78)$$

其中最后的等式从(4.75)和(4.76b)得到，显示 $A^T A$ 的特征值使得 $\lambda_i = \sigma_i^2$ 。

因此， $A^T A$ 的特征向量，我们知道它们是右奇异向量 v_i ，以及它们在A下的归一化像，即左奇异向量 u_i ，形成两个自洽的ONB，通过奇异值矩阵 Σ 连接。

让我们重新整理(4.78)得到奇异值方程

$$Av_i = \sigma_i u_i, \quad i = 1, \dots, r. \quad (4.79)$$

[《机器学习数学》草稿(2023-10-18)。反馈：][<https://mml-book.com>][。][4.5 奇异值分解][125]

这个方程与特征值方程(4.25)非常相似，但左边和右边的向量不相同。

对于 $n < m$ ，(4.79)仅对 $i \leq n$ 成立，但(4.79)对于 $i > n$ 的 u_i 没有说明什么。然而，我们通过构造知道它们是标准正交的。相反，对于 $m < n$ ，(4.79)仅对 $i \leq m$ 成立。对于 $i > m$ ，我们有 $Av_i = 0$ ，我们仍然知道 v_i 形成标准正交集。这意味着SVD还提供了A的核(零空间)的标准正交基，即满足 $Ax = 0$ 的向量x的集合(见第2.7.3节)。

将 $v[i]$ 作为 V 的列, $u[i]$ 作为 U 的列进行拼接, 得到

$$AV = U\Sigma, \quad (4.80)$$

其中 Σ 与 A 具有相同的维度, 对于第 $1, \dots, r$ 行具有对角结构。因此, 右乘 V^T 得到 $A = U\Sigma V^T$, 这就是 A 的 SVD。

例题 4.13 (计算 SVD)

让我们求解以下矩阵的奇异值分解

$$A = \begin{bmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix}. \quad (4.81)$$

SVD 需要我们计算右奇异向量 $v[j]$ 、奇异值 $\sigma[k]$ 和左奇异向量 $u[i]$ 。

步骤 1: 右奇异向量作为 $A^T A$ 的特征基

我们首先计算

$$A^T A = \begin{bmatrix} 1 & -2 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 5 & -2 & 1 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}. \quad (4.82)$$

我们通过 $A^T A$ 的特征值分解来计算奇异值和右奇异向量 $v[j]$, 结果为

$$A^T A = \begin{bmatrix} 5/\sqrt{30} & -1/\sqrt{6} & 0 \\ -2/\sqrt{30} & 1/\sqrt{5} & -2/\sqrt{6} \\ 1/\sqrt{30} & 2/\sqrt{5} & 1/\sqrt{6} \end{bmatrix} \begin{bmatrix} 6 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 5/\sqrt{30} & -2/\sqrt{30} & 1/\sqrt{30} \\ -1/\sqrt{6} & 1/\sqrt{5} & 2/\sqrt{5} \\ 0 & -2/\sqrt{6} & 1/\sqrt{6} \end{bmatrix} = P D P^T, \quad (4.83)$$

我们得到右奇异向量作为 P 的列, 因此

$$V = P = \begin{bmatrix} 5/\sqrt{30} & -1/\sqrt{6} & 0 \\ -2/\sqrt{30} & 1/\sqrt{5} & -2/\sqrt{6} \\ 1/\sqrt{30} & 2/\sqrt{5} & 1/\sqrt{6} \end{bmatrix}. \quad (4.84)$$

步骤 2: 奇异值矩阵

由于奇异值 $\sigma[i]$ 是 $A^T A$ 特征值的平方根, 我们直接从 D 中获得它们。由于 $\text{rk}(A) = 2$, 只有两个非零奇异值: $\sigma_1 = \sqrt{6}$ 和 $\sigma_2 = 1$ 。奇异值矩阵必须与 A 具有相同的尺寸, 我们得到

$$\Sigma = \begin{bmatrix} \sqrt{6} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad (4.85)$$

步骤 3: 左奇异向量作为右奇异向量的归一化像

我们通过计算右奇异向量在 A 下的像并除以对应的奇异值来归一化, 从而找到左奇异向量。我们得到

$$u_1 = (1/\sigma_1) A v_1 = (1/\sqrt{6}) \begin{bmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix} \begin{bmatrix} 5/\sqrt{30} \\ -2/\sqrt{30} \\ 1/\sqrt{30} \end{bmatrix} = \begin{bmatrix} 1/\sqrt{5} \\ -2/\sqrt{5} \end{bmatrix}, \quad (4.86)$$

$$u_2 = (1/\sigma_2) A v_2 = \begin{bmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix} \begin{bmatrix} -1/\sqrt{6} \\ 1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix} = \begin{bmatrix} 2/\sqrt{5} \\ 1/\sqrt{5} \end{bmatrix}, \quad (4.87)$$

$$U = [u_1, u_2] = \begin{bmatrix} 1/\sqrt{5} & 2/\sqrt{5} \\ -2/\sqrt{5} & 1/\sqrt{5} \end{bmatrix}. \quad (4.88)$$

注意在计算机上, 这里说明的方法具有较差的数值行为, A 的 SVD 通常不通过 $A^T A$ 的特征值分解来计算。

4.5.3 特征值分解与奇异值分解

让我们考虑特征分解 $A = PDP^{-1}$ 和 SVD $A = U\Sigma V^T$ ，并回顾前面章节的核心要素。

SVD 对于任何 $m \times n$ 的矩阵 $A \in \mathbb{R}^{(m \times n)}$ 总是存在的。特征分解只对 $n \times n$ 的方阵 $A \in \mathbb{R}^{(n \times n)}$ 定义，并且只有当我们能找到 $\mathbb{R}^n$ 的特征向量基时才存在。特征分解矩阵 P 中的向量不一定正交，即基变换不是简单的旋转和缩放。另一方面，SVD 中矩阵 U 和 V 中的向量是标准正交的，因此它们确实表示旋转。

特征分解和 SVD 都是三个线性映射的复合：

1. 定义域中的基变换
2. 每个新基向量的独立缩放以及从定义域到值域的映射
3. 值域中的基变换

图 4.10 三个人对四部电影的评分及其 SVD 分解。

Ali Beatrix Chandra Star Wars [5 4 1] = [-0.6710 0.0236 0.4647] [-0.5774] Blade Runner [5 5 0] [-0.7197 0.2054 -0.4759] [0.4619]
Amelie [0 0 5] [-0.0939 -0.7705 -0.5268] [-0.3464] Delicatessen [1 0 4] [-0.1515 -0.6030 0.5293] [-0.5774]

$\times [9.6438 \ 0 \ 0] \times [-0.7367 \ -0.6515 \ -0.1811] [0 \ 6.3639 \ 0] [0.0852 \ 0.1762 \ -0.9807] [0 \ 0 \ 0.7056] [0.6708 \ -0.7379 \ -0.0743]$

特征分解和 SVD 的一个关键区别是，在 SVD 中，定义域和值域可以是不同维度的向量空间。

在 SVD 中，左奇异向量矩阵和右奇异向量矩阵 U 和 V 通常不互为逆矩阵（它们在不同的向量空间中执行基变换）。在特征分解中，基变换矩阵 P 和 P^{-1} 互为逆矩阵。

在 SVD 中，对角矩阵 Σ 中的条目都是实数且非负的，而特征分解中的对角矩阵通常不满足这个条件。

SVD 和特征分解通过它们的投影密切相关：

- A 的左奇异向量是 AA^T 的特征向量
- A 的右奇异向量是 A^TA 的特征向量
- A 的非零奇异值是 AA^T 和 A^TA 的非零特征值的平方根

对于对称矩阵 $A \in \mathbb{R}^{(n \times n)}$ ，特征值分解和 SVD 是相同的，这由谱定理 4.15 得出。

示例 4.14（在电影评分和消费者中发现结构）

让我们通过分析人们及其偏爱电影的数据，为SVD添加一个实际的解释。考虑三个观众（Ali、Beatrix、Chandra）对四部不同电影（星球大战、银翼杀手、天使爱美丽、黑店狂想曲）进行评分。他们的评分是0（最差）到5（最好）之间的值，并且

编码在如图4.10所示的数据矩阵 $A \in \mathbb{R}^8$ 中。每一行代表一部电影，每一列代表一个用户。因此，电影评分的列向量，每个观众一个，分别是 $x[\text{Ali}]$ 、 $x[\text{Beatrix}]$ 、 $x[\text{Chandra}]$ 。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[128] [矩阵分解]

使用SVD对 A 进行因式分解为我们提供了一种捕捉人们如何对电影评分关系的方法，特别是如果存在一种连接哪些人喜欢哪些电影的结构。将SVD应用于我们的数据矩阵 A 做出了许多假设：

1. 所有观众都使用相同的线性映射一致地对电影评分。
2. 评分中没有错误或噪声。
3. 我们将左奇异向量 $u[i]$ 解释为典型电影，将右奇异向量 $v[j]$ 解释为典型观众。

然后我们假设任何观众的特定电影偏好都可以表示为 $v[j]$ 的线性组合。类似地，任何电影的受欢迎程度都可以表示为 $u[i]$ 的线性组合。因此，SVD域中的向量可以解释为“典型观众空间”中的观众，SVD余域中的向量相应地可以解释为“典型电影空间”中的电影。

这两个“空间”只有在相应的观众和电影数据本身涵盖了足够多样性的观众和电影时才有意义。

让我们检查我们的电影-用户矩阵的SVD。第一个左奇异向量 u_1 对两部科幻电影有很大的绝对值，并且有很大的第一个奇异值（图4.10中的红色阴影）。因此，这将一类用户与一组特定的电影（科幻主题）归为一组。类似地，第一个右奇异向量 v_1 对Ali和Beatrix显示很大的绝对值，他们对科幻电影给出高分（图4.10中的绿色阴影）。这表明 v_1 反映了科幻爱好者的概念。

类似地， u_2 似乎捕获了法国艺术电影主题， v_2 表明Chandra接近这类电影的理想化爱好者。理想化的科幻爱好者是纯粹主义者，只喜欢科幻电影，所以科幻爱好者 v_1 对除科幻主题之外的所有内容都给出零评分——这种逻辑是由奇异值矩阵 Σ 的对角子结构暗示的。因此，特定电影通过它如何分解（线性地）为典型电影来表示。同样，一个人将通过他们如何分解（通过线性组合）为电影主题来表示。

值得简要讨论SVD术语和约定，因为文献中使用了不同的版本。虽然这些差异可能令人困惑，但数学对它们保持不变。

为了符号表示和抽象的方便，我们使用SVD符号表示，其中SVD被描述为具有两个方形左奇异向量矩阵和右奇异向量矩阵，但具有非方形奇异值矩阵。我们对SVD的定义(4.64)有时被称为完全SVD(full SVD)。

一些作者对SVD的定义稍有不同，专注于方形奇异矩阵。然后，对于 $A \in \mathbb{R}^9$ 且 $m \geq n$,

$$A^{\wedge T} = U \Sigma V^{\wedge T}. \quad (4.89) \quad [m \times n] \quad [m \times n] \quad [n \times n] \quad [n \times n]$$

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>.]

4.6 矩阵近似 [129]

有时这种公式被称为 *简化SVD*(reduced SVD) (例如, Datta (2010)) 或 *SVD* (例如, Press et al. (2007))。这种替代格式仅改变矩阵的构造方式, 但保持SVD的数学结构不变。这种替代公式的便利性在于 Σ 是对角的, 就像特征值分解一样。

在第4.6节中, 我们将学习使用SVD的矩阵近似技术, 这也被称为 *截断SVD*(truncated SVD)。

可以定义秩为 r 的矩阵 A 的SVD, 使得 U 是 $m \times r$ 矩阵, Σ 是对角矩阵 $r \times r$, V 是 $r \times n$ 矩阵。这种构造与我们的定义非常相似, 并确保对角矩阵 Σ 在对角线上只有非零条目。这种替代符号表示的主要便利性在于 Σ 是对角的, 就像特征值分解一样。

SVD对 A 只适用于 $m > n$ 的 $m \times n$ 矩阵的限制在实践中是不必要的。当 $m < n$ 时, SVD分解将产生具有比行数更多零列的 Σ , 因此, 奇异值 $\sigma[m+1], \dots, \sigma[n]$ 为0。

SVD在机器学习的各种应用中被使用, 从曲线拟合中的最小二乘问题到求解线性方程组。这些应用利用了SVD的各种重要特性、它与矩阵秩的关系, 以及它用低秩矩阵近似给定秩矩阵的能力。用矩阵的SVD替换矩阵通常具有使计算对数值舍入误差更加鲁棒的优势。正如我们将在下一节中探索的, SVD以原则性方式用“更简单”的矩阵近似矩阵的能力开启了从维度降低和主题建模到数据压缩和聚类的机器学习应用。

4.6 矩阵近似

我们将SVD视为将 $A \in \mathbb{R}^{(m \times n)}$ 分解为 $A = U \Sigma V^T$ 的方法，即分解为三个矩阵的乘积，其中 $U \in \mathbb{R}^{(m \times m)}$ 和 $V \in \mathbb{R}^{(n \times n)}$ 是正交矩阵， Σ 在主对角线上包含奇异值。与其进行完整的SVD分解，我们现在将研究SVD如何允许我们将矩阵A表示为更简单的（低秩）矩阵 A_i 的和，这有助于建立比完整SVD计算成本更低的矩阵近似方案。

我们构造一个秩-1矩阵 $A_i \in \mathbb{R}^{(m \times n)}$ 为

$$A_i := u_i v_i^T, \quad (4.90)$$

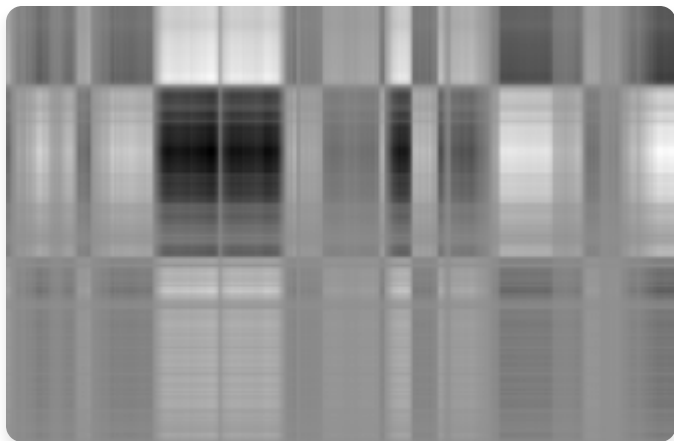
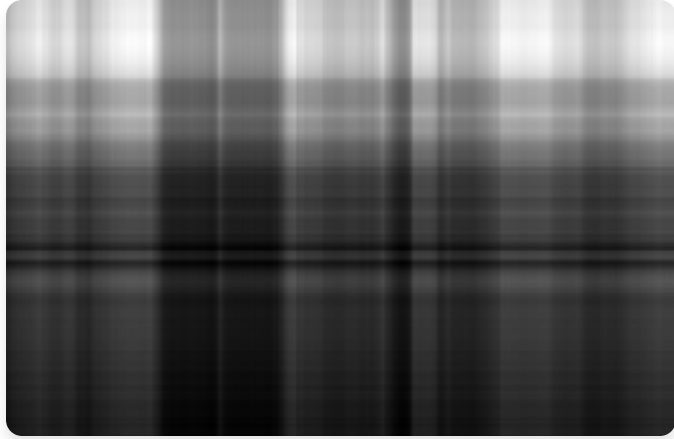
它由U和V的第i个正交列向量的外积形成。图4.11显示了巨石阵的图像，可以用矩阵 $A \in \mathbb{R}^{(1432 \times 1910)}$ 表示，以及一些外积 A_i ，如(4.90)中定义的。

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

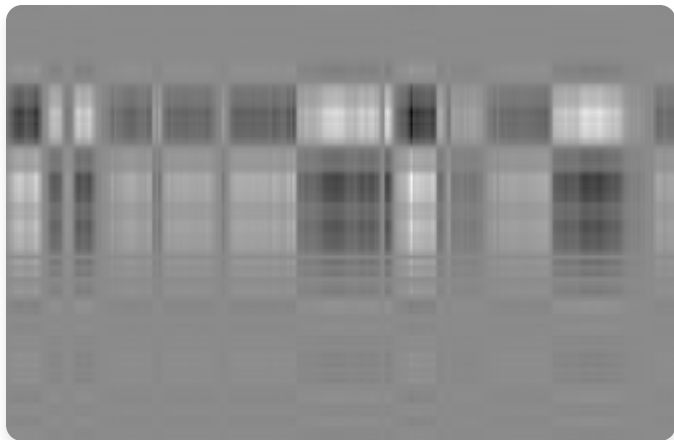
130 Matrix Decompositions

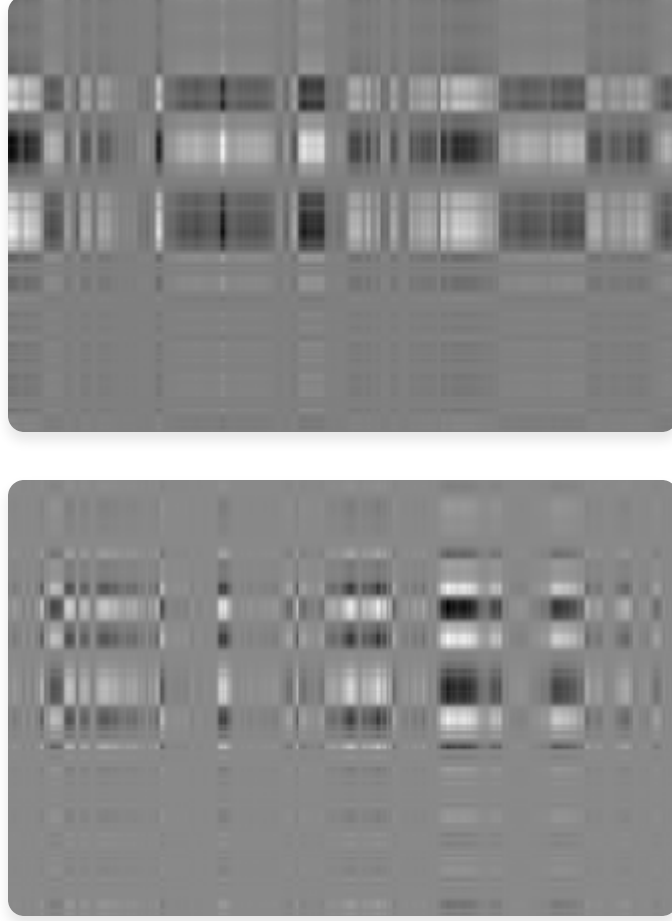
图4.11 使用SVD进行图像处理。(a) 原始灰度图像是一个 1432×1910 的矩阵，值在0（黑色）和1（白色）之间。(b)-(f) 秩-1矩阵 $A_1, \dots, A_5$ 及其对应的奇异值 $\sigma_1, \dots, \sigma_5$ 。每个秩-1矩阵的网格状结构是由左奇异向量和右奇异向量的外积产生的。





a. 原始图像A. (b) $A_1, \sigma_1 \approx 228,052$. (c) $A_2, \sigma_2 \approx 40,647$.





d. $A_3, \sigma_3 \approx 26,125$. (e) $A_4, \sigma_4 \approx 20,232$. (f) $A_5, \sigma_5 \approx 15,436$.

秩为 r 的矩阵 $A \in \mathbb{R}^{(m \times n)}$ 可以写成秩-1矩阵 A_i 的和，使得

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T = \sum_{i=1}^r \sigma_i A_i, \quad (4.91)$$

其中外积矩阵 A_i 由第 i 个奇异值 σ_i 加权。我们可以看出为什么(4.91)成立：奇异值矩阵 Σ 的对角结构只将匹配的左奇异向量和右奇异向量 $u_i v_i^T$ 相乘，并用相应的奇异值 σ_i 缩放它们。所有项 $\sum_{ij} \sigma_{ij} u_i v_j^T$ 在 $i \neq j$ 时消失，因为 Σ 是对角矩阵。任何 $i > r$ 的项都消失，因为相应的奇异值为0。

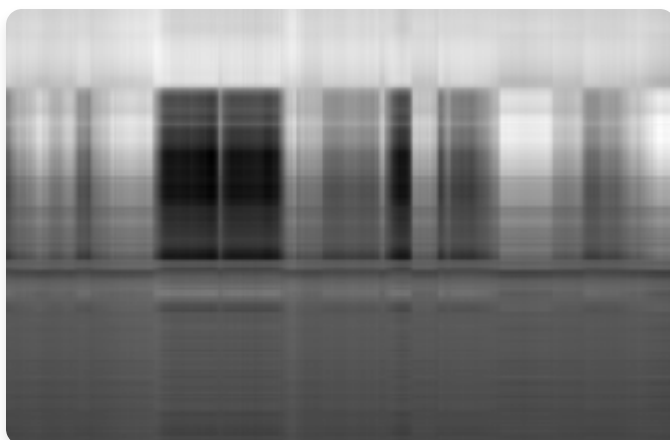
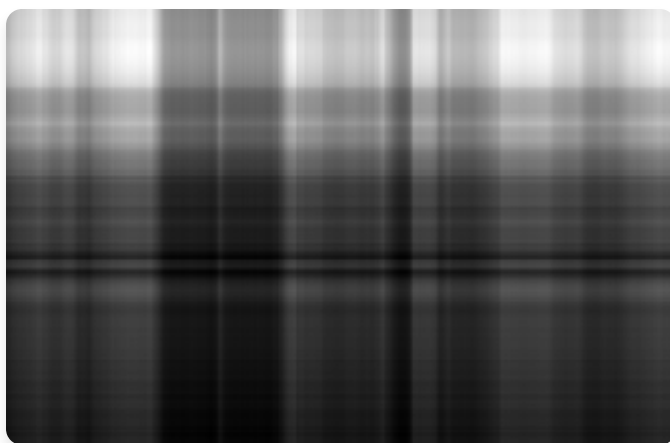
在(4.90)中，我们引入了秩-1矩阵 A_i 。我们将 r 个独立的秩-1矩阵求和得到秩- r 矩阵 A ；见(4.91)。如果求和不是对所有矩阵 $A_i, i = 1, \dots, r$ 进行，而只到中间值 $k < r$ ，我们得到一个**秩- k 近似**

$$\hat{A}^{(k)} := \sum_{i=1}^k \sigma_i u_i v_i^T = \sum_{i=1}^k \sigma_i A_i \quad (4.92)$$

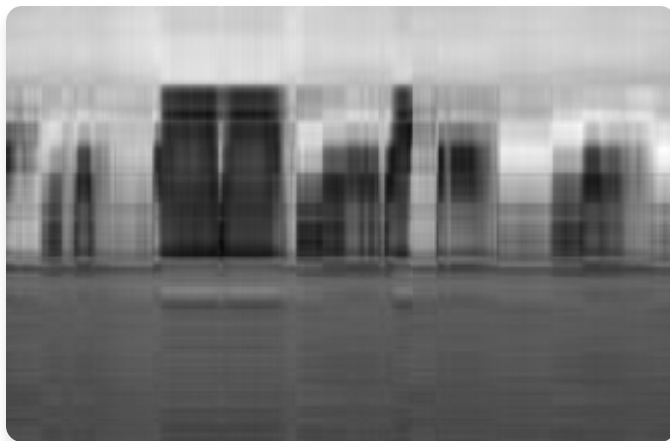
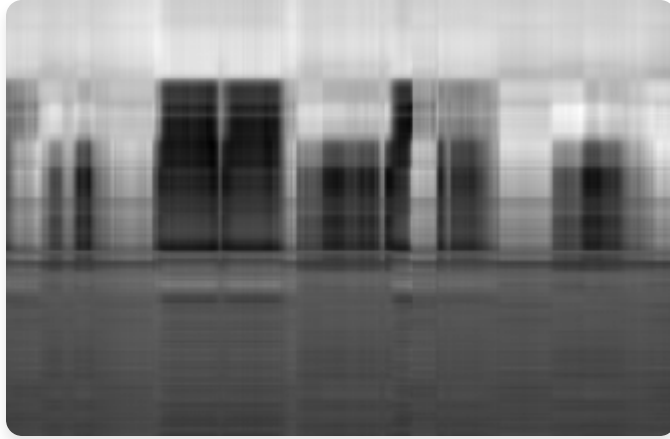
A 的近似，其中 $\text{rk}(\hat{A}^{(k)}) = k$ 。图4.12显示了巨石阵原始图像 A 的低秩近似 $\hat{A}^{(k)}$ 。在秩-5近似中，岩石的形状变得越来越清晰可见和容易识别。虽然原始图像需要 $1,432 \cdot 1,910 = 2,735,120$ 个数字，但秩-5近似只需要存储五个奇异值和五个左、右奇异向量（分别为1,432维和1,910维），总共 $5 \cdot (1,432 + 1,910 + 1) = 16,715$ 个数字——仅占原始数据的0.6%多一点。

要测量 A 与其秩- k 近似 $\hat{A}^{(k)}$ 之间的差异（误差），我们需要范数的概念。在第3.1节中，我们已经使用了

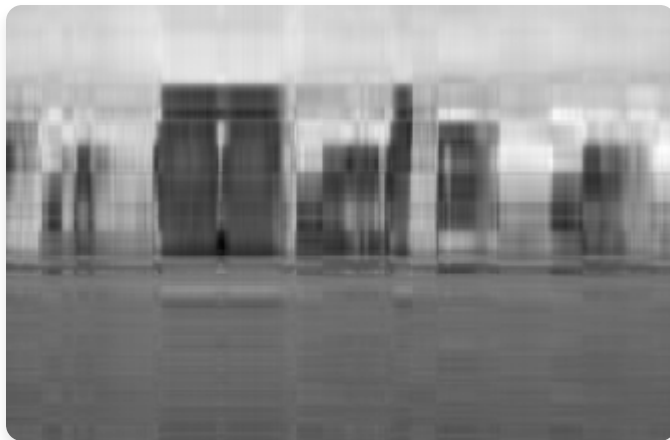
图4.12 使用SVD进行图像重建。(a) 原始图像。(b)-(f) 使用SVD的低秩近似进行图像重建，其中秩-k近似由 $\hat{A}^{(k)} = \sum_{i=1}^k \sigma_i A_i$ 给出。



a. 原始图像 (b) 秩-1近似 $\hat{A}^{(1)}$ (c) 秩-2近似 $\hat{A}^{(2)}$



d. 秩-3近似 $\hat{A}^{(3)}$ (e) 秩-4近似 $\hat{A}^{(4)}$ (f) 秩-5近似 $\hat{A}^{(5)}$



测量向量长度的向量范数。类似地，我们也可以定义矩阵上的范数。

定义4.23（矩阵的谱范数）。对于 $x \in \mathbb{R}^n \setminus \{0\}$ ，矩阵 $A \in \mathbb{R}^{(m \times n)}$ 的**谱范数**定义为

$$\|A\|_2 := \max\{x\} \frac{\|Ax\|_2}{\|x\|_2} \quad (4.93)$$

我们在矩阵范数中引入下标记号（左边），类似于向量的欧几里得范数（右边），它有个下标2。谱范数(4.93)确定了任何向量x在被A相乘时最多能变得多长。

定理4.24. A的谱范数是其最大奇异值 σ_1 。

我们将此定理的证明留作练习。

定理4.25 (Eckart-Young定理 (Eckart and Young, 1936))。考虑秩为r的矩阵 $A \in \mathbb{R}^{m \times n}$ ，设 $B \in \mathbb{R}^{m \times n}$ 为秩为k的矩阵。对于任意 $k \leq r$ ，其中 $\hat{A}(k) = \sum_{i=1}^k \sigma_i u_i v_i^T$ ，有

$$\hat{A}(k) = \operatorname{argmin}_{\{\operatorname{rk}(B)=k\}} \|A - B\|_2, \quad (4.94)$$

$$\|A - \hat{A}(k)\|_2 = \sigma_{k+1}. \quad (4.95)$$

Eckart-Young定理明确说明了通过使用秩-k近似来近似A时我们引入的误差大小。我们可以将用SVD获得的秩-k近似解释为将全秩矩阵A投影到秩不超过k的矩阵的低维空间上。在所有可能的投影中，SVD最小化了A与任何秩-k近似之间的误差（相对于谱范数）。

我们可以回顾一些步骤来理解为什么 [(4.95)] 应该成立。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 剑桥大学出版社出版 (2020).]

[132] [[矩阵分解]]

我们观察到 $A - \hat{A}(k)$ 之间的差是一个包含剩余秩-1矩阵之和的矩阵

[r]

$$A - \hat{A}(k) = \sum_{i=k+1}^r \sigma_i u_i v_i^T \quad (4.96)$$

$$[i] = [k] + 1$$

根据定理 [4.24,] 我们立即得到 σ_{k+1} 作为差矩阵的谱范数。让我们仔细看看 [(4.94).]如果我们假设存在另一个矩阵 B 满足 $\operatorname{rk}(B) \leq k$ ，使得

$$\|A - B\|_2 \leq \|A - \hat{A}(k)\|_2, \quad [(4.97) [2]]$$

那么存在一个至少 $(n - k)$ 维的零空间 $Z \subseteq \mathbb{R}^n$ ，使得 $x \in Z$ 意味着 $Bx = 0$ 。然后可得

$$\|Ax\|_2 = \|(A - B)x\|_2, \quad (4.98) [2]$$

通过使用涵盖矩阵范数(norms)的 Cauchy-Schwartz 不等式 [(3.17)] 的一个版本，我们得到

$$\|Ax\|_2 \leq \|A - B\|_2 \|x\|_2 \leq \sigma_{k+1} \|x\|_2. \quad (4.99) [2] \|x\|_2$$

然而，存在一个 $(k + 1)$ 维子空间，其中 $\|Ax\|_2 \geq \sigma_{k+1} \|x\|_2$ ，该子空间由 A 的右奇异向量 $v_j, j = k + 1, \dots, n$ 张成。将这两个空间的维数相加得到一个大于 n 的数，因为在两个空间中必须存在一个非零向量。这与第 [2.7.3] 节中的秩-零化定理 (Theorem [2.24]) 相矛盾。

Eckart-Young 定理意味着我们可以使用 SVD 以一种有原则的、最优的（在谱范数意义下）方式将秩- r 矩阵 A 降低为秩- k 矩阵 A_k 。我们可以将用秩- k 矩阵逼近 A 解释为一种有损压缩形式。因此，矩阵的低秩逼近出现在许多机器学习应用中，例如图像处理、噪声滤除以及病态问题的正则化。此外，正如我们将在第 [10] 章中看到的，它在降维和主成分分析中起着关键作用。

例 4.15 (在电影评分和消费者中发现结构 (续))

回到我们的电影评分例子，我们现在可以应用低秩逼近的概念来逼近原始数据矩阵。回想一下，我们的第一个奇异值捕获了电影中的科幻主题概念和科幻爱好者。因此，通过在电影评分矩阵的秩-1 分解中仅使用第一个奇异值项，我们得到预测评分

$$-0.6710$$

$$A_1 = u_1 v_1^T = \begin{bmatrix} -0.7367 & -0.6515 & -0.1811 \end{bmatrix} \begin{bmatrix} 4.100a \\ -0.7197 & 0.0939 \end{bmatrix}$$

$$-0.1515$$

[《机器学习数学》草稿 (2023-10-18)。反馈:] [<https://mml-book.com>][.] [4.7 矩阵系统发生学] [[133]]

$$\begin{bmatrix} 0.4943 & 0.4372 & 0.1215 \end{bmatrix}$$

$$\begin{bmatrix} 0.5302 & 0.4689 & 0.1303 \end{bmatrix}$$

$$= \begin{bmatrix} . & (4.100b) \end{bmatrix}$$

$$\begin{bmatrix} 0.0692 & 0.0612 & 0.0170 \end{bmatrix}$$

$$\begin{bmatrix} 0.1116 & 0.0987 & 0.0274 \end{bmatrix}$$

这个第一个秩-1逼近 A_1 很有洞察力：它告诉我们 Ali 和 Beatrix 喜欢科幻电影，如 *Star Wars* 和 *Bladerunner*（条目值 > 0.4 ），但未能捕获 Chandra 对其他电影的评分。这并不令人惊讶，因为 Chandra 的电影类型没有被第一个奇异值捕获。第二个奇异值为那些电影主题爱好者提供了更好的秩-1逼近：

$$0.0236$$

$$A_2 = u_2 v_2^T = \begin{bmatrix} 0.0852 & 0.1762 \end{bmatrix} \begin{bmatrix} -0.2054 \end{bmatrix}$$

$$0.7705$$

$$- \begin{bmatrix} 0.9807 \end{bmatrix} \begin{bmatrix} (4.101a) \end{bmatrix}$$

$$-0.6030$$

$$\begin{bmatrix} 0.0020 & 0.0042 & -0.0231 \end{bmatrix}$$

$$\begin{bmatrix} 0.0175 & 0.0362 & -0.2014 \end{bmatrix}$$

$$= \begin{bmatrix} . & (4.101b) \end{bmatrix} - -$$

0.0656 0.1358 0.7556

-0.0514 -0.1063 0.5914

在这个第二个秩-1逼近 $A[2]$ 中，我们很好地捕获了 Chandra 的评分和电影类型，但没有捕获科幻电影。这使我们考虑秩-2逼近 $A^{(2)}$ ，其中我们结合前两个秩-1逼近

4.7801 4.2419 1.0244 5.2252 4.7522 - 0.0250

$A^{(2)} = \sigma_1 A_1 + \sigma_2 A_2 = (4.102) -$

0.2493 0.2743 4.9724

0.7495 0.2756 4.0278

$A^{(2)}$ 与原始电影评分表相似

5 4 1

5 5 0

$A = (4.103)$

0 0 5

1 0 4

这表明我们可以忽略 A_3 的贡献。我们可以这样解释：在数据表中没有第三个电影主题/电影爱好者类别的证据。这也意味着我们例子中电影主题/电影爱好者的整个空间是一个由科幻和法国艺术电影及其爱好者张成的二维空间。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 剑桥大学出版社出版 (2020).]

[134] [矩阵分解]

[图 4.13] [机器学习中遇到的矩阵的功能系统发生学。]

[实矩阵] [伪逆] $\exists$ [SVD]

$[n] \times [n]$ $[n] \times [m]$ $[R]$ $[R]$ [方阵]

$\exists$ [行列式] [非方阵]

$\exists$ [迹]

[无特征向量基] $[\det = 0]$ $[\det]$ $[\det]$

[缺陷的] [特征向量基] [奇异的] [奇异的] $\exists$ $[\neq]$ $\exists$ $[\neq]$

$[0]$ $[0]$

[非缺陷的]

[(可对角化的)]

[A][T] [T] [T] [T] [A] [=] [AA] [A] [A] / [=] [AA]

[正规的] [非正规的]

[A] [T]

[A]

[=]

[AA]

[对称的] [=] [I] [正则的] [T] [3] [逆矩阵]

[特征值] [€] [R] [(可逆的)]

[对角的] [列是正交的]

[正定的] [特征向量]

[Cholesky]

[单位的] [特征值] [>] [0] [旋转] [正交矩阵]

4.7 矩阵系统发生学

在第 [2] 和 3 章中，我们涵盖了线性代数和解析几何的基础。在本章中，我们研究了矩阵和线性映射的基本特征。图 [4.13] 描绘了不同类型矩阵之间关系的系统发生树（黑色箭头表示

[词语]

[[“系统发生的”]]

[[描述了我们如何]]

[捕获]

[个体或]

[之间的关系]

“是一个子集”)以及我们可以对它们执行的覆盖操作(在[[groups and derived]]

蓝色中)。我们考虑所有来自希腊语的实矩阵 $A \in \mathbb{R}^{(n \times m)}$ ，对于非方矩阵(其中 $n \neq m$)，SVD总是存在的，正如我们在本章中看到的。专注于方矩阵 $A \in \mathbb{R}^{(n \times n)}$ ，行列式告诉我们方矩阵是否拥有逆矩阵，即是否属于正则的、可逆矩阵类。如果方矩阵 $n \times n$ 拥有 n 个线性无关的特征向量，那么该矩阵是非缺陷的，并且存在特征分解(定理 4.12)。我们知道重复的特征值可能导致缺陷矩阵，这些矩阵无法对角化。

非奇异矩阵和非缺陷矩阵不是同一个概念。例如，旋转矩阵是可逆的(行列式非零)，但在实数范围内不可对角化(特征值不保证是实数)。

我们进一步深入非缺陷方矩阵 $n \times n$ 的分支。如果满足条件 $A^T A = A A^T$ ，则矩阵是正规的。此外，如果满足更严格的条件 $A^T A = A A^T = I$ ，则 A 被称为正交的(见定义 3.8)。正交矩阵集合是正则(可逆)矩阵的子集，并满足 $A^{(-1)} = A^T$ 。

正规矩阵有一个经常遇到的子集，即对称矩阵 $S \in \mathbb{R}^{(n \times n)}$ ，满足 $S^T = S$ 。对称矩阵只有实特征值。对称矩阵的一个子集包括正定矩阵 P ，满足对所有 $x \in \mathbb{R}^n \setminus \{0\}$ 都有 $x^T P x > 0$ 的条件。在这种情况下，存在唯一的 Cholesky 分解(定理 4.18)。正定矩阵只有正特征值，且总是可逆的(即具有非零行列式)。

对称矩阵的另一个子集包括对角矩阵 D 。对角矩阵在乘法和加法下封闭，但不一定成群(只有当所有对角元素都非零使得矩阵可逆时才是这种情况)。一个特殊的对角矩阵是单位矩阵 I 。

4.8 延伸阅读

本章的大部分内容建立了底层数学基础，并将它们与研究映射的方法联系起来，其中许多方法是机器学习在底层软件解决方案层面的核心，也是几乎所有机器学习理论的构建块。使用行列式、特征谱和特征空间进行矩阵特征化为分类和分析矩阵提供了基本特征和条件。这扩展到数据表示和涉及数据的映射的所有形式，以及判断此类矩阵计算操作的数值稳定性(Press et al., 2007)。

行列式是“手工”计算矩阵逆和特征值的基本工具。然而，对于几乎所有但最小的实例，通过高斯消元法进行数值计算优于行列式方法(Press et al., 2007)。尽管如此，行列式仍然是一个强大的理论概念，例如，根据行列式的符号获得关于基底方向的直觉。特征向量可用于执行基变换，将数据转换为有意义的正交特征向量坐标。类似地，矩阵分解方法，如Cholesky分解，在我们计算或模拟随机事件时经常重新出现(Rubinstein and Kroese, 2016)。因此，Cholesky分解使我们能够计算重参数化技巧，其中我们希望对随机变量执行连续微分，例如在变分自编码器中(Jimenez Rezende et al., 2014; Kingma and Welling, 2014)。

特征分解是使我们能够提取表征线性映射的有意义和可解释信息的基础。因此，特征分解是一类称为谱方法的机器学习算法的基础，这些算法对正定核执行特征分解。这些谱分解方法涵盖了统计数据分析的经典方法，如下所示：

主成分分析(PCA, Pearson, 1901, 另见第10章)，其中寻求解释数据中大部分变异性的低维子空间。

Fisher判别分析，旨在确定用于数据分类的分离超平面(Mika et al., 1999)。

多维标度(MDS, Carroll and Chang, 1970)。

这些方法的计算效率通常来自于找到对称正半定矩阵的最佳rank-k近似。谱方法的更现代例子有不同的起源，但每种方法都需要计算正定核的特征向量和特征值，如Isomap(Tenenbaum et al., 2000)、Laplacian特征映射(Belkin and Niyogi, 2003)、Hessian特征映射(Donoho and Grimes, 2003)和谱聚类(Shi and Malik, 2000)。这些方法的核心计算通常由低秩矩阵近似技术支撑(Belabbas and Wolfe, 2009)，正如我们在这里通过SVD遇到的那样。

SVD允许我们发现与某些相同类型的信息

特征分解(eigendecomposition)。然而，SVD 对非方阵和数据表具有更广泛的适用性。每当我们想要识别数据中的异质性、通过近似进行数据压缩（例如，不存储 $n \times m$ 个值，而是存储 $(n+m)k$ 个值）或执行数据预处理（例如，去除设计矩阵预测变量的相关性）时，这些矩阵分解方法就变得相关[(Ormonde et al., 2001).]。SVD 操作矩阵，我们可以将其解释为具有两个索引（行和列）的矩形数组。将类似矩阵的结构扩展到高维数组称为张量(tensors)。事实证明，SVD 是在这些张量上操作的更通用分解族的特殊情况[(Kolda and Bader, 2009).]。张量上的类 SVD 操作和低秩近似例如有 Tucker 分解[(Tucker, 1966)] 或 CP 分解[(Carroll and Chang, 1970).]。SVD 低秩近似在机器学习中经常用于计算效率原因。这是因为它减少了我们需要在可能非常大的数据矩阵上执行的内存量和非零乘法运算[(Trefethen and Bau III, 1997).]。此外，低秩近似用于操作可能包含缺失值的矩阵，以及用于有损压缩和降维[(Moonen and De Moor, 1995; Markovsky, 2011).]。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>.] 练习 [137]

练习

4.1 使用拉普拉斯展开（使用第一行）和萨吕斯规则计算行列式

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 0 & 2 & 4 \end{bmatrix}$$

4.2 高效计算以下行列式：

$$\begin{vmatrix} 2 & 0 & 1 & 2 & 0 \\ 2 & -1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 1 & 2 \\ -2 & 0 & 2 & -1 & 2 \\ 2 & 0 & 0 & 1 & 1 \end{vmatrix}$$

4.3 计算特征空间(eigenspaces)

a. $A := \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$

b. $B := \begin{bmatrix} -2 & 2 \\ 2 & 1 \end{bmatrix}$

4.4 计算所有特征空间

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

4.5 矩阵的可对角化性与其可逆性无关。确定以下四个矩阵是否可对角化和/或可逆

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

4.6 计算以下变换矩阵的特征空间。它们是否可对角化？

a. 对于 $A = \begin{bmatrix} 2 & 3 & 0 \\ 1 & 4 & 3 \\ 0 & 0 & 1 \end{bmatrix}$

b. 对于 $A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

[138] 矩阵分解

4.7 以下矩阵是否可对角化？如果是，确定它们的对角形式和变换矩阵相对于其为对角形式的基。如果不是，说明它们不可对角化的原因。

a. $A = \begin{bmatrix} 0 & 1 \\ -8 & 4 \end{bmatrix}$

b. $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

c. $A = \begin{bmatrix} 5 & 4 & 2 & 1 \\ 0 & 1 & -1 & -1 \\ -1 & -1 & 3 & 0 \\ 1 & 1 & -1 & 2 \end{bmatrix}$

d. $A = \begin{bmatrix} 5 & -6 & -6 \\ -1 & 4 & 2 \\ 3 & -6 & -4 \end{bmatrix}$

4.8 求矩阵的 SVD

$$A = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$$

4.9 求奇异值分解

$$A = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}$$

4.10 求秩-1 近似

$$A = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$$

4.11 证明对于任何 $A \in \mathbb{R}^{m \times n}$, 矩阵 $A^T A$ 和 AA^T 具有相同的非零特征值。

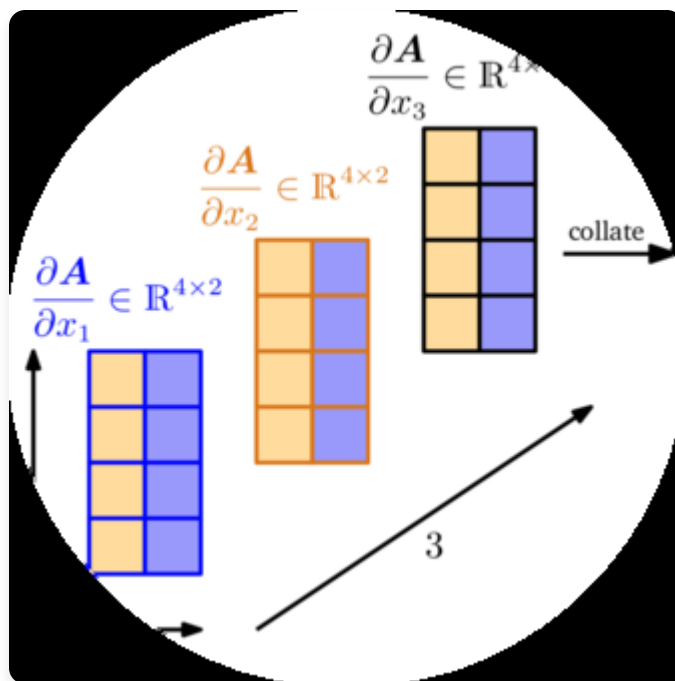
4.12 证明对于 $x \neq 0$, 定理 4.24 成立, 即证明

$$\max_x \frac{\|Ax\|_2}{\|x\|_2} = \sigma_1$$

其中 σ_1 是 $A \in \mathbb{R}^{m \times n}$ 的最大奇异值。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>.]

向量微积分



机器学习中的许多算法针对一组期望的模型参数优化目标函数，这些参数控制模型解释数据的好坏程度：寻找好的参数可以表述为优化问题（见第 8.2 和 8.3 节）。例子包括：(i) 线性回归（见第 9 章），我们研究曲线拟合问题并优化线性权重参数以最大化似然；(ii) 用于降维和数据压缩的神经网络自编码器，其中参数是每层的权重和偏置，我们通过链式法则的重复应用来最小化重构误差；以及 (iii) 用于建模数据分布的高斯混合模型（见第 11 章），我们优化每个混合成分的位置和形状参数以最大化模型的似然。图 5.1 说明了这些问题中的一些，我们通常通过使用利用梯度信息的优化算法来解决（第 7.1 节）。图 5.2 概述了本章概念如何相关以及它们如何与本书其他章节相连接。

本章的核心概念是函数。函数 f 是一个将两个量相互关联的量。在本书中，这些量通常是输入 $x \in \mathbb{R}^D$ 和目标值（函数值） $f(x)$ ，除非另有说明，我们假设它们是实值的。这里 $\mathbb{R}^D$ 是 f 的定义域，函数值 $f(x)$ 是 f 的值域/陪域。

图5.1 向量

训练数据微积分在回归（曲线拟合）和密度估计（即数据分布建模）中起着核心作用。

- a. 回归问题：找到参数，使得曲线能很好地解释观察结果（交叉点）。
- (b) 用高斯混合模型进行密度估计：找到均值和协方差，使得数据（点）能得到很好的解释。

这些材料由剑桥大学出版社出版，书名为《机器学习数学》，作者为 Marc Peter Deisenroth、A. Aldo Faisal 和 Cheng Soon Ong (2020)。此版本仅供个人查看和下载使用。不得重新分发、转售或用于衍生作品。

图5.2 本章介绍概念的思维导图，以及它们在本书其他部分的使用情况。

差商在第9章中定义，在回归中使用；偏导数在第7章的优化中使用，在第10章的降维中使用；雅可比矩阵在第6章的概率中使用，在第11章的密度估计中使用；海塞矩阵；泰勒级数在第12章的分类中使用。

第2.7.3节在线性函数的背景下提供了更详细的讨论。我们经常写作

$$f: \mathbb{R}^D \rightarrow \mathbb{R} \quad (5.1a) \quad x \mapsto f(x) \quad (5.1b)$$

来指定一个函数，其中(5.1a)指定 f 是从 $\mathbb{R}^D$ 到 $\mathbb{R}$ 的映射，(5.1b)指定将输入 x 明确分配给函数值 $f(x)$ 。函数 f 为每个输入 x 分配唯一的函数值 $f(x)$ 。

例5.1

回忆点积作为内积的特例（第3.2节）。在之前的记号中，函数 $f(x) = x^T x$ ， $x \in \mathbb{R}^2$ ，将被指定为

$$f: \mathbb{R}^2 \rightarrow \mathbb{R} \quad (5.2a) \quad x \mapsto x_1^2 + x_2^2 \quad (5.2b)$$

在本章中，我们将讨论如何计算函数的梯度，这对于促进机器学习模型中的学习通常是必要的，因为梯度指向最陡上升的方向。因此，

图5.3 函数 f 在 x_0 和 $x_0 + \delta x$ 之间的平均斜率是通过 $f(x_0)$ 和 $f(x_0 + \delta x)$ 的割线（蓝色）的斜率，由 $\delta y / \delta x$ 给出。

向量微积分是我们在机器学习中需要的基本数学工具之一。在整本书中，我们假设函数是可微的。通过一些额外的技术定义（我们在此不涉及），所呈现的许多方法可以扩展到次微分（在某些点连续但不可微的函数）。我们将在第7章中讨论有约束函数情况的扩展。

5.1 单变量函数的微分

在下面，我们简要回顾单变量函数的微分，这可能在高中数学中已经熟悉。我们从单变量函数 $y = f(x)$, $x, y \in \mathbb{R}$ 的差商开始，随后我们将用它来定义导数。

定义5.1 (差商)。差商

$$\delta y / \delta x := [f(x + \delta x) - f(x)] / \delta x \quad (5.3)$$

计算通过 f 图上两点的割线的斜率。在图5.3中，这些是 x 坐标为 x_0 和 $x_0 + \delta x$ 的点。

如果我们假设 f 是线性函数，差商也可以被认为是 f 在 x 和 $x + \delta x$ 之间的平均斜率。在 $\delta x \rightarrow 0$ 的极限下，如果 f 是可微的，我们得到 f 在 x 处的切线。切线就是 f 在 x 处的导数。

定义5.2 (导数)。更正式地，对于 $h > 0$, f 在 x 处的导数定义为极限

$$df/dx := \lim_{h \rightarrow 0} [f(x + h) - f(x)]/h \quad (5.4)$$

图5.3中的割线变成切线。

f 的导数指向 f 最陡上升的方向。

例5.2 (多项式的导数)

我们想计算 $f(x) = x^n$, $n \in \mathbb{N}$ 的导数。我们可能已经知道答案是 nx^{n-1} ，但我们想使用导数作为差商极限的定义来推导这个结果。

使用(5.4)中导数的定义，我们得到

$$df/dx = \lim_{h \rightarrow 0} [f(x + h) - f(x)]/h \quad (5.5a) = \lim_{h \rightarrow 0} [(x + h)^n - x^n]/h \quad (5.5b) = \lim_{h \rightarrow 0} [\sum_{i=0}^n \binom{n}{i} x^{(n-i)h} - x^n]/h \quad (5.5c)$$

我们看到 $\binom{n}{0} x^n = x^n h^0$ 。从 $i=1$ 开始求和， x^n 项消除，我们得到

$$df/dx = \lim_{h \rightarrow 0} [\sum_{i=1}^n \binom{n}{i} x^{(n-i)h}]/h \quad (5.6a) = \lim_{h \rightarrow 0} \sum_{i=1}^n \binom{n}{i} x^{(n-i)h} (i-1) \quad (5.6b)$$

$$[i]=[1]$$

$$n! \cdot [n] \cdot n$$

$$= \lim [n] \times [-] \cdot 1 [n] [-] [i] [i] [-] 1 x + x h \quad (5.6c)$$

$$[h] \rightarrow [0] \cdot 1 \cdot i$$

$$[i]=[2] \mid \{z\}$$

$$[\rightarrow][0] [as] [h] [\rightarrow][0]$$

$$= [n! \cdot (-1)^{n-1} (5.6d) \times n \cdot 1!(n-1)!]$$

5.1.1 Taylor级数

Taylor级数是将函数 f 表示为无穷项之和的一种方法。这些项是使用在 x_0 处求值的 f 的导数确定的。

定义5.3 (Taylor多项式)。函数 $f: \mathbb{R} \rightarrow \mathbb{R}$ 在 x_0 处的 n 次 Taylor多项式定义为

$$T_n^{f, x_0}(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

$$T_n^{f, x_0}(x) := \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k, \quad (5.7)$$

$$f^{(k)}(x_0)$$

$$k=0$$

其中 $f^{(k)}(x_0)$ 是 f 在 x_0 处的 k 次导数（我们假设它存在）， $\frac{f^{(k)}(x_0)}{k!}$ 是多项式的系数。 $f^{(k)}(x_0)$

定义5.4 (Taylor级数)。对于光滑函数 $f \in C^\infty$, $f: \mathbb{R} \rightarrow \mathbb{R}$, f 在 x_0 处的 Taylor级数定义为

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

$$T_n^{f, x_0}(x) = \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k. \quad (5.8)$$

$$f^{(k)}(x_0)$$

$$k=0$$

当 $x_0 = 0$ 时，我们得到 Maclaurin级数作为Taylor级数的特殊情况。如果 $f \in C^\infty$ （意味着 f 是无穷次连续可微的），那么 f 被称为解析的。

注释。一般来说， n 次Taylor多项式是一个函数的近似，该函数不一定是多项式。Taylor多项式在 x_0 附近的邻域内与 f 相似。然而， n 次Taylor多项式是次数 $k \leq n$ 的多项式 f 的精确表示，因为所有导数 $f^{(i)}(x_0)$, $i > n$ 都为零。◇

例5.3 (Taylor多项式)

我们考虑多项式

$$f(x) = x^4 \quad (5.9)$$

并寻求在 $x_0 = 1$ 处求值的Taylor多项式 T_6 。我们首先计算 $k = 0, \dots, 6$ 的系数 $f^{(k)}(1)$:

$$f(1) = 1 \quad (5.10)$$

$$f'(1) = 4 \quad (5.11)$$

$$f''(1) = 12 \quad (5.12)$$

$$f^{(3)}(1) = 24 \quad (5.13)$$

$$f(4) = 24 \quad (5.14)$$

$$f(5) = 0 \quad (5.15)$$

$$f(6) = 0 \quad (5.16)$$

因此，所需的Taylor多项式是

$$T_k(x) = \sum_{j=0}^k \frac{f^{(j)}(0)}{j!} x^j$$

$$T_6(x) = \sum_{j=0}^6 \frac{f^{(j)}(0)}{j!} x^j \quad (5.17a)$$

$$T_6(x) =$$

$$\sum_{j=0}^6 \frac{f^{(j)}(0)}{j!} x^j$$

$$= 1 + 4x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + \frac{1}{720}x^6 \quad (5.17b)$$

展开并重新排列得到

$$T_6(x) = (1 + 4x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + \frac{1}{720}x^6)$$

$$+ \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + \frac{1}{720}x^6 \quad (5.18a)$$

$$= x^4 = f(x), \quad (5.18b)$$

即，我们得到了原函数的精确表示。

图 5.4 Taylor 多项式。原函数 $f(x) = \sin(x) + \cos(x)$ （黑色实线）在 $x=0$ 处被 Taylor 多项式（虚线）近似。高阶 Taylor 多项式更好地全局近似函数。 T_{10} 在 $[-4, 4]$ 中已经与 f 相似。

例 5.4 (Taylor 级数)

考虑图 5.4 中给出的函数

$$f(x) = \sin(x) + \cos(x) \in C^\infty. \quad (5.19)$$

我们寻求 f 在 $x=0$ 处的 Taylor 级数展开，这是 f 的 Maclaurin 级数展开。我们得到以下导数：

$$f(0) = \sin(0) + \cos(0) = 1 \quad (5.20)$$

$$f'(0) = \cos(0) - \sin(0) = 1 \quad (5.21)$$

$$f''(0) = -\sin(0) - \cos(0) = -1 \quad (5.22)$$

$$f^{(3)}(0) = -\cos(0) + \sin(0) = -1 \quad (5.23)$$

$$f^{(4)}(0) = \sin(0) + \cos(0) = f(0) = 1 \quad (5.24)$$

...

我们可以看到一个模式：我们Taylor级数中的系数只有 ± 1 （因为 $\sin(0) = 0$ ），每个系数在切换到另一个之前出现两次。此外， $f(0) = f(0)$ 。

因此， f 在 $x[0] = 0$ 处的完整Taylor级数展开为

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k$$

$$T_n(x) = \sum_{k=0}^n \frac{f^{(k)}(0)}{k!} x^k$$

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k$$

$$= 1 + x - \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \frac{x^5}{5!} - \cdots \quad (5.25b)$$

$$= 1 + x - \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \frac{x^5}{5!} - \cdots \quad (5.25c)$$

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k$$

$$= \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k = \cos(x) + \sin(x) \quad (5.25d)$$

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k$$

$$= \cos(x) + \sin(x) \quad (5.25e)$$

其中我们使用了 *power series* 表示

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k$$

$$\cos(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} x^{2k} \quad (5.26)$$

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k$$

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k$$

$$\sin(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1} \quad (5.27)$$

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} x^k$$

图5.4显示了 $n = 0, 1, 5, 10$ 的相应前几个Taylor多项式 $T[n]$ 。

注释。Taylor级数是power series的特殊情况

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(c)}{k!} (x-c)^k$$

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(c)}{k!} (x-c)^k \quad (5.28)$$

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(c)}{k!} (x-c)^k$$

其中 $a[k]$ 是系数， c 是常数，它具有定义5.4中的特殊形式。◇

5.1.2 微分法则

在下面，我们简要说明基本的微分法则，其中我们用 f' 表示 f 的导数。

乘积法则： $[f(x)g(x)]' = f'(x)g(x) + f(x)g'(x)$ (5.29)

$[f(x)g(x)]' = f'(x)g(x) + f(x)g'(x)$

商法则： $\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - f(x)g'(x)}{g(x)^2}$ (5.30)

和法则： $[f(x) + g(x)]' = f'(x) + g'(x)$ (5.31)

链式法则： $[f(g(x))]' = f'(g(x))g'(x)$

$(f \circ g)(x) = f(g(x))$ (5.32)

这里， $g \circ f$ 表示函数复合 $x \mapsto f(x) \mapsto g(f(x))$ 。

例5.5（链式法则）

让我们使用链式法则计算函数 $h(x) = (2x + 1)^4$ 的导数。设

$h(x) = (2x + 1)^4 = g(f(x))$, (5.33) $f(x) = 2x + 1$, (5.34) $g(u) = u^4$, (5.35)

我们得到 f 和 g 的导数为

$f'(x) = 2$, (5.36) $g'(u) = 4u^3$, (5.37)

因此 h 的导数为

$h'(x) = [g(f(x))]' = g'(f(x))f'(x) = (4f^3) \cdot 2 = 8(2x + 1)^3$, (5.38)

其中我们使用了链式法则(5.32)并将 f 的定义(5.34)代入 $g'(f)$ 中。

5.2 偏微分与梯度

第5.1节讨论的微分适用于标量变量 $x \in \mathbb{R}$ 的函数 f 。接下来，我们考虑函数 f 依赖于一个或多个变量 $x \in \mathbb{R}^n$ 的一般情况，例如， $f(x) = f(x_1, x_2)$ 。导数对多变量函数的推广就是梯度。

我们通过一次改变一个变量并保持其他变量不变来找到函数 f 关于 x 的梯度。梯度就是这些偏导数的集合。

定义5.5 (偏导数)。对于函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$, $x \mapsto f(x)$, $x \in \mathbb{R}^n$ 有 n 个变量 $x_1, \dots, x_n$ ，我们定义偏导数为

$$\partial f / \partial x_1 = \lim_{h \rightarrow 0} [f(x_1 + h, x_2, \dots, x_n) - f(x)] / h$$

... (5.39)

$$\partial f / \partial x_n = \lim_{h \rightarrow 0} [f(x_1, \dots, x_{n-1}, x_n + h) - f(x)] / h$$

并将它们收集在行向量中

$$\nabla_x f = \text{grad } f = [\partial f(x) / \partial x_1 \ \partial f(x) / \partial x_2 \ \dots \ \partial f(x) / \partial x_n] \in \mathbb{R}^{1 \times n}, \quad (5.40)$$

其中 n 是变量的数量，1是 f 的像/值域/余域的维数。这里，我们定义了列向量 $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ 。方程(5.40)中的行向量称为 f 的梯度或Jacobian，是第5.1节导数的推广。

注释。 Jacobian的这个定义是向量值函数Jacobian一般定义的特例，即偏导数的集合。我们将在第5.3节回到这个话题。◇

例5.6 (使用链式法则的偏导数)

对于 $f(x, y) = (x^3 + 2y)^2$ ，我们得到偏导数

$$\partial f(x, y) / \partial x = 2(x^3 + 2y) \partial(x^3 + 2y) / \partial x = 2(x^3 + 2y) \cdot 3x^2, \quad (5.41)$$

$$\partial f(x, y) / \partial y = 2(x^3 + 2y) \partial(x^3 + 2y) / \partial y = 12(x^3 + 2y)y, \quad (5.42)$$

其中我们使用链式法则(5.32)来计算偏导数。

注释(作为行向量的梯度)。 在文献中，遵循向量通常是列向量的惯例，将梯度向量定义为列向量并不少见。我们将梯度向量定义为行向量的原因有两个：第一，我们可以一致地将梯度推广到向量值函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ (这时梯度变成矩阵)。第二，我们可以立即应用多元链式法则而不必注意梯度的维数。我们将在第5.3节讨论这两点。◇

例5.7 (梯度)

对于 $f(x_1, x_2) = x_1^2 x_2 + x_1 x_2^3$, $x \in \mathbb{R}^2$ ，偏导数(即 f 关于 x_1 和 x_2 的导数)为

$$\partial f(x_1, x_2) / \partial x_1 = 2x_1 x_2 + x_2^3 \quad (5.43)$$

$$\partial f(x_1, x_2) / \partial x_2 = x_1^2 + 3x_1 x_2^2 \quad (5.44)$$

梯度为

$$df/dx = [\partial f(x_1, x_2)/\partial x_1 \quad \partial f(x_1, x_2)/\partial x_2] = [2x_1x_2 + x_2^3 \quad x_1^2 + 3x_1x_2^2] \in \mathbb{R}^{1 \times 2}, (5.45)$$

5.2.1 偏微分的基本规则

在多元情况下，其中 $x \in \mathbb{R}^n$ ，我们从学校了解的基本微分规则(例如，和规则、积规则、链式法则；另见第5.1.2节)仍然适用。然而，当我们计算关于向量 $x \in \mathbb{R}^n$ 的导数时，我们需要注意：我们的梯度现在涉及向量和矩阵，而矩阵乘法不满足交换律(第2.2.1节)，即顺序很重要。

以下是一般的积规则、和规则和链式法则：

$$\text{积规则：} \partial/\partial x [f(x)g(x)] = g(x)\partial f/\partial x + f(x)\partial g/\partial x \quad (5.46)$$

$$\text{和规则：} \partial/\partial x [f(x) + g(x)] = \partial f/\partial x + \partial g/\partial x \quad (5.47)$$

$$\text{链式法则：} \partial/\partial x [(g \circ f)(x)] = \partial/\partial x [g(f(x))] = \partial g/\partial f \cdot \partial f/\partial x \quad (5.48)$$

让我们仔细看看链式法则。链式法则(5.48)在某种程度上类似于矩阵乘法的规则，我们说邻接维数必须匹配才能定义矩阵乘法；见第2.2.1节。如果我们从左到右，链式法则表现出类似的性质： ∂f 出现在第一个因子的”分母”中和第二个因子的”分子”中。如果我们将因子相乘，乘法被定义，即 ∂f 的维数匹配， ∂f ”约掉”，使得 $\partial g/\partial x$ 保留。

5.2.2 链式法则

考虑一个二变量函数 $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ ，变量为 x_1, x_2 。此外， $x_1(t)$ 和 $x_2(t)$ 本身是 t 的函数。要计算

关于 t 的 f 导数，我们需要应用多元函数的链式法则[(5.48)]：

$$d/dt = \partial/\partial x_1 [dx_1/dt] + \partial/\partial x_2 [dx_2/dt] \quad (5.49)$$

$$d/dt = [\partial f/\partial x_1] [dx_1/dt] + [\partial f/\partial x_2] [dx_2/dt] \quad (5.49)$$

其中 d 表示梯度， ∂ 表示偏导数。

例5.8

考虑 $f(x, y) = x + 2y$ ，其中 $x = \sin t$ 和 $y = \cos t$ ，则

$$d/dt = \partial/\partial x + 2\partial/\partial y$$

$$d/dt = \partial/\partial x + 2\partial/\partial y$$

$$= \partial/\partial x + 2\partial/\partial y$$

$$d/dt = \partial/\partial x + 2\partial/\partial y$$

$$d/dt = \partial/\partial x + 2\partial/\partial y$$

$$= 2 \sin t + 2 \quad (5.50b)$$

$$\frac{\partial f}{\partial t}$$

$$= 2 \sin t \cos t - 2 \sin t = 2 \sin t (\cos t - 1) \quad [(5.50c)]$$

这是f关于t的相应导数。

如果f(x1, x2)是x1和x2的函数，其中x1和x2本身是两个变量s和t的函数，则链式法则产生偏导数

$$\frac{\partial f}{\partial s} \frac{\partial f}{\partial t} \frac{\partial x_1}{\partial s} \frac{\partial x_2}{\partial s}$$

$$= +, \quad (5.51)$$

$$\frac{\partial f}{\partial s} \frac{\partial x_1}{\partial s} \frac{\partial x_2}{\partial s} \frac{\partial f}{\partial x_1} \frac{\partial f}{\partial x_2}$$

$$= +, \quad (5.52)$$

$$\frac{\partial f}{\partial t} \frac{\partial x_1}{\partial t} \frac{\partial x_2}{\partial t}$$

[《Machine Learning数学基础》草稿(2023-10-18)。反馈：] [https://mml-book.com][.] [5.3 向量值函数的梯度] [149]

梯度通过矩阵乘法获得

$$\frac{\partial f}{\partial x} \frac{\partial x}{\partial t}$$

$$d \left(\frac{\partial f}{\partial x} \frac{\partial x}{\partial t} \right) = \frac{\partial f}{\partial x} \frac{\partial x}{\partial t} \quad (5.53) \quad \frac{\partial f}{\partial x} \frac{\partial x}{\partial t} \quad \frac{\partial f}{\partial x} \frac{\partial x}{\partial t} \quad \frac{\partial f}{\partial x} \frac{\partial x}{\partial t}$$

$$\left| \frac{\partial f}{\partial x} \frac{\partial x}{\partial t} \right|$$

$$\left| \frac{\partial f}{\partial x} \frac{\partial x}{\partial t} \right|$$

$$\left| \frac{\partial f}{\partial x} \frac{\partial x}{\partial t} \right|$$

这种将链式法则写成矩阵乘法的紧凑方式只有在梯度定义为行向量时才有意义。否则，我们需要开始转置梯度以使矩阵维度匹配。[[链式法则可以]]

只要梯度是向量或矩阵，这仍然可能很简单；然而，当梯度变成张量(tensor)时（我们将在下文讨论），转置就不再是小事了。[[写成矩阵乘法。]]

备注（验证梯度实现的正确性）。偏导数作为相应差商极限的定义（见[(5.39)]）可以在数值检查计算机程序中梯度的正确性时加以利用：当我们计算[[梯度检查]]

梯度并实现它们时，我们可以使用有限差分来数值测试我们的计算和实现：我们选择h值很小（例如，[-][4] h = 10），并将[(5.39)]中的有限差分近似与我们的（解析）梯度实现进行比较。如果误差很小，我们的梯度实现可能是正确的。“小”

可能意味着q[P][2][][dh][+][df][i][i][i][i] - [P][2] < 10，其中dh[i]是有限差分[][dh][df][i][i][+][i][i]

近似， $df[i]$ 是 f 关于第 i 个变量 $x[i]$ 的解析梯度。◇

5.3 向量值函数的梯度

到目前为止，我们讨论了映射到实数的函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 的偏导数和梯度。接下来，我们将梯度概念推广到向量值函数（向量场） $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ ，其中 $n \geq 1$ 且 $m > 1$ 。

对于函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 和向量 $x = [x_1, \dots, x_n] \in \mathbb{R}^n$ 相应的函数值向量为

$$f(x)$$

$$= [f_1(x), \dots, f_m(x)] \in \mathbb{R}^m. \quad (5.54)$$

$$f(x) = [f_1(x), \dots, f_m(x)] \in \mathbb{R}^m. \quad (5.54)$$

$$f(x) = [f_1(x), \dots, f_m(x)] \in \mathbb{R}^m$$

以这种方式写向量值函数允许我们将向量值函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 视为函数向量 $[f_1, \dots, f_m]$ ，

每个 $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$ 映射到 $\mathbb{R}$ 。每个 f_i 的微分规则 [5.2.1]

正是我们在第 5.2 节中讨论的那些。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由 Cambridge University Press 出版 (2020).]

[150] [向量微积分]

因此，向量值函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$

$\mathbb{R}^m$ 关于 $x \in \mathbb{R}^n$, $i = 1, \dots, n$ 的偏导数，给出为向量

$$\frac{\partial f}{\partial x_i}(x) = \lim_{h \rightarrow 0} \frac{f(x + h e_i) - f(x)}{h} \in \mathbb{R}^m$$

$$\frac{\partial f}{\partial x_i}(x) = [\frac{\partial f_1}{\partial x_i}(x), \dots, \frac{\partial f_m}{\partial x_i}(x)] \in \mathbb{R}^m$$

$$\frac{\partial f}{\partial x_i}(x) = [\frac{\partial f_1}{\partial x_i}(x), \dots, \frac{\partial f_m}{\partial x_i}(x)] \in \mathbb{R}^m$$

$$\frac{\partial f}{\partial x_i}(x) = [\frac{\partial f_1}{\partial x_i}(x), \dots, \frac{\partial f_m}{\partial x_i}(x)] \in \mathbb{R}^m$$

$$\frac{\partial f}{\partial x_i}(x) = [\frac{\partial f_1}{\partial x_i}(x), \dots, \frac{\partial f_m}{\partial x_i}(x)] \in \mathbb{R}^m$$

$$(5.55)$$

从 (5.40)，我们知道 f 关于向量的梯度是偏导数的行向量。在 (5.55) 中，每个偏导数

$\partial f / \partial x_i$ 本身是列向量。因此，我们通过收集这些偏导数获得 f ：

$\mathbb{R}^n \rightarrow \mathbb{R}^m$ 关于 $x \in \mathbb{R}^n$ 的梯度：

$$[d \quad]f(x)$$

$$[dx \quad \partial x \quad [\cdot \cdot \cdot]] = [\partial] [f] (l) [\partial][f] (l) \quad (5.56a)$$

$$[\partial x][n]$$

$$[] []$$

$$\partial f(l) \quad \partial f(l) [\cdot \cdot \cdot]$$

$$[] [\partial x] \quad 1 \quad [\partial x][n] []$$

$$= [] [] [m] [\times][n] [...] [...] [] [] \mathbf{R} \in \cdot \quad (5.56b) [] []$$

$$[] [\partial f][m](l) [\partial f][m](l) [] [\cdot \cdot \cdot]$$

$$[\partial x] \quad 1 \quad [\partial x][n]$$

定义5.6 (Jacobian)。向量值函数 $f: \mathbf{R}[n] \rightarrow \mathbf{R}[m]$ 的所有一阶偏导[Jacobian] [n] [m] 数的集合称为 *Jacobian*。Jacobian J是一个 $m \times n$ 矩阵，我们定义并排列如下：[函数的梯度]

$$[n] [m] \quad df(x) \quad \partial f(x) \quad \partial f(x)$$

$$[f] [:] [\mathbf{R}] [\rightarrow] [\mathbf{R}] [\text{是一个}] J = \nabla f = [\cdot \cdot \cdot] \quad (5.57) [\text{大小为}] x \quad dx \quad \partial x$$

$$1 [n]$$

$$[m] [\times] [n] \text{的矩阵。} \quad \partial f(x) \quad \partial f(x)$$

$$1 \quad 1 \quad \cdot \cdot \cdot$$

$$\partial x \quad \partial x \quad ..$$

$$= .. \quad 1 [n]$$

$$.. [, [(5.58)]]$$

$$\partial f(x) \quad \partial f(x) [m] [m] \cdot \cdot \cdot$$

$$\partial x \quad 1 \quad \partial x[n]$$

$$x$$

$$x = [i] .. \quad 1 \quad \partial f$$

$$\partial x \quad , \quad J(i, j) = \cdot \quad (5.59)$$

$$x [j]$$

$$[n]$$

作为[(5.58)]的特殊情况，函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ，将

向量 $\mathbf{x} \in \mathbb{R}^n$ 映射到标量（例如， $f(\mathbf{x}) = x_1$ ），拥有一个 Jacobian $\mathbf{x} \cdot \mathbf{e}_1 = x_1$

即行向量（维度为 $1 \times n$ 的矩阵）；见[(5.40)] [分子布局] 备注。在本书中，我们使用导数的分子布局，即

$f \in \mathbb{R}^m$ 关于 $\mathbf{x} \in \mathbb{R}^n$ 的导数 $df/d\mathbf{x}$ 是一个 $m \times$

n 矩阵，其中 f 的元素定义行， $\mathbf{x}$ 的元素定义对应 Jacobian 的列；参见 (5.58)。

图 5.5 $f(\cdot)$ 的 Jacobian 行列式 b 可以用来计算 $2c_1c_2b$ 蓝色区域和橙色区域之间的 1 放大倍数。

还存在分母布局(denominator layout)，它是分子布局的转置。在本书中，我们将使用分子布局。◇

我们将在第 6.7 节中看到如何在概率分布的变量变换方法中使用 Jacobian。变换引起的缩放量由行列式提供。

在第 4.1 节中，我们看到行列式可以用来计算平行四边形的面积。如果给定两个向量 $\mathbf{b}_1^T = [1, 0]$ ， $\mathbf{b}_2^T = [0, 1]$ 作为单位正方形的边（蓝色；见图 5.5），则该正方形的面积为

$$\det \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = 1. \quad (5.60)$$

如果我们取一个边为 $\mathbf{c}_1^T = [-2, 1]$ ， $\mathbf{c}_2^T = [1, 1]$ 的平行四边形（图 5.5 中的橙色），其面积由行列式的绝对值给出（见第 4.1 节）

$$\det \begin{vmatrix} -2 & 1 \\ 1 & 1 \end{vmatrix} = |-3| = 3, \quad (5.61)$$

即，该面积恰好是单位正方形面积的三倍。我们可以通过找到一个将单位正方形变换为另一个正方形的映射来找到这个缩放因子。在线性代数术语中，我们实际上执行了从 $(\mathbf{b}_1, \mathbf{b}_2)$ 到 $(\mathbf{c}_1, \mathbf{c}_2)$ 的变量变换。在我们的情况下，映射是线性的，该映射行列式的绝对值给出了我们要寻找的缩放因子。

我们将描述两种识别该映射的方法。首先，我们利用映射是线性的这一事实，因此可以使用第 2 章的工具来识别该映射。其次，我们将使用本章讨论的工具，通过偏导数找到映射。

方法 1 要开始线性代数方法，我们将 $\{\mathbf{b}_1, \mathbf{b}_2\}$ 和 $\{\mathbf{c}_1, \mathbf{c}_2\}$ 都识别为 $\mathbb{R}^2$ 的基（参见第 2.6.1 节的回顾）。我们实际执行的是从 $(\mathbf{b}_1, \mathbf{b}_2)$ 到 $(\mathbf{c}_1, \mathbf{c}_2)$ 的基变换，我们正在寻找实现基变换的变换矩阵。使用第 2.7.2 节的结果，我们将所需的基变换矩阵识别为

$$\mathbf{J} = \begin{vmatrix} -2 & 1 \\ 1 & 1 \end{vmatrix}, \quad (5.62)$$

使得 $\mathbf{J}\mathbf{b}_1 = \mathbf{c}_1$ 和 $\mathbf{J}\mathbf{b}_2 = \mathbf{c}_2$ 。 $\mathbf{J}$ 的行列式的绝对值，即我们要寻找的缩放因子，为

$|\det(\mathbf{J})| = 3$ ，即由 $(\mathbf{c}_1, \mathbf{c}_2)$ 张成的正方形面积是由 $(\mathbf{b}_1, \mathbf{b}_2)$ 张成的面积的三倍。

方法 2 线性代数方法适用于线性变换；对于非线性变换（在第 6.7 节中变得相关），我们采用使用偏导数的更一般方法。

对于这种方法，我们考虑一个执行变量变换的函数 $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ 。在我们的例子中， f 将任何向量 $\mathbf{x} \in \mathbb{R}^2$ 相对于 $(\mathbf{b}_1, \mathbf{b}_2)$ 的坐标表示映射到相对于 $(\mathbf{c}_1, \mathbf{c}_2)$ 的坐标表示 $\mathbf{y} \in \mathbb{R}^2$ 。我们想要识别映射，以便计算当区域（或体积）被 f 变换时如

何变化。为此，我们需要找出如果稍微修改 x , $f(x)$ 如何变化。这个问题正好由 Jacobian 矩阵 $\partial f / \partial x \in \mathbb{R}^{2 \times 2}$ 回答。由于我们可以写成

$$y_1 = -2x_1 + x_2 \quad (5.63) \quad y_2 = x_1 + x_2 \quad (5.64)$$

我们得到 x 和 y 之间的函数关系，这允许我们获得偏导数

$$\partial y_1 / \partial x_1 = -2, \partial y_1 / \partial x_2 = 1, \partial y_2 / \partial x_1 = 1, \partial y_2 / \partial x_2 = 1 \quad (5.65)$$

并组成 Jacobian 为

$$J = \begin{bmatrix} \partial y_1 / \partial x_1 & \partial y_1 / \partial x_2 \\ \partial y_2 / \partial x_1 & \partial y_2 / \partial x_2 \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1 & 1 \end{bmatrix}. \quad (5.66)$$

从几何上讲，Jacobian 表示我们要寻找的坐标变换。如果坐标变换是线性的（如我们的情况），它是精确的，(5.66) 正好恢复了 (5.62) 中的基变换矩阵。如果坐标变换是非线性的，Jacobian 用线性变换局部近似这种非线性变换。*Jacobian* 行列式的绝对值 $|\det(J)|$ 是坐标变换时面积或体积缩放的因子。我们的情况产生 $|\det(J)| = 3$ 。

Jacobian 行列式和变量变换将在第 6.7 节中变得相关，当我们变换随机变量和概率分布时。这些变换在机器学习中极其相关，特别是在使用 *重参数化技巧*(reparametrization trick)（也称为 *无限扰动分析*(infinite perturbation analysis)）训练深度神经网络的背景下。

在本章中，我们遇到了函数的导数。图 5.6 总结了这些导数的维度。如果 $f: \mathbb{R} \rightarrow \mathbb{R}$ ，梯度只是一个标量（左上角条目）。对于 $f: \mathbb{R}^D \rightarrow \mathbb{R}$ ，梯度是一个 $1 \times D$ 行向量（右上角条目）。对于 $f: \mathbb{R} \rightarrow \mathbb{R}^E$ ，梯度是一个 $E \times 1$ 列向量，对于 $f: \mathbb{R}^D \rightarrow \mathbb{R}^E$ ，梯度是一个 $E \times D$ 矩阵。

图 5.6 （偏）导数的维度。 $x f(x) \partial f / \partial x$

例 5.9 (向量值函数的梯度)

给定

$$f: \mathbb{R}^M \rightarrow \mathbb{R}^N, f(x) = Ax, A \in \mathbb{R}^{N \times M}, x \in \mathbb{R}^M.$$

为了计算梯度 df/dx ，我们首先确定 $d[N] \times [M] df/dx$ 的维度：由于 $f: \mathbb{R}^M \rightarrow \mathbb{R}^N$ ，因此 $df/dx \in \mathbb{R}^{N \times M}$ 。其次，为了计算梯度，我们确定 f 对每个 $x[j]$ 的偏导数：

$$[N] \partial f \Rightarrow$$

$$f \times [i] (x) = A x = A \quad (5.67)$$

$$[i] [j] [i] \partial x [j]$$

$$[j]=[1]$$

我们将偏导数收集到 Jacobian 中，得到梯度

$$\partial f \partial x = A \cdot \cdot \cdot \cdot \cdot$$

$$d \cdot [A \in \mathbb{R}^{N \times M}] \cdot x \cdot [\partial f] [M] d \partial x [\partial x] [1] 1 [N] [N] f = [M] \cdot \cdot \cdot \cdot \cdot [\times] [N] =$$

$$[\partial x] [M] 1 1 [\partial x] [M N] [N] \cdot \cdot \cdot A \cdot \cdot \cdot [\partial f][M] A$$

例 5.10 (链式法则)

考虑函数 $h: \mathbb{R} \rightarrow \mathbb{R}$, $h(t) = (f \circ g)(t)$, 其中

$$f: \mathbb{R} \rightarrow \mathbb{R} \quad (5.69) \quad g: \mathbb{R} \rightarrow \mathbb{R} \quad (5.70)$$

$$f(x) = \exp(x^2), \quad (5.71)$$

$$g(t) =$$

$$t \cos t$$

$$x(t) = g(t) = \quad (5.72)$$

$$x(t) = t \sin t$$

并计算 h 对 t 的梯度。由于 $f: \mathbb{R} \rightarrow \mathbb{R}$ 且 $g: \mathbb{R} \rightarrow \mathbb{R}$, 我们注意到

$$\frac{\partial}{\partial t} f(g(t)) = \frac{\partial f}{\partial x} [g(t)] \frac{dg}{dt}(t), \quad x \in \mathbb{R}, \quad t \in \mathbb{R}. \quad (5.73)$$

通过应用链式法则计算所需的梯度:

$$\frac{\partial}{\partial x}$$

$$1$$

$$\frac{dh}{dt} = \frac{\partial f}{\partial x} \frac{dg}{dt} = \frac{\partial f}{\partial x} \frac{d}{dt} g(t) = \frac{\partial f}{\partial x} \frac{d}{dt} (t \sin t) \quad (5.74a)$$

$$\frac{\partial}{\partial t}$$

$$\cos t - t \sin t$$

$$= \frac{\partial f}{\partial x} (g(t)) \frac{dg}{dt}(t) = \frac{\partial f}{\partial x} (t \sin t) \frac{d}{dt} (t \sin t) \quad (5.74b)$$

$$= \frac{\partial f}{\partial x} (t \sin t) \frac{d}{dt} (t \sin t) = \frac{\partial f}{\partial x} (t \sin t) (\sin t + t \cos t)$$

$$= \frac{\partial f}{\partial x} (t \sin t) (\sin t + t \cos t) = 2 t \sin t \cos t + t^2 \cos^2 t, \quad (5.74c)$$

其中 $x(t) = t \cos t$ 且 $x'(t) = t \sin t$; 见 [(5.72).]

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

向量微积分

例 5.11 (线性模型中最小二乘损失的梯度)

让我们考虑线性模型

$$[(5.75) \ y = \Phi \theta,]$$

其中 $\theta \in [\mathbf{D}] \mathbf{R}$ 是参数向量, $\Phi \in \mathbf{R} [\mathbf{N}] [\times]$ 是输入特征, $y \in \mathbf{R} [\mathbf{N}]$ 是相应的观测值。我们定义函数

$$[\text{loss}] [, [(5.76)] [\mathbf{L}] [\mathbf{2}] \mathbf{L}(\mathbf{e}) := \|\mathbf{e}\|]$$

$$\mathbf{e}(\theta) := y - \Phi \theta. (5.77)$$

我们寻求 $[\partial \mathbf{L}]$, 为此我们将使用链式法则。 $\mathbf{L}$ 被称为

$$[\partial][\theta]$$

最小二乘损失函数。

在开始计算之前, 我们确定梯度的维度为

$$\partial \mathbf{L} [\mathbf{1}] [\times] [\mathbf{D}] \cdot \in \mathbf{R} (5.78) \partial \theta$$

链式法则允许我们计算梯度为

$$\partial \mathbf{L} \partial \mathbf{L} \partial \mathbf{e}$$

$$=, (5.79)$$

$$\partial \theta \partial \mathbf{e} \partial \theta$$

其中第 d 个元素由以下给出

$$[\mathbf{N}] \partial \mathbf{L} \partial \mathbf{L} \partial \mathbf{e} [\mathbf{1} \times, d] = [\mathbf{n}] [\mathbf{n}, d]. (5.80)$$

$$\partial \theta \partial \mathbf{e} \partial \theta [\mathbf{n}] [=1]$$

我们知道 $[\mathbf{2}] [\tau] \|\mathbf{e}\| = \mathbf{e} \mathbf{e}$ (见第 [3.2] 节) 并确定

$$\partial \mathbf{L} [\tau] [\mathbf{1}] [\times] [\mathbf{N}] (5.81) = 2 \mathbf{e} \in \mathbf{R} \cdot \partial \mathbf{e}$$

此外, 我们得到

$$\partial \mathbf{e} [\mathbf{N}] [\times] [\mathbf{D}] = - \Phi \in \mathbf{R}, (5.82) \partial \theta$$

因此我们所需的导数是

$$\partial \mathbf{L} [(5.77)] [\tau] [\tau] [\tau] [\tau] [\mathbf{1}] [\times] [\mathbf{D}] = - 2 \mathbf{e} \Phi = - 2(y - \theta \Phi) \Phi \in \mathbf{R}. (5.83) \partial \theta | \{z\} | \{z\} [\mathbf{N}] [\times] [\mathbf{D}] \mathbf{1} [\times] [\mathbf{N}]$$

备注。我们可以在不使用链式法则的情况下获得相同的结果, 直接查看函数

$$\mathbf{L} [\mathbf{2}] [\tau] (\theta) := \|\mathbf{y} - \Phi \theta\|^2 = (\mathbf{y} - \Phi \theta)(\mathbf{y} - \Phi \theta). (5.84) [\mathbf{2}]$$

这种方法对于像 $\mathbf{L}[\mathbf{2}]$ 这样的简单函数仍然实用, 但对于深度函数组合则变得不实用。◇

5.4 矩阵的梯度

我们会遇到需要计算矩阵对向量(或其他矩阵)的梯度的情况,这会导致多维张量。我们可以将这个张量看作是收集偏导数的多维数组。例如,如果我们计算 $m \times n$ 矩阵 A 对 $p \times q$ 矩阵 B 的梯度,结果 Jacobian 将是 $(m \times n) \times (p \times q)$,即四维张量 J ,其条目由 $J[ijkl] = \partial A[ij] / \partial B[kl]$ 给出。

由于矩阵表示线性映射,我们可以利用这一事实

存在一个向量空间同构(线性可逆映射),在

$m \times n$ 矩阵的空间 $R^{(m \times n)}$ 和 mn 维向量的空间 R^{mn} 之间。

因此,我们可以将矩阵重新形状为长度为 mn 和 pq 的向量。使用这些 mn 维向量的梯度会产生一个大小为 $mn \times pq$ 的 Jacobian 矩阵。图 5.7 可视化了这两种方法。在实际应用中,通常希望将矩阵重新形状为向量并继续使用这个 Jacobian 矩阵:链式法则(5.48)简化为简单的矩阵乘法,而在 Jacobian 张量的情况下,我们需要更加注意需要求和的维度。

例 5.12 (向量相对于矩阵的梯度)

让我们考虑以下例子,其中

$$f = Ax, f \in R^M, A \in R^{(M \times N)}, x \in R^N \quad (5.85)$$

我们寻求梯度 df/dA 。让我们先确定梯度的维度:

$$df/dA \in R^{(M \times (M \times N))} \quad (5.86)$$

根据定义,梯度是偏导数的集合:

$$df/dA = [\partial f_1 / \partial A, \dots, \partial f_i / \partial A, \dots, \partial f_M / \partial A] \in R^{(1 \times (M \times N))} \quad (5.87)$$

为了计算偏导数,显式写出矩阵向量乘法会很有帮助:

$$f_i = \sum_{j=1 \text{ to } N} A_{ij} x_j, i = 1, \dots, M \quad (5.88)$$

偏导数为:

$$\partial f_i / \partial A_{iq} = x_q \quad (5.89)$$

这使我们能够计算 f_i 相对于 A 的一行的偏导数:

$$\partial f_i / \partial A_{i,:} = x^T \in R^{(1 \times 1 \times N)} \quad (5.90)$$

$$\partial f_i / \partial A_{k \neq i,:} = 0 \in R^{(1 \times 1 \times N)} \quad (5.91)$$

其中我们必须注意正确的维度。由于 f_i 映射到 R ， A 的每一行大小为 $1 \times N$ ，我们得到一个 $1 \times 1 \times N$ 大小的张量作为 f_i 相对于 A 的一行的偏导数。

我们堆叠偏导数 (5.91) 并通过以下方式得到 (5.87) 中的期望梯度：

$$\partial f / \partial A = [x^T, 0, \dots, 0, x^T, 0, \dots, 0]^T \in R^{(1 \times (M \times N))} \quad (5.92)$$

例 5.13（矩阵相对于矩阵的梯度）

考虑矩阵 $R \in R^{(M \times N)}$ 和 $f: R^{(M \times N)} \rightarrow R^{(N \times N)}$ ，其中

$$f(R) = R^T R =: K \in R^{(N \times N)} \quad (5.93)$$

我们寻求梯度 dK/dR 。

为了解决这个难题，让我们首先写出我们已知的：梯度的维度为

$$dK/dR \in R^{((N \times N) \times (M \times N))} \quad (5.94)$$

这是一个张量。此外，

$$dK_{pq}/dR \in R^{(1 \times M \times N)} \quad (5.95)$$

对于 $p, q = 1, \dots, N$ ，其中 K_{pq} 是 $K = f(R)$ 的第 (p, q) 个元素。用 r_i 表示 R 的第 i 列， K 的每个元素由 R 的两列的点积给出：

$$K_{pq} = r_p^T r_q = \sum_{m=1}^M R_{mp} R_{mq} \quad (5.96)$$

当我们计算偏导数 $\partial K_{pq} / \partial R_{ij}$ 时，我们得到：

$$\partial K_{pq} / \partial R_{ij} = \partial / \partial R_{ij} \sum_{m=1}^M R_{mp} R_{mq} = \partial_{pqij} \quad (5.97)$$

$$\partial_{pqij} = \begin{cases} R_{iq} & \text{如果 } j = p, p \neq q \\ R_{ip} & \text{如果 } j = q, p \neq q \\ 2R_{iq} & \text{如果 } j = p, p = q \\ 0 & \text{其他情况} \end{cases} \quad (5.98)$$

从 (5.94) 我们知道期望的梯度维度为 $(N \times N) \times (M \times N)$ ，这个张量的每个元素由 (5.98) 中的 ∂_{pqij} 给出，其中 $p, q, j = 1, \dots, N$ 和 $i = 1, \dots, M$ 。

5.5 计算梯度的有用恒等式

以下列出了在机器学习中经常需要的一些有用梯度 (Petersen and Pedersen, 2012)。这里我们使用 $\text{tr}(\cdot)$ 表示迹 (见定义 4.4) , $\det(\cdot)$ 表示行列式 (见第 4.1 节) , $f(X)^{-1}$ 表示 $f(X)$ 的逆矩阵, 假设它存在。

$$\partial f(X)^T / \partial X = (\partial f(X) / \partial X)^T \quad (5.99)$$

$$\partial \text{tr}(f(X)) / \partial X = \text{tr}(\partial f(X) / \partial X) \quad (5.100)$$

$$\partial \det(f(X)) / \partial X = \det(f(X)) \text{tr}(f(X)^{-1} \partial f(X) / \partial X) \quad (5.101)$$

$$\partial f(X)^{-1} / \partial X = -f(X)^{-1} \partial f(X) / \partial X f(X)^{-1} \quad (5.102)$$

$$\partial a^T X b / \partial X = -(X^{(-1)})^T a b^T (X^{(-1)})^T \quad (5.103)$$

$$\partial x^T a / \partial x = a^T \quad (5.104)$$

$$\partial a^T x / \partial x = a^T \quad (5.105)$$

$$\partial a^T X b / \partial X = a b^T \quad (5.106)$$

$$\partial x^T B x / \partial x = x^T (B + B^T) \quad (5.107)$$

$$\partial (x - A s)^T W (x - A s) / \partial s = -2(x - A s)^T W A \quad (\text{对于对称 } W) \quad (5.108)$$

备注。在本书中, 我们只涵盖矩阵的迹和转置。然而, 我们已经看到导数可以是高维张量, 在这种情况下, 通常的迹和转置是未定义的。在这些情况下, $D \times D \times E \times F$ 张量的迹将是一个 $E \times F$ 维矩阵。这是张量收缩的特殊情况。类似地, 当我们“转置”张量时, 我们指的是交换前两个维度。具体地, 在 (5.99) 到 (5.102) 中, 当我们处理多变量函数 $f(\cdot)$ 并计算相对于矩阵的导数时 (选择不像第 5.4 节讨论的那样向量化它们), 我们需要与张量相关的计算。◇

5.6 反向传播和自动微分

在许多机器学习应用中，我们通过执行梯度下降（第 7.1 节）来找到好的模型参数，这依赖于我们能够计算学习目标相对于

对于给定的目标函数，我们可以使用微积分和链式法则来获得关于模型参数的梯度；参见第 5.2.2 节。我们在第 5.3 节中已经体验过，当时我们研究了平方损失相对于线性回归模型参数的梯度。

考虑函数

$$f(x) = 2x^2 + \exp(2x) + \cos x + \exp(x^2) \quad (5.109)$$

通过应用链式法则，并注意到微分是线性的，我们计算梯度

$$df/dx = 4x + 2\exp(2x) - \sin x + \exp(x^2)2x = 4x + 2\exp(2x) - \sin x + \exp(x^2)(1 + 2x)/(2\sqrt{(x^2 + \exp(x^2))}) \quad (5.110)$$

以这种明确的方式写出梯度通常是不实际的，因为它经常导致导数的非常冗长的表达式。在实践中，这意味着如果我们不小心，梯度的实现可能比计算函数本身昂贵得多，这会造成不必要的开销。对于训练深度神经网络模型，**backpropagation** 算法 (Kelley, 1960; Bryson, 1961; Dreyfus, 1962; Rumelhart et al., 1986) 是计算误差函数相对于模型参数梯度的有效方法。

5.6.1 深度网络中的梯度

链式法则被极度使用的一个领域是深度学习，其中函数值 y 被计算为多层函数复合

$$y = (f^K \circ f^{(K-1)} \circ \dots \circ f^1)(x) = f^K(f^{(K-1)}(\dots(f^1(x)) \dots)) \quad (5.111)$$

其中 x 是输入(例如，图像)， y 是观测值(例如，类标签)，每个函数 $f^i, i = 1, \dots, K$ ，都有自己的参数。

图5.8 多层神经网络中的前向传播，用于计算损失 L 作为输入 x 和参数 A_i, b_i 的函数

在具有多层的神经网络中，我们有函数 $f^i(x_{(i-1)}) = \sigma(A^{(i)}x_{(i-1)} + b^{(i)})$ 在第 i 层。这里 $x_{(i-1)}$ 是第 $i-1$ 层的输出， σ 是激活函数，如logistic sigmoid $1/(1+e^{-(x)})$ 、tanh或rectified linear unit (ReLU)。为了训练这些模型，我们需要损失函数 L 相对于所有模型参数 A_j, b_j ($j = 1, \dots, K$)的梯度。这也需要我们计算 L 相对于每层输入的梯度。例如，如果我们有输入 x 和观测值 y ，以及由以下定义的网络结构

$$f^0 := x \quad (5.112) \quad f^i := \sigma_i(A^{(i)}f^{(i-1)} + b^{(i)}), i = 1, \dots, K \quad (5.113)$$

参见图5.8的可视化，我们可能有兴趣找到 A_j, b_j ($j = 0, \dots, K-1$)，使得平方损失

$$L(\theta) = \|y - f^K(\theta, x)\|^2 \quad (5.114)$$

最小化，其中 $\theta = \{A^0, b^0, \dots, A^{(K-1)}, b^{(K-1)}\}$ 。

为了获得相对于参数集 θ 的梯度，我们需要 L 相对于每层 $j = 0, \dots, K-1$ 的参数 $\theta_j = \{A_j, b_j\}$ 的偏导数。链式法则允许我们确定偏导数为

$$\partial L / \partial \theta^{(K-1)} = \partial L / \partial f^K \times \partial f^K / \partial \theta^{(K-1)} \quad (5.115)$$

$$\partial L / \partial \theta^{(K-2)} = \partial L / \partial f^K \times \partial f^K / \partial f^{(K-1)} \times \partial f^{(K-1)} / \partial \theta^{(K-2)} \quad (5.116)$$

$$\partial L / \partial \theta^{(K-3)} = \partial L / \partial f^K \times \partial f^K / \partial f^{(K-1)} \times \partial f^{(K-1)} / \partial f^{(K-2)} \times \partial f^{(K-2)} / \partial \theta^{(K-3)} \quad (5.117)$$

$$\partial L / \partial \theta_j = \partial L / \partial f^K \times \partial f^K / \partial f^{(K-1)} \times \dots \times \partial f^{(i+1)} / \partial \theta_j \quad (5.118)$$

橙色项是一层输出相对于其输入的偏导数，而**蓝色**项是一层输出相对于其参数的偏导数。假设我们已经计算了偏导数 $\partial L / \partial \theta^{(i+1)}$ ，那么大部分计算可以重用来计算 $\partial L / \partial \theta_i$ 。我们需要计算的额外项由框中表示。图5.9可视化了梯度通过网络向后传播。

图5.9 多层神经网络中的反向传播，用于计算损失函数的梯度

图5.10 简单图示说明了数据从 x 通过一些中间变量 a, b 流向 y 的过程

5.6.2 自动微分(Automatic Differentiation)

事实证明，backpropagation是数值分析中称为**automatic differentiation**的通用技术的特例。我们可以将automatic differentiation视为一组数值技术(与

符号化地)通过处理中间变量并应用链式法则来评估函数的精确(在机器精度范围内)梯度。自动微分应用一系列基本算术运算，例如加法和乘法以及基本函数，例如 sin、cos、exp、log。通过对这些运算应用链式法则，可以自动计算相当复杂函数的梯度。

自动微分适用于一般的计算机程序，具有前向和反向模式。Baydin等人(2018)对机器学习中的自动微分给出了很好的概述。

图5.10显示了一个简单的图，表示从输入x到输出y通过一些中间变量a、b的数据流。如果我们要计算导数dy/dx，我们会应用链式法则并得到

$$dy/dx = (dy/db) \cdot (db/da) \cdot (da/dx) \quad (5.119)$$

直观地说，前向和反向模式在乘法顺序上有所不同。由于矩阵乘法的结合律，我们可以在以下之间选择：

$$dy/dx = (dy/db) \cdot (db/da) \cdot (da/dx) \quad (5.120)$$

$$dy/dx = (dy/db) \cdot (db/da) \cdot (da/dx) \quad (5.121)$$

方程(5.120)是**反向模式**，因为梯度通过图向后传播，即与数据流方向相反。方程(5.121)是**前向模式**，其中梯度与数据从左到右通过图流动。

在以下内容中，我们将专注于反向模式自动微分，即反向传播。在神经网络的背景下，输入维度通常远高于标签的维度，反向模式在计算上比前向模式便宜得多。让我们从一个有指导意义的例子开始。

例5.14

考虑函数

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos(x^2 + \exp(x^2)) \quad (5.122)$$

来自(5.109)。如果我们要在计算机上实现函数f，我们可以通过使用**中间变量**来节省一些计算：

$$a = x^2 \quad (5.123) \quad b = \exp(a) \quad (5.124) \quad c = a + b \quad (5.125) \quad d = \sqrt{c} \quad (5.126) \quad e = \cos(c) \quad (5.127) \quad f = d + e \quad (5.128)$$



图5.11 计算图，包含输入x、函数值f和中间变量a、b、c、d、e

图5.11 计算图，包含输入x、函数值f和中间变量a、b、c、d、e

这与应用链式法则时发生的思维过程相同。注意前面的方程组比直接实现(5.109)中定义的函数f(x)需要更少的运算。

图5.11中相应的计算图显示了获得函数值f所需的数据流和计算。

包含中间变量的方程组可以被视为计算图，这种表示在神经网络软件库的实现中被广泛使用。我们可以通过回忆基本函数导数的定义，直接计算中间变量相对于其相应输入的导数。我们得到以下结果：

$$\begin{aligned} \partial a / \partial x &= 2x \quad (5.129) \quad \partial b / \partial a = \exp(a) \quad (5.130) \quad \partial c / \partial a = 1 \quad \partial c / \partial b = 1 \quad (5.131) \quad \partial d / \partial c = 1 / (2\sqrt{c}) \quad (5.132) \quad \partial e / \partial c = -\sin(c) \quad (5.133) \quad \partial f / \partial d = 1 \quad \partial f / \partial e = 1 \quad (5.134) \end{aligned}$$

通过查看图5.11中的计算图，我们可以通过从输出向后工作来计算 $\partial f / \partial x$ ，得到

$$\begin{aligned} \partial f / \partial c &= (\partial f / \partial d) \cdot (\partial d / \partial c) + (\partial f / \partial e) \cdot (\partial e / \partial c) \quad (5.135) \quad \partial f / \partial b = (\partial f / \partial c) \cdot (\partial c / \partial b) \quad (5.136) \quad \partial f / \partial a = (\partial f / \partial b) \cdot (\partial b / \partial a) + (\partial f / \partial c) \cdot (\partial c / \partial a) \quad (5.137) \quad \partial f / \partial x = (\partial f / \partial a) \cdot (\partial a / \partial x) \quad (5.138) \end{aligned}$$

注意我们隐式地应用了链式法则来获得 $\partial f / \partial x$ 。通过代入基本函数导数的结果，我们得到

$$\begin{aligned} \partial f / \partial c &= 1 \cdot (1 / (2\sqrt{c})) + 1 \cdot (-\sin(c)) \quad (5.139) \quad \partial f / \partial b = (\partial f / \partial c) \cdot 1 \quad (5.140) \quad \partial f / \partial a = (\partial f / \partial b) \cdot \exp(a) + (\partial f / \partial c) \cdot 1 \quad (5.141) \quad \partial f / \partial x = (\partial f / \partial a) \cdot 2x \quad (5.142) \end{aligned}$$

通过将上面的每个导数视为一个变量，我们观察到计算导数所需的计算与计算函数本身的复杂度相似。这是相当反直觉的，因为导数 $\partial f / \partial x$ (5.110)的数学表达比(5.109)中函数 $f(x)$ 的数学表达复杂得多。

自动微分是例5.14的形式化。设 $x_1, \dots, x_d$ 为函数的输入变量， $x_{d+1}, \dots, x_{D-1}$ 为中间变量， x_D 为输出变量。那么计算图可以表示如下：

$$\text{对于 } i = d+1, \dots, D: \quad x_i = g_i(x_{Pa(x_i)}) \quad (5.143)$$

其中 $g_i(\cdot)$ 是基本函数， $x_{Pa(x_i)}$ 是图中变量 x_i 的父节点。给定以这种方式定义的函数，我们可以使用链式法则逐步计算函数的导数。回忆根据定义 $f = x_D$ ，因此

$$\partial f / \partial x_D = 1 \quad (5.144)$$

对于其他变量 x_i ，我们应用链式法则

$$\partial f / \partial x_i = \sum (\partial f / \partial x_j) \cdot (\partial x_j / \partial x_i) = \sum (\partial f / \partial x_i) \cdot (\partial g_i / \partial x_i) \quad (5.145)$$

$$\partial x[i] \quad \partial x[j] \quad \partial x[i] \quad \partial x[j] \quad \partial x[i] \quad x[i] \quad [i] \quad x[i] \quad [i] \quad [i] \quad [Pa[i] \quad x[j] \quad D] \quad x[i] \quad [i] \quad x[i] \quad [i] \quad [Pa[i] \quad x[j] \quad D]$$

其中 $Pa(x[i])$ 是计算图中 $x[i]$ 的父节点集合。

自动微分方程(5.143)是函数的前向传播，而(5.145)在反向模式中是梯度通过计算图的反向传播。

需要一个解析树。对于神经网络训练，我们反向传播预测的误差

相对于标签。

上述自动微分方法在我们拥有可以表示为计算图的函数时都能工作，其中基本函数是可微的。实际上，函数甚至可能不是数学函数而是计算机程序。然而，并非所有计算机程序都可以自动微分，例如，如果我们找不到可微的基本函数。编程结构，如for循环和if语句，也需要更加小心处理。

5.7 高阶导数

到目前为止，我们已经讨论了梯度，即一阶导数。有时，我们对高阶导数感兴趣，例如，当我们想要使用Newton方法进行优化时，这需要二阶导数(Nocedal and Wright, 2006)。在第5.1.1节中，我们讨论了使用多项式近似函数的Taylor级数。在多变量情况下，我们可以做完全相同的事情。在下面，我们将做完全相同的事情。但让我们从一些记号开始。

考虑一个两变量 x, y 的函数 $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ 。我们使用以下记号表示高阶偏导数（和梯度）：

$\partial^2 f / \partial x^2$ 是 f 关于 x 的二阶偏导数。

$\partial^n f / \partial x^n$ 是 f 关于 x 的 n 阶偏导数。

$\partial^2 f / \partial y \partial x = \partial / \partial y (\partial f / \partial x)$ 是通过首先关于 x 偏微分然后关于 y 偏微分得到的偏导数。

$\partial^2 f / \partial x \partial y$ 是通过首先关于 y 偏微分然后关于 x 偏微分得到的偏导数。

Hessian是所有二阶偏导数的集合。

图5.12 函数的线性近似。原始函数 f 在 $x_0 = -2$ 处使用一阶Taylor级数展开进行线性化。

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$

如果 $f(x, y)$ 是二次（连续）可微函数，那么

$$\partial^2 f / \partial x \partial y = \partial^2 f / \partial y \partial x, \quad (5.146)$$

即微分的顺序无关紧要，相应的Hessian矩阵

$$H = [\partial^2 f / \partial x^2 \quad \partial^2 f / \partial x \partial y] [\partial^2 f / \partial x \partial y \quad \partial^2 f / \partial y^2] \quad (5.147)$$

是对称的。Hessian记为 $\nabla^2 f(x, y)$ 。一般地，对于 $x \in \mathbb{R}^n$ 和 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ，Hessian是一个 $n \times n$ 矩阵。Hessian测量函数在 (x, y) 周围局部的曲率。

备注（向量场的Hessian）。如果 $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 是向量场，Hessian是一个 $(m \times n \times n)$ 张量。◇

5.8 线性化和多变量Taylor级数

函数 f 的梯度 ∇f 通常用于 f 在 $\mathbf{x}_0$ 周围的局部线性近似：

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + (\nabla_{\mathbf{x}} f)(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0). \quad (5.148)$$

这里 $(\nabla_{\mathbf{x}} f)(\mathbf{x}_0)$ 是 f 关于 $\mathbf{x}$ 的梯度，在 $\mathbf{x}_0$ 处求值。

图5.12说明了函数 f 在输入 $\mathbf{x}_0$ 处的线性近似。原始函数被一条直线近似。这种近似在局部是准确的，但我们离 $\mathbf{x}_0$ 越远，近似就越差。方程(5.148)是 f 在 $\mathbf{x}_0$ 处的多变量Taylor级数展开的特殊情况，其中我们只考虑前两项。我们在下面讨论更一般的情况，这将允许更好的近似。

图5.13 可视化外积。向量的外积使数组的维数每项增加1。(a)给定向量 $\delta \in \mathbb{R}^4$ ，我们得到外积 $\delta := \delta \otimes \delta = \delta \delta^T \in \mathbb{R}^{4 \times 4}$ 作为矩阵。两个向量的外积产生一个矩阵；(b)三个向量的外积产生一个三阶张量。

(b)外积 $\delta \otimes \delta \otimes \delta := \delta \delta \delta \in \mathbb{R}^{4 \times 4 \times 4}$ 产生一个三阶张量（“三维矩阵”），即具有三个索引的数组。

定义5.7（多变量Taylor级数）。我们考虑一个函数

$$f: \mathbb{R}^D \rightarrow \mathbb{R} \quad (5.149) \quad \mathbf{x} \mapsto f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^D, \quad (5.150)$$

它在 $\mathbf{x}_0$ 处是光滑的。当我们定义差向量 $\delta := \mathbf{x} - \mathbf{x}_0$ 时， f 在 $(\mathbf{x}_0)$ 处的多变量Taylor级数定义为

$$f(\mathbf{x}) = \sum_{k=0}^{\infty} (D_{\mathbf{x}}^k f)(\mathbf{x}_0) / k! \delta^k, \quad (5.151)$$

其中 $D_{\mathbf{x}}^k f(\mathbf{x}_0)$ 是 f 关于 $\mathbf{x}$ 的 k 阶（全）导数，在 $\mathbf{x}_0$ 处求值。

定义5.8（Taylor多项式）。 f 在 $\mathbf{x}_0$ 处的 n 次Taylor多项式包含级数(5.151)中的前 $n+1$ 个分量，定义为

$$T_n(\mathbf{x}) = \sum_{k=0}^n (D_{\mathbf{x}}^k f)(\mathbf{x}_0) / k! \delta^k. \quad (5.152)$$

在(5.151)和(5.152)中，我们使用了略微不严格的记号 $[\delta^k]$ ，

对于向量 $\mathbf{x} \in \mathbb{R}^D$ ， $D > 1$ 和 $k > 1$ ，这是未定义的。注意向量可以既是 k 阶张量，也是 f_{δ} 的 k 阶张量，即 k 维数组。

x

实现为 k 次

数组，矩阵作为二维数组的 k 折外积， $\in$ 一维数组 $\mathbf{z}$ | $\{k D \times D \times \dots \times D$ 第 k 阶张量 δ

$\mathbb{R}^D$ 作为向量 $\delta \in \mathbb{R}^D$ 的 k 折外积 $\otimes$ 获得。例如：

$$\delta^{10} := \delta \otimes \delta = \delta \delta^T, \delta[i,j] = \delta[i]\delta[j] \quad (5.153)$$

$$\delta^{11} := \delta \otimes \delta \otimes \delta, \delta[i,j,k] = \delta[i]\delta[j]\delta[k] \quad (5.154)$$

图5.13可视化了两个这样的外积。一般来说，我们在Taylor级数中获得项

$$D^k_x f(x_0) \delta^k = \sum \sum \dots \sum D^k_x f(x_0) [i_1, \dots, i_k] \delta[i_1] \dots \delta[i_k] \quad (5.155)$$

其中 $D^k_x f(x_0) \delta^k$ 包含k阶多项式。

现在我们定义了向量场的Taylor级数，让我们明确写出Taylor级数展开的前几项 $D^k_x f(x_0) \delta^k$ ，对于 $k = 0, \dots, 3$ 和 $\delta := x - x_0$:

$$k = 0: D^0_x f(x_0) \delta^0 = f(x_0) \in \mathbb{R} \quad (5.156)$$

$$k = 1: D^1_x f(x_0) \delta^1 = \nabla_x f(x_0) \delta = \sum \partial f(x_0) / \partial x[i] \delta[i] \in \mathbb{R} \quad (5.157)$$

$$k = 2: D^2_x f(x_0) \delta^2 = \text{tr}(H(x_0) \delta \delta^T) = \delta^T H(x_0) \delta \quad (5.158) = \sum \sum H[i,j] \delta[i] \delta[j] \in \mathbb{R} \quad (5.159)$$

$$k = 3: D^3_x f(x_0) \delta^3 = \sum \sum \sum D^3_x f(x_0) [i,j,k] \delta[i] \delta[j] \delta[k] \in \mathbb{R} \quad (5.160)$$

这里， $H(x_0)$ 是在 x_0 处评估的f的Hessian矩阵。

例5.15（两变量函数的Taylor级数展开）

考虑函数

$$f(x,y) = x^2 + 2xy + y^3 \quad (5.161)$$

我们想要计算f在 $(x_0, y_0) = (1, 2)$ 处的Taylor级数展开。

在开始之前，让我们讨论一下期望的结果：(5.161)中的函数是一个3次多项式。我们寻找Taylor级数展开，它本身是多项式的线性组合。因此，我们不期望Taylor级数展开包含四次或更高次项来表示三次多项式。这意味着确定(5.151)的前四项应该足以获得(5.161)的精确替代表示。

为了确定Taylor级数展开，我们从常数项和一阶导数开始：

$$f(1,2) = 13 \quad (5.162)$$

$$\partial f / \partial x = 2x + 2y \Rightarrow \partial f / \partial x(1,2) = 6 \quad (5.163)$$

$$\partial f / \partial y = 2x + 3y^2 \Rightarrow \partial f / \partial y(1,2) = 14 \quad (5.164)$$

因此，我们得到

$$D^1_{\{x,y\}} f(1,2) = \nabla_{\{x,y\}} f(1,2) = [\partial f / \partial x(1,2), \partial f / \partial y(1,2)] = [6, 14] \in \mathbb{R}^{1 \times 2} \quad (5.165)$$

使得

$$D^1_{\{x,y\}} f(1,2) \delta = [6, 14][x-1; y-2] = 6(x-1) + 14(y-2) \quad (5.166)$$

注意 $D^1_{\{x,y\}} f(1,2) \delta$ 只包含线性项，即一阶多项式。

二阶偏导数为：

$$\partial^{2f}/\partial x^2 = 2 \Rightarrow \partial^{2f}/\partial x^2(1,2) = 2 \quad (5.167)$$

$$\partial^{2f}/\partial y^2 = 6y \Rightarrow \partial^{2f}/\partial y^2(1,2) = 12 \quad (5.168)$$

$$\partial^2 f / \partial y \partial x = 2 \Rightarrow \partial^2 f / \partial y \partial x(1,2) = 2 \quad (5.169)$$

$$\partial^2 f / \partial x \partial y = 2 \Rightarrow \partial^2 f / \partial x \partial y(1,2) = 2 \quad (5.170)$$

当我们收集二阶偏导数时，我们得到Hessian矩阵

$$H = [\partial^{2f}/\partial x^2, \partial^2 f / \partial x \partial y; \partial^2 f / \partial y \partial x, \partial^{2f}/\partial y^2] = [2, 2; 2, 6y] \quad (5.171)$$

使得

$$H(1,2) = [2, 2; 2, 12] \in \mathbb{R}^{2 \times 2} \quad (5.172)$$

因此，Taylor级数展开的下一项为：

$$D^2_{\{x,y\}} f(1,2) \delta^2 / 2! = (1/2) \delta^T H(1,2) \delta \quad (5.173a)$$

$$= (1/2)[x-1, y-2][2, 2, 12][x-1; y-2] \quad (5.173b)$$

$$= (x-1)^2 + 2(x-1)(y-2) + 6(y-2)^2 \quad (5.173c)$$

这里， $D^2_{\{x,y\}} f(1,2) \delta^2$ 只包含二次项，即二阶多项式。

三阶导数为：

$$D^3_{\{x,y\}} f = [\partial H / \partial x, \partial H / \partial y] \in \mathbb{R}^{2 \times 2 \times 2} \quad (5.174)$$

$$D^3_{\{x,y\}} f[:,1] = \partial H / \partial x = [\partial^{3f}/\partial x^3, \partial^{3f}/\partial x^2 \partial y; \partial^3 f / \partial x \partial y \partial x, \partial^{3f}/\partial x \partial y^2] \quad (5.175)$$

$$D^3_{\{x,y\}} f[:,2] = \partial H / \partial y = [\partial^{3f}/\partial y \partial x^2, \partial^3 f / \partial y \partial x \partial y; \partial^{3f}/\partial y^2 \partial x, \partial^{3f}/\partial y^3] \quad (5.176)$$

由于(5.171)中Hessian矩阵的大多数二阶偏导数是常数，唯一非零的三阶偏导数是

$$\partial^{3f}/\partial y^3 = 6 \Rightarrow \partial^{3f}/\partial y^3(1,2) = 6 \quad (5.177)$$

更高阶导数和3次混合导数（例如 $\partial^{3f}/\partial x^2 \partial y$ ）为零，因此

$$D^3_{\{x,y\}} f[:,1] = [0, 0; 0, 0], D^3_{\{x,y\}} f[:,2] = [0, 0; 0, 6] \quad (5.178)$$

以及

$$D^3_{\{x,y\}} f(1,2) \delta^3 / 3! = (y-2)^3 \quad (5.179)$$

这收集了Taylor级数的所有三次项。总的来说， f 在 $(x_0, y_0) = (1,2)$ 处的（精确）Taylor级数展开是

$$f(x) = f(1,2) + D^1_{x,y} f(1,2)\delta + D^2_{x,y} f(1,2)\delta^2/2! + D^3_{x,y} f(1,2)\delta^3/3! \quad (5.180a)$$

$$= f(1,2) + \partial f/\partial x(1,2)(x-1) + \partial f/\partial y(1,2)(y-2) + (1/2!)[\partial^2 f/\partial x^2(1,2)(x-1)^2 + \partial^2 f/\partial y^2(1,2)(y-2)^2] + 2\partial^2 f/\partial x\partial y(1,2)(x-1)(y-2) + (1/6)\partial^3 f/\partial y^3(1,2)(y-2)^3 \quad (5.180b)$$

$$= 13 + 6(x-1) + 14(y-2)$$

$$\bullet (x-1)^2 + 6(y-2) + 2(x-1)(y-2) + (y-2)^3. \quad (5.180c)$$

在这种情况下，我们获得了(5.161)中多项式的精确Taylor级数展开，即(5.180c)中的多项式与(5.161)中的原始多项式相同。在这个特殊例子中，这个结果并不令人惊讶，因为原始函数是一个三阶多项式，我们通过常数项、一阶、二阶和三阶多项式的线性组合来表达它，如(5.180c)所示。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[170] **向量微积分**

5.9 延伸阅读

关于矩阵微分的更多细节，以及所需线性代数的简短回顾，可以在Magnus and Neudecker (2007)中找到。自动微分有着悠久的历史，我们参考Griewank and Walther (2003), Griewank and Walther (2008)和Elliott (2009)及其中的参考文献。

在机器学习（和其他学科）中，我们经常需要计算期望，即我们要求解形式为

$$\int \mathbb{E}[x][f(x)] = f(x)p(x)dx. \quad (5.181)$$

的积分。

即使 $p(x)$ 是方便的形式（例如，高斯分布），这个积分通常也不能解析求解。 f 的Taylor级数展开是找到近似解的一种方法：假设 $p(x) = N(\mu, \Sigma)$ 是高斯分布，那么围绕 μ 的一阶Taylor级数展开局部线性化了非线性函数 f 。对于线性函数，如果 $p(x)$ 是高斯分布，我们可以精确计算均值（和协方差）（见第6.5节）。这个性质被**扩展Kalman滤波器**(Maybeck, 1979)大量用于非线性动态系统（也称为”状态空间模型”）的在线状态估计。近似(5.181)中积分的其他确定性方法有**unscented变换**(Julier and Uhlmann, 1997)，它不需要任何梯度，或**Laplace近似**(MacKay, 2003; Bishop, 2006; Murphy, 2012)，它使用二阶Taylor级数展开（需要Hessian矩阵）对 $p(x)$ 在其模态周围进行局部高斯近似。

练习

5.1 计算 $f'(x)$, 其中 $f(x) = \log(x^4) \sin(x^3)$.

5.2 计算 logistic sigmoid 的导数 $f'(x) f(x) = 1/(1 + \exp(-x))$.

5.3 计算函数的导数 $f'(x) f(x) = \exp(-1/2\sigma^2(x - \mu)^2)$, 其中 $\mu, \sigma \in \mathbb{R}$ 是常数。

5.4 计算 $f(x) = \sin(x) + \cos(x)$ 在 $x_0 = 0$ 处的 Taylor 多项式 T_n , $n = 0, \dots, 5$ 。

5.5 考虑以下函数:

$$f_1(x) = \sin(x_1) \cos(x_2), x \in \mathbb{R}^2 \quad f_2(x, y) = x^T y, x, y \in \mathbb{R}^n$$

$$f_3(x) = x^T x, x \in \mathbb{R}^n$$

a. $\partial f_i / \partial x$ 的维度是什么?

b. 计算 Jacobian 矩阵。

5.6 对 f 关于 t 求导, 对 g 关于 X 求导, 其中

$$f(t) = \sin(\log(t^T t)), t \in \mathbb{R}^D \quad g(X) = \text{tr}(AXB), A \in \mathbb{R}^{(D \times E)}, X \in \mathbb{R}^{(E \times F)}, B \in \mathbb{R}^{(F \times D)},$$

其中 $\text{tr}(\cdot)$ 表示迹。

5.7 通过使用链式法则计算以下函数的导数 df/dx 。提供每个偏导数的维度。详细描述你的步骤。

a. $f(z) = \log(1 + z^T z), z = x^T x, x \in \mathbb{R}^D$

b. $f(z) = \sin(z), z = Ax + b, A \in \mathbb{R}^{(E \times D)}, x \in \mathbb{R}^D, b \in \mathbb{R}^E$

其中 $\sin(\cdot)$ 应用于 z 的每个元素。

5.8 计算以下函数的导数 df/dx 。详细描述你的步骤。

a. 使用链式法则。提供每个偏导数的维度。

$$f(z) = \exp(-1/2 z^T z) \quad z = g(y) = y^T S^{-1} y, y = h(x) = x - \mu$$

其中 $x, \mu \in \mathbb{R}^D, S \in \mathbb{R}^{(D \times D)}$ 。

b. $f(x) = \text{tr}(xx^T + \sigma^2 I), x \in \mathbb{R}^D$

这里 $\text{tr}(A)$ 是 A 的迹, 即对角元素 A_{ii} 的和。提示: 显式写出外积。

c. 使用链式法则。提供每个偏导数的维度。你不需要显式计算偏导数的乘积。

$f = \tanh(z) \in \mathbb{R}^M$ $z = Ax + b$, $x \in \mathbb{R}^N$, $A \in \mathbb{R}^{(M \times N)}$, $b \in \mathbb{R}^M$.

这里， $\tanh$ 应用于 z 的每个分量。

5.9 我们定义

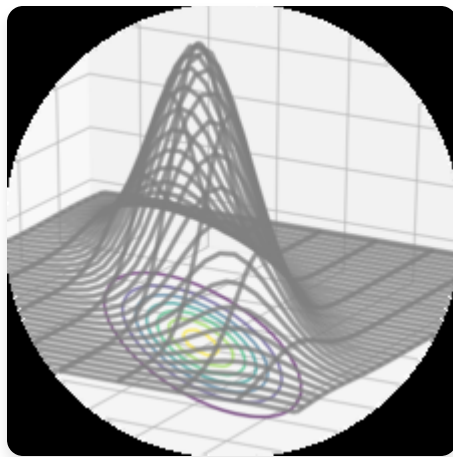
$g(x, z, v) := \log p(x, z) - \log q(z, v)$ $z := t(\epsilon, v)$

对于可微函数 p, q, t 和 $x \in \mathbb{R}^D, z \in \mathbb{R}^E, v \in \mathbb{R}^F, \epsilon \in \mathbb{R}^G$ 。通过使用链式法则，计算梯度

$d/dv g(x, z, v)$.

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

概率与分布



概率，粗略地说，涉及对不确定性的研究。概率可以被视为事件发生的次数分数，或对事件的信念程度。然后我们希望使用这个概率来测量

确定实验中某些事件发生的概率。如第1章所述，我们经常量化数据中的不确定性、machine learning模型中的不确定性，以及模型产生的预测中的不确定性。量化不确定性需要 $random\ variable$ 的概念，它是一个将随机实验结果映射到我们感兴趣的一组属性的函数。与 $random\ variable$ 相关的是一个函数，用于测量特定结果（或结果集合）发生的概率；这被称为 $probability\ distribution$ 。 $Probability\ distribution$ 被用作其他概念的构建块，如 $probabilistic\ modeling$ （第[8.4]节）、 $graphical\ models$ （第[8.5]节）和 $model\ selection$ （第[8.6]节）。在下一节中，我们将介绍定义 $probability\ space$ 的三个概念（ $sample\ space$ 、 $events$ 和 $event$ 的 $probability$ ）以及它们如何与第四个概念 $random\ variable$ 相关。这里的介绍有意略显简化，因为过于严格的表述可能会模糊概念背后的直觉。本章所介绍概念的概要如图[6.1]所示。

6.1 Probability Space的构建

概率论旨在定义一个数学结构来描述实验的随机结果。例如，在抛掷单枚硬币时，我们无法确定结果，但通过大量抛硬币，我们可以观察到平均结果的规律性。使用概率的这种数学结构，目标是进行自动推理，从这个意义上说，概率概括了逻辑推理[(Jaynes, 2003)]。

6.1.1 哲学问题

在构建自动推理系统时，经典的Boolean logic不允许我们表达某些形式的合理推理。考虑

Mathematics for Machine Learning by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong (2020). <https://mml-book.com>

图6.1 与random variables和probability distributions相关概念的思维导图，如本章所述。

[Mean] [Variance] [Bayes' Theorem]

[Summary statistics] [Product rule] [Sum rule] [Regression] [第9章]

[Property]

[Transformations] [Random variable] [Example] [Gaussian] [& distribution]

[Property] [Example] [第10章]

[Dimensionality reduction]

[Independence] [Bernoulli]

[Similarity] [Conjugate] [Sufficient statistics]

[Finite] [第11章]

[Density estimation]

[Inner product] [Beta]

[Exponential family]

以下场景：我们观察到A是错误的。我们发现B变得不太可信，尽管无法从经典逻辑中得出结论。我们观察到B是真的。似乎A变得更可信了。我们每天都使用这种推理形式。我们在等一个朋友，考虑三种可能性：H1，她准时；H2，她被交通延误了；H3，她被外星人绑架了。当我们观察到朋友迟到时，我们必须从逻辑上排除H1。我们也倾向于认为H2更有可能，尽管我们在逻辑上不需要这样做。最后，我们可能认为H3是可能的，但我们继续认为它相当不可能。我们如何得出H2是最合理答案的结论？以这种方式看，概率论可以被认为是Boolean logic的泛化。在machine

learning的背景下，它经常以这种方式应用于形式化自动推理系统的设计。关于概率论如何成为推理系统基础的进一步论据可以在[Pearl(1988)]中找到。

概率的哲学基础以及它应该如何与我们认为应该是真的（在逻辑意义上）相关的问题由Cox[(Jaynes, 2003)]研究。另一种思考方式是，如果我们对常识很精确，我们最终会构建概率。E. T. Jaynes (1922-1998) 确定了三个数学标准，必须适用于所有合理性：

1. 合理性的程度用实数表示。
2. 这些数字必须基于常识规则。

“对于合理推理，有必要将离散的真假值扩展到连续的可能性” [(Jaynes, 2003)]

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

Probability and Distributions

3. 由此产生的推理必须是一致的，“一致”一词有以下三个含义：
 - a. 一致性或无矛盾性：当通过不同方式可以达到相同结果时，在所有情况下都必须找到相同的可能性值。
 - b. 诚实性：必须考虑所有可用数据。
 - c. 可重现性：如果我们对两个问题的知识状态相同，那么我们必须为它们分配相同的可能性程度。

Cox-Jaynes定理证明这些可能性足以定义适用于可能性 p 的通用数学规则，直到通过任意单调函数进行变换。至关重要的是，这些规则就是概率规则。

备注。在machine learning和统计学中，概率有两种主要解释：Bayesian和frequentist解释[(Bishop, 2006; Efron and Hastie, 2016)]。Bayesian解释使用概率

描述事件不确定性程度的能力。它有时被称为”主观概率”或”信念程度”。

频率派解释考虑感兴趣事件相对于发生的事件总数的相对频率。事件的概率被定义为当拥有无限数据时事件的相对频率极限。[◇]

一些关于概率模型的机器学习教科书使用懒惰记号和术语，这令人困惑。本文也不例外。多个不同的概念都被称为”概率分布”，读者必须经常从上下文中理解其含义。帮助理解概率分布的一个技巧是检查我们是否试图建模某些分类的东西（离散随机变量）或某些连续的东西（连续随机变量）。我们在机器学习中处理的问题类型与是否考虑分类或连续模型密切相关。

6.1.2 概率和随机变量

在讨论概率时，有三个不同的概念经常被混淆。首先是概率空间的概念，它允许我们量化概率的概念。然而，我们大多数时候不直接使用这个基本概率空间。相反，我们使用随机变量（第二个概念），它将概率转移到一个更方便（通常是数值）的空间。第三个概念是与随机变量相关的分布或定律的概念。我们将在本节中介绍前两个概念，并在第6.2节中扩展第三个概念。

现代概率基于Kolmogorov提出的一套公理

《机器学习数学》草稿(2023-10-18)。反馈： 6.1 概率空间的构造 175

[(Grinstead and Snell, 1997; Jaynes, 2003)], 引入了样本空间、事件空间和概率测度三个概念。概率空间建模具有随机结果的现实世界过程（称为实验）。

样本空间 Ω

样本空间是实验所有可能结果的集合，通常用 Ω 表示。例如，连续两次抛硬币的样本空间为 $\{hh, tt, ht, th\}$ ，其中“h”表示“正面”，“t”表示“反面”。

事件空间 A

事件空间是实验潜在结果的空间。样本空间 Ω 的子集 A 在事件空间 A 中，如果在实验结束时我们可以观察到特定结果 $\omega \in \Omega$ 是否在 A 中。事件空间 A 通过考虑 Ω 的子集的集合获得，对于离散概率分布（第6.2.1节）， A 通常是 Ω 的幂集。

概率 P

对于每个事件 $A \in A$ ，我们关联一个数字 $P(A)$ ，它测量事件将发生的概率或信念程度。 $P(A)$ 被称为 A 的概率。

单个事件的概率必须在区间 $[0, 1]$ 内，样本空间 Ω 中所有结果的总概率必须为 1，即 $P(\Omega) = 1$ 。给定概率空间 (Ω, A, P) ，我们想用它来建模某些现实世界现象。在机器学习中，我们经常避免明确引用概率空间，而是引用感兴趣量的概率，我们用 T 表示。在本书中，我们将 T 称为 *目标空间*，将 T 的元素称为状态。我们引入一个函数 $X: \Omega \rightarrow T$ ，它取 Ω 的一个元素（结果）并返回一个特定的感兴趣量 x ，即 T 中的一个值。这种从 Ω 到 T 的关联/映射被称为 *随机变量*。例如，在抛两枚硬币并计算正面数量的情况下，随机变量 X 映射到三个可能的结果： $X(hh) = 2, X(ht) = 1, X(th) = 1$ ，和 $X(tt) = 0$ 。在这个特定情况下， $T = \{0, 1, 2\}$ ，我们感兴趣的是 T 元素上的概率。对于有限样本空间 Ω 和有限 T ，对应于随机变量的函数本质上是一个查找表。对于任何子集 $S \subseteq T$ ，我们关联 $PX \in [0, 1]$ （概率）到对应于随机变量 X 的特定事件发生。例6.1提供了术语的具体说明。

备注。上述样本空间 Ω 不幸地在不同书籍中被称为不同的名称。 Ω 的另一个常见名称是“状态空间”[(Jacod and Protter, 2004)]，但状态空间有时保留用于指代动力系统状态[(Hasselblatt and Katok, 2003)]。有时用来描述 Ω 的其他名称包括：“样本描述空间”、“可能性空间”和“事件空间”。[◇]

例6.1

我们假设读者已经熟悉计算事件集合的交集和并集的概率。可以在[Walpole et al. (2011)]第2章中找到对概率的更温和介绍，其中包含许多例子。

考虑一个统计实验，我们建模一个游乐场游戏，包括从袋子中抽取两枚硬币（有替换）。袋子里有来自美国（表示为 $\$$ ）和英国（表示为 $\pounds$ ）的硬币，由于我们从袋子中抽取两枚硬币，总共有四种结果。这个实验的状态空间或样本空间 Ω 是 $(\$, \$), (\$, \pounds), (\pounds, \$), (\pounds, \pounds)$ 。让我们假设硬币袋的组成是

使得抽取随机返回一个 $\$$ 的概率为 0.3。

我们感兴趣的事件是重复抽取返回 $\$$ 的总次数。让我们定义一个随机变量 X ，它将样本空间 Ω 映射到 T ，表示我们从袋子中抽取 $\$$ 的次数。从前面的样本空间可以看出，我们可以得到零个 $\$$ 、一个 $\$$ 或两个 $\$$ ，因此 $T = \{0, 1, 2\}$ 。随机变量 X （一个函数或查找表）可以表示为如下表格：

$$X((\$, \$)) = 2 \quad [(6.1)]$$

$$X((\$, \pounds)) = 1 \quad [(6.2)]$$

$$X((\pounds, \$)) = 1 \quad [(6.3)]$$

$$X((\pounds, \pounds)) = 0. \quad [(6.4)]$$

由于我们在抽取第二个硬币之前要归还第一个硬币，这意味着两次抽取彼此独立，我们将在第 [6.4.5] 节中讨论这一点。注意有两个实验结果映射到同一个事件，即只有一次抽取返回 $\$$ 。

因此， X 的概率质量函数（第 [6.2.1] 节）由下式给出：

$$P(X = 2) = P((\$, \$))$$

$$= P(\$) \cdot P(\$) = 0.3 \cdot 0.3 = 0.09 \quad [(6.5)]$$

$$P(X = 1) = P((\$, \pounds) \cup (\pounds, \$))$$

$$= P((\$, \pounds)) + P((\pounds, \$))$$

$$= 0.3 \cdot (1 - 0.3) + (1 - 0.3) \cdot 0.3 = 0.42 \quad [(6.6)]$$

$$P(X = 0) = P((\pounds, \pounds))$$

$$= P(\pounds) \cdot P(\pounds) = (1 - 0.3) \cdot (1 - 0.3) = 0.49. \quad [(6.7)]$$

[《机器学习数学》草稿（2023-10-18）。反馈：] <https://mml-book.com> [6.1 概率空间的构造] [177]

在计算中，我们等同了两个不同的概念： X 输出的概率和 Ω 中样本的概率。例如，在 [(6.7)] 中我们说 $P(X = 0) = P((\pounds, \pounds))$ 。考虑随机变量 $X: \Omega \rightarrow T$ 和子集 $S \subseteq T$ （例如， T 的单个元素，如投掷两枚硬币时获得一个正面的结果）。设 $X^{-1}(S)$ 为 S 在 X 下的原像，即 Ω 中在 X 下映射到 S 的元素集合； $\{\omega \in \Omega : X(\omega) \in S\}$ 。理解概率从 Ω 中的事件通过随机变量 X 转换的一种方法是将其与 S 的原像的概率关联起来 [Jacod and Protter, 2004]。对于 $S \subseteq T$ ，我们有记号

$$P_{\{X^{-1}\}}(S) = P(X \in S) = P(X(S)) = P(\{\omega \in \Omega : X(\omega) \in S\}) . \quad (6.8)$$

[(6.8)] 的左边是我们感兴趣的¹可能结果集合的概率（例如，\$ 的数量 = 1）。通过将状态映射到结果的随机变量 X ，我们在 [(6.8)] 的右边看到这是具有该属性的状态集合（在 Ω 中）的概率（例如， $\mathcal{L}, \mathcal{L}$ ）。我们说随机变量 X 根据特定的概率分布 P_X 分布，它定义了事件与随机变量结果概率之间的概率映射。换句话说，函数 $P_{\{X^{-1}\}}$ 或等价地 $P \circ X^{-1}$ 是随机变量 X 的规律或分布。

备注。目标空间，即随机变量 X 的范围 T ，用于表示概率空间的类型，即 T 随机变量。当 T 是有限的或可数无限的时，这被称为离散随机变量（第 [6.2.1] 节）。对于连续随机变量（第 [6.2.2] 节），我们只考虑 $T = \mathbb{R}$ 或 $T = \mathbb{R}^D$ 。◇

6.1.3 统计学

概率论和统计学经常一起出现，但它们涉及不确定性的不同方面。对比它们的一种方法是通过所考虑的问题类型。使用概率论，我们可以考虑某个过程的模型，其中潜在的不确定性被随机变量捕获，我们使用概率规则来推导发生的情况。在统计学中，我们观察到某事已经发生，并试图找出解释观察结果的潜在过程。从这个意义上说，机器学习在其构建充分代表生成数据过程的模型的目标方面更接近统计学。我们可以使用概率规则来获得某些数据的“最佳拟合”模型。

机器学习系统的另一个方面是我们对泛化误差感兴趣（见第 [8] 章）。这意味着我们实际上对系统在未来观察到的实例上的性能感兴趣，这些实例与我们迄今为止看到的实例不相同。对未来性能的这种分析依赖于概率和统计学，其中大部分内容超出了本章将要介绍的范围。鼓励感兴趣的读者查看 [Boucheron et al. (2013)] 和 [Shalev-Shwartz and Ben-David (2014)] 的书籍。我们将在第 [8] 章中看到更多关于统计学的内容。

6.2 离散和连续概率

让我们将注意力集中在描述第 [6.1] 节中介绍的事件概率的方法上。根据目标空间是离散的还是连续的，引用分布的自然方式是不同的。当目标空间 T 是离散的时，我们可以指定随机变量 X 取特定值 $x \in T$ 的概率，记作 $P(X = x)$ 。离散随机变量 X 的表达式 $P(X = x)$ 被称为 *概率质量函数*。当目标空间 T 是连续的，例如实数线 R 时，更自然的是指定随机变量 X 在区间内的概率，对于 $a < b$ ，记作 $P(a \leq X \leq b)$ 。按照惯例，我们指定随机变量 X 小于特定值 x 的概率，记作 $P(X \leq x)$ 。连续随机变量 X 的表达式 $P(X \leq x)$ 被称为 *累积分布函数*。我们将在第 [6.2.2] 节中讨论连续随机变量。

我们将重新审视术语并对比离散和连续

随机变量在第6.2.3节中。

备注。我们将使用 *单变量分布* 这一短语来指代单个随机变量的分布（其状态用非粗体 x 表示）。我们将把超过一个随机变量的分布称为 *多变量分布*，通常会考虑随机变量向量（其状态用粗体 x 表示）。[◇]

6.2.1 离散概率

当目标空间是离散的时，我们可以将多个随机变量的概率分布想象为填充一个（多维）数字数组。图6.2展示了一个例子。联合概率的目标空间是每个随机变量目标空间的笛卡尔积。我们将 *联合概率* 定义为两个值联合的条目

$$P(X = x_i, Y = y_j) = n_{ij}/N, (6.9)$$

其中 n_{ij} 是状态为 x_i 和 y_j 的事件数量， N 是事件总数。联合概率是两个事件交集的概率，即 $P(X = x_i, Y = y_j) = P(X = x_i \cap Y = y_j)$ 。

图6.2说明了离散概率分布的 *概率质量函数* (pmf)。对于两个随机变量 X 和 Y ，概率

图6.2 离散双变量概率质量函数的可视化，包含随机变量 X 和 Y 。该图改编自 Bishop (2006)。

$X = x$ 且 $Y = y$ 的概率（简称为） $p(x, y)$ ，称为联合概率。可以将概率视为一个函数，它接受状态 x 和 y 并返回一个实数，这就是我们写成 $p(x, y)$ 的原因。 X 取值 x 而不考虑随机变量 Y 的值的 *边际概率*（简称为） $p(x)$ 。我们写 $X \sim p(x)$ 表示随机变量 X 按照 $p(x)$ 分布。如果我们只考虑 $X = x$ 的实例，那么 $Y = y$ 的实例比例（*条件概率*）写成（简称为） $p(y | x)$ 。

示例6.2

考虑两个随机变量 X 和 Y ，其中 X 有五个可能状态， Y 有三个可能状态，如图6.2所示。我们用 n_{ij} 表示状态 $X = x_i$ 和 $Y = y_j$ 的事件数量，用 N 表示事件总数。值 c_i 是第 i 列单个频率的和，即 $c_i = \sum (j=1 \text{ 到 } 3) n_{ij}$ 。类似地，值 r_j 是行和，即 $r_j = \sum (i=1 \text{ 到 } 5) n_{ij}$ 。使用这些定义，我们可以紧凑地表达 X 和 Y 的分布。

每个随机变量的概率分布，即边际概率，可以看作是行或列的和

$$P(X = x_i) = c_i/N = (\sum (j=1 \text{ 到 } 3) n_{ij})/N (6.10)$$

和

$$P(Y = y_j) = r_j / N = (\sum_{i=1}^5 n_{ij}) / N \quad (6.11)$$

其中 c_i 和 r_j 分别是概率表的第 i 列和第 j 行。按照惯例，对于具有有限事件数的离散随机变量，我们假设概率之和为一，即

$$\sum_{i=1}^5 P(X = x_i) = 1 \text{ 且 } \sum_{j=1}^3 P(Y = y_j) = 1. \quad (6.12)$$

条件概率是特定单元格中行或列的比例。例如，给定 X 的 Y 的条件概率是

$$P(Y = y_j | X = x_i) = n_{ij} / c_i, \quad (6.13)$$

给定 Y 的 X 的条件概率是

$$P(X = x_i | Y = y_j) = n_{ij} / r_j. \quad (6.14)$$

在机器学习中，我们使用离散概率分布来建模分类变量，即取有限有序值集合的变量。它们可能是分类特征，如预测人员薪资时使用的大学学位，或分类标签，如手写识别中的字母表字母。离散分布也经常用于构建结合有限数量连续分布的概率模型（第11章）。

6.2.2 连续概率

在本节中我们考虑实值随机变量，即我们考虑实数线 $\mathbb{R}$ 区间的目标空间。在本书中，我们假装可以对实随机变量执行操作，就像我们有具有有限状态的离散概率空间一样。然而，这种简化在两种情况下不够精确：当我们无限次重复某事时，以及当我们想要从区间中抽取一个点时。第一种情况出现在我们讨论机器学习中的泛化误差时（第8章）。第二种情况出现在我们想要讨论连续分布时，如Gaussian（第6.5节）。对于我们的目的，精度的缺乏允许对概率进行更简洁的介绍。

备注。在连续空间中，还有两个反直觉的技术性问题。首先，所有子集的集合（用于定义第6.1节中的事件空间 $\mathcal{A}$ ）不够良好。 $\mathcal{A}$ 需要被限制为在集合补集、集合交集和集合并集下表现良好。其次，集合的大小（在离散空间中可以通过计算元素获得）变得复杂。集合的大小称为其测度(measure)。例如，离散集合的基数，

[d]维区间在 $\mathbb{R}$ 中的长度，以及 $\mathbb{R}$ 中区域的体积都是测度。在集合运算下表现良好并且具有拓扑结构的集合称为 *Borel σ -代数*。Betancourt 详细介绍了从集合论构造概率空间的仔细方法，而不会陷入技术细节；见 <https://tinyurl.com/yb3t6mfd>。更精确的构造，我们参考[Billingsley (1995)]和[Jacod and Protter](2004).]

在本书中，我们考虑实值随机变量及其相应的Borel σ -代数。我们将取值在 $\mathbb{R}^D$ 中的随机变量视为实值随机变量的向量。◇

定义 6.1 (概率密度函数)。函数 $f: \mathbb{R} \rightarrow \mathbb{R}$ 称为 *概率密度函数(pdf)*，如果

$$1. \forall x \in \mathbb{R}: f(x) \geq 0$$

$$2. \text{ 其积分存在且}$$

$$\int_{\mathbb{R}} f(x) dx = 1. \quad (6.15) \quad [\mathbb{R}]^D$$

对于离散随机变量的概率质量函数(pmf), [(6.15)]中的积分用求和[(6.12)]替代。

观察到概率密度函数是任何非负且积分为1的函数 f 。我们通过以下方式将随机变量 X 与此函数 f 相关联:

$$\int_a^b P(a \leq X \leq b) = \int_a^b f(x) dx, (6.16)$$

其中 $a, b \in \mathbb{R}$, $x \in \mathbb{R}$ 是连续随机变量 X 的结果。状态 $x \in \mathbb{R}$ 通过考虑 $x \in \mathbb{R}$ 的向量来类似定义。这种关联[(6.16)]称为随机变量 X 的律或分布。

注释。与离散随机变量相比, 连续随机变量 X 取特定值的概率 $P(X = x)$ 为零。这就像试图在[(6.16)]中指定 $a = b$ 的区间。 $\diamond$

定义 6.2 (累积分布函数)。多元实值随机变量 X 的累积分布函数(cdf), 其状态 $x \in \mathbb{R}[D]$ 由下式给出:

$$F_X(x) = P(X_1 \leq x_1, \dots, X[D] \leq x[D]), (6.17)$$

其中 $X = [X_1, \dots, X[D]]^T$, $x = [x_1, \dots, x[D]]^T$, 右侧表示随机变量 $X[i]$ 取值小于或等于 $x[i]$ 的概率。

cdf也可以表示为概率密度函数 $f(x)$ 的积分, 使得

$$\int_{-\infty}^x \int_{-\infty}^{x[D]} F_X(x) = \int_{-\infty}^{x_1} \dots \int_{-\infty}^{x[D]} f(z_1, \dots, z[D]) dz_1 \dots dz[D]. (6.18)$$

注释。我们重申, 谈论分布时实际上有两个不同的概念。首先是pdf的概念(用 $f(x)$ 表示), 它是一个非负函数且积分为1。其次是随机变量 X 的律, 即随机变量 X 与pdf $f(x)$ 的关联。 $\diamond$

图 6.3 离散和连续均匀分布的示例。详见例6.3。

a. 离散分布

b. 连续分布

对于本书的大部分内容, 我们不会使用记号 $f(x)$ 和 F_X , 因为我们通常不需要区分pdf和cdf。但是, 在第6.7节中我们需要小心处理pdf和cdf。

6.2.3 离散分布与连续分布的对比

回顾第6.1.2节，概率是正的，总概率和为1。对于离散随机变量(见[(6.12)]），这意味着每个状态的概率必须在区间[0, 1]内。但是，对于连续随机变量，归一化(见[(6.15)]）并不意味着密度值对于所有值都小于或等于1。我们在图6.3中使用均匀分布来说明离散和连续随机变量的这一点。

例 6.3

我们考虑均匀分布的两个例子，其中每个状态出现的可能性相等。这个例子说明了离散和连续概率分布之间的一些差异。

设 Z 是具有三个状态 $\{z = -1.1, z = 0.3, z = 1.5\}$ 的离散均匀随机变量。概率质量函数可以表示为概率值表：

z	-1.1	0.3	1.5
$P(Z = z)$	1/3	1/3	1/3

或者，我们可以将此视为图形(图6.3(a))，其中我们使用状态可以位于 x 轴上的事实， y 轴表示特定状态的概率。图6.3(a)中的 y 轴故意延长，使其与图6.3(b)相同。

设 X 是在范围 $0.9 \leq X \leq 1.6$ 内取值的连续随机变量，如图6.3(b)所示。观察到

表 6.1 点概率和区间概率的术语

离散和连续概率分布

离散 $P(X = x)$ 不适用 概率

概率质量函数 分布

连续 $p(x)$ $P(X \leq x)$

概率密度函数 累积分布函数

密度可以大于1。然而，需要满足

$$\int_{0.9}^{1.6} p(x) dx = 1. \quad (6.19)$$

备注。 关于离散概率分布还有一个额外的微妙之处。状态 $z_1, \dots, z_d$ 原则上没有任何结构，即通常没有办法比较它们，例如 $z_1 = \text{红色}$ ， $z_2 = \text{绿色}$ ， $z_3 = \text{蓝色}$ 。然而，在许多 machine learning 应用中，离散状态取数值，例如 $z_1 = -1.1$ ， $z_2 = 0.3$ ， $z_3 = 1.5$ ，我们可以说 $z_1 < z_2 < z_3$ 。取数值的离散状态特别有用，因为我们经常考虑随机变量的期望值（第 6.4.1 节）。◇

不幸的是，machine learning 文献使用的记号和术语隐藏了样本空间 Ω 、目标空间 T 和随机变量 X 之间的区别。对于随机变量 X 可能结果集合的值 x ，即 $x \in T$ ， $p(x)$ 表示随机变量 X 有结果 x 的概率。对于离散随机变量，这写作 $P(X = x)$ ，称为概率质量函数。pmf 通常被称为“分布”。对于连续变量， $p(x)$ 称为概率密度函数（通常称为密度）。更令人困惑的是，累积分布函数 $P(X \leq x)$ 也经常被被称为“分布”。在本章中，我们将使用记号 X 来指代单变量和多变量随机变量，并分别用 x 和 $\mathbf{x}$ 表示状态。我们在表 6.1 中总结了术语。

备注。 我们将使用“概率分布”这个表达不仅用于离散概率质量函数，也用于连续概率密度函数，尽管这在技术上是不正确的。与大多数 machine learning 文献一致，我们也依赖上下文来区分概率分布这个短语的不同用法。◇

6.3 和规则、积规则和Bayes定理

我们将概率论视为逻辑推理的扩展。正如我们在第6.1.1节中讨论的，这里提出的概率规则自然地满足要求中产生（Jaynes, 2003, 第2章）。概率建模（第8.4节）为设计machine learning方法提供了原理性基础。一旦我们定义了对应于数据和问题不确定性的概率分布（第6.2节），事实证明只有两个基本规则：和规则和积规则。

回忆从(6.9)可知 $p(x, y)$ 是两个随机变量 x, y 的联合分布。分布 $p(x)$ 和 $p(y)$ 是对应的边际分布， $p(y | x)$ 是给定 x 时 y 的条件分布。根据第6.2节中离散和连续随机变量的边际和条件概率定义，我们现在可以提出概率论中的两个基本规则。

第一个规则，**和规则**，表明

$$p(x) = \{ \sum_{y \in Y} p(x, y) \text{ 如果 } y \text{ 是离散的 } \int_Y p(x, y) dy \text{ 如果 } y \text{ 是连续的} \} \quad (6.20)$$

其中 Y 是随机变量 Y 的目标空间的状态。这意味着我们对随机变量 Y 的状态集合 y 求和（或积分）。和规则也称为**边际化性质**。

和规则将联合分布与边际分布联系起来。一般来说，当联合分布包含两个以上随机变量时，和规则可以应用于随机变量的任何子集，产生可能包含多个随机变量的边际分布。更具体地，如果 $\mathbf{x} = [x_1, \dots, x_D]^T$ ，我们通过重复应用和规则得到边际分布

$$p(x_{-i}) = \int p(x_1, \dots, x_D) d\mathbf{x} \quad (6.21)$$

其中我们对除 x_i 之外的所有随机变量进行积分/求和，这由，读作“除 i 之外的所有”。

备注。 概率建模的许多计算挑战都源于和规则的应用。当有许多变量或有许多状态的离散变量时，和规则归结为执行高维求和或积分。执行高维求和或积分在计算上通常是困难的，即没有已知的多项式时间算法来精确计算它们。

◇

第二个规则，称为**积规则**，通过以下方式将联合分布与条件分布联系起来

$$p(x, y) = p(y | x)p(x) \quad (6.22)$$

积规则可以解释为每个两个随机变量的联合分布都可以因式分解（写成乘积）为两个其他分布的乘积这一事实。两个因子是第一个随机变量的边际分布 $p(x)$ ，和给定第一个随机变量时第二个随机变量的条件分布 $p(y | x)$ 。由于随机变量在 $p(x, y)$ 中的顺序是任意的，积规则也意味着

$p(x, y) = p(x | y)p(y)$ 。确切地说，[(6.22)]是用离散随机变量的概率质量函数表示的。对于连续随机变量，乘积规则用概率

密度函数表示（第[6.2.3]节）。

在机器学习和贝叶斯统计中，我们经常对在观察到其他随机变量的情况下对未观察到的（潜在）随机变量进行推断感兴趣。让我们假设我们对未观察到的随机变量 x 有一些先验知识 $p(x)$ ，以及 x 和第二个随机变量 y 之间的某种关系 $p(y | x)$ ，我们可以观察到 y 。如果我们观察到 y ，我们可以使用贝叶斯定理根据观察到的 y 值对 x 得出一些结论。**贝叶斯定理**（也称为[[Bayes' theorem]]）

贝叶斯规则 或 贝叶斯定律) [[Bayes' rule]]

[[likelihood]] [[Bayes' law]] [[prior]]

$z \mid \{z\} \mid \{$

$p(y \mid x)p(x \mid y) = (6.23) p \mid \{z\}(y)$

[[posterior]] $\mid \{z\}$

[[evidence]]

是 [[(6.22)]] 中乘积规则的直接结果，因为

$$p(x, y) = p(x \mid y)p(y) \quad (6.24)$$

并且

$$p(x, y) = p(y \mid x)p(x) \quad (6.25)$$

所以

$$p(y \mid x)p(x \mid y)p(y) = p(y \mid x)p(x) \iff p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} \quad (6.26)$$

在 [[(6.23)]] 中， $p(x)$ 是 先验，它封装了我们在观察任何数据之前对未观察到的（潜在）变量 x 的主观先验 [[prior]] 知识。我们可以选择任何对我们有意义的先验，但确保先验在所有合理的 x 上具有非零 pdf（或 pmf）是至关重要的，即使它们非常罕见。

似然 $p(y \mid x)$ 描述了 x 和 y 是如何相关的，在 [[likelihood]]

离散概率分布的情况下，如果我们知道潜在变量 [似然] x ，它是数据 y 的概率 [[有时也]]。注意似然不是 x 中的]

分布 [[称为]]，而只是 y 中的分布。我们称 $p(y \mid x)$ 为“ x 的似然 [[“测量]]

（给定 y ）“或”给定 x 的 y 的概率”，但从不称为 y 的似然 [模型”.]

[[MacKay, 2003).]

后验 $p(x \mid y)$ 是贝叶斯统计中感兴趣的量 [[posterior]]，因为它准确地表达了我们感兴趣的内容，即在观察到 y 后我们对 x 的了解。

[[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[[概率和分布]]

量

Z

$$p(y) := \int p(y \mid x)p(x)dx = E[X][p(y \mid x)] \quad (6.27)$$

[marginal likelihood] 是 *边际似然/证据*。[(6.27)] 的右侧使用了

[evidence] [期望算子，我们在第 [6.4.1] 节中定义。根据定义，边际]

似然对 [(6.23)] 的分子关于

潜在变量 x 进行积分。因此，边际似然与

x 无关，它确保后验 $p(x | y)$ 被归一化。边际

似然也可以解释为期望似然，其中我们

相对于先验 $p(x)$ 取期望。除了后验的归一化之外，

边际似然在

贝叶斯模型选择中也起着重要作用，正如我们将在第 [8.6] 节中讨论的。由于

[贝叶斯定理也] [(8.44)] 中的积分，证据通常很难计算。

[称为] [贝叶斯定理 [(6.23)] 允许我们反转似然给出的 x]

[“概率”和 y 之间的关系。因此，贝叶斯定理有时 [逆.”]]

[概率逆] [称为 *概率逆*。我们将在]

第 [8.4] 节中进一步讨论贝叶斯定理。

备注。在贝叶斯统计中，后验分布是感兴趣的量，

因为它封装了来自先验和

数据的所有可用信息。与其携带后验，不如专注于

后验的某些统计量，例如后验的最大值，

我们将在第 [8.3] 节中讨论。然而，专注于后验的某些统计量

会导致信息丢失。如果我们从更大的背景来思考，

那么后验可以在决策系统中使用，

拥有完整的后验可能非常有用，并导致对

干扰具有鲁棒性的决策。例如，在基于模型的强化学习背景下，

[Deisenroth et al. (2015)] 表明使用合理转换函数的完整

后验分布导致非常快速的

(数据/样本高效) 学习, 而专注于
后验的最大值会导致一致的失败。因此, 拥有完整的后验
对于下游任务可能非常有用。在第 9 章中, 我们将
在线性回归的背景下继续这个讨论。[◇]

6.4 汇总统计和独立性

我们经常对汇总随机变量集合和比较

随机变量感兴趣。随机变量的统计量是该

随机变量的确定性函数。分布的汇总统计提供了随机变量行为的一个有用视图，

顾名思义，提供了汇总和特征化

分布的数字。我们描述均值和方差这两个众所周知的汇总统计。然后我们讨论比较一对

随机变量的两种方法：首先，如何说两个随机变量是独立的；

其次，如何计算它们之间的内积。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] [<https://mml-book.com>][.] [6.4 汇总统计和独立性]

6.4.1 均值和协方差

均值和（协）方差通常用于描述概率分布的性质（期望值和展开）。我们将在第 [6.6] 节中看到有一个有用的分布族（称为指数族），其中随机变量的统计量捕获所有可能的信息。

期望值概念是机器学习的中心概念，概率论本身的基础概念都可以从期望值中推导出来[(Whittle, 2000).]

定义 6.3（期望值）。对于服从分布 $p(x)$ 的单变量连续随机变量 X ，函数 $g: \mathbb{R} \rightarrow \mathbb{R}$ 的期望值由下式给出：

$\mathbb{Z}$

$$E[X][g(x)] = \int_{\mathbb{Z}} g(x)p(x)dx. \quad (6.28)$$

$\mathbb{X}$

相应地，对于服从分布 $p(x)$ 的离散随机变量 X ，函数 g 的期望值由下式给出：

$$E[X][g(x)] = \sum_{x \in \mathbb{X}} g(x)p(x), \quad (6.29)$$

其中 $\mathbb{X}$ 是随机变量 X 的可能结果集合（目标空间）。

在本节中，我们认为离散随机变量具有数值结果。这可以通过观察函数 g 以实数作为输入来理解。

备注。我们将多变量随机变量 X 视为单变量随机变量的有限向量 $[X_1, \dots, X_D]^T$ 。对于多变量随机变量，我们按元素定义期望值：

$$[E[X_1][g(x_1)]]$$

$$E[X][g(x)] = \int \dots \int_{\mathbb{R}^D} g(x) p(x) dx, (6.30) \quad [E[X_D][g(x_D)]]$$

其中下标 $E[X_d]$ 表示我们对向量 x 的第 d 个元素取期望值。◇

定义6.3定义了记号 EX 的含义，作为操作符表示我们应该对概率密度进行积分（对于连续分布）或对所有状态求和（对于离散分布）。均值的定义（定义6.4）是期望值的特殊情况，通过选择 g 为恒等函数获得。

定义 6.4（均值）。状态为 $x_D \in \mathbb{R}^D$ 的随机变量 X 的均值是一个平均值，定义为：

$$[E[X_1][x_1]]$$

$$EX = \int \dots \int_{\mathbb{R}^D} x p(x) dx, (6.31) \quad [E[X_D][x_D]]$$

其中

$$E[X_d][x_d] := \begin{cases} \int x_d p(x_d) dx_d & \text{如果 } X_d \text{ 是连续随机变量} \\ \sum x_i p(x_d = x_i) & \text{如果 } X_d \text{ 是离散随机变量 } x_i \in X \end{cases} (6.32)$$

对于 $d = 1, \dots, D$ ，其中下标 d 表示 x 的相应维度。积分和求和是对随机变量 X 目标空间的状态 X 进行的。

在一维情况下，还有两个其他的直观“平均”概念，即中位数和众数。中位数是排序后的“中间”值，即50%的值大于中位数，50%的值小于中位数。通过考虑累积分布函数(cdf)（定义6.2）为0.5的值，这个概念可以推广到连续值。对于不对称或具有长尾的分布，中位数提供的典型值估计比均值更接近人类直觉。此外，中位数比均值对异常值更鲁棒。中位数向高维的推广是非平凡的，因为在多于一个维度中没有明显的“排序”方法[(Hallin et al., 2010; Kong and Mizera, 2012)]。众数是最频繁出现的值。对于离散随机变量，众数定义为具有最高出现频率的 x 值。对于连续随机变量，众数定义为密度 $p(x)$ 中的峰值。特定密度 $p(x)$ 可能有多个众数，而且在高维分布中可能有非常大量的众数。因此，找到分布的所有众数在计算上可能是具有挑战性的。

例 6.4

考虑图6.4所示的二维分布：

$$p(x) = 0.4N(x; [1 \ 0]^T, [0.8 \ 0; 0 \ 4.2]) + 0.6N(x; [0 \ 0]^T, [2.0 \ 0; 0 \ 2.0 \ 1.7]) . (6.33)$$

我们将在第6.5节中定义高斯分布 $N(\mu, \sigma^2)$ 。还显示了每个维度中相应的边际分布。观察到该分布是双峰的（有两个众数），但其中一个边际分布是单峰的（有一个众数）。水平双峰单变量分布说明了均值和中位数可能彼此不同。虽然将二维中位数定义为每个维度中位数的连接很诱人，但我们无法定义二维点的排序这一事实使其变得困难。当我们说“无法定义排序”时，我们意思是有多多种方式定义关系 $<$ ，使得 $[3 \ 2]^T < [0 \ 3]^T$ 。

图 6.4 二维数据集的均值、众数和中位数的说明，以及其边际密度。

备注。期望值（定义6.3）是线性算子。例如，给定实值函数 $f(x) = ag(x) + bh(x)$ ，其中 $a, b \in \mathbb{R}$ 且 $x \in \mathbb{R}^D$ ，我们得到：

$$E[X][f(x)] = \int f(x)p(x)dx \quad (6.34a) = \int [ag(x) + bh(x)]p(x)dx \quad (6.34b) = a \int g(x)p(x)dx + b \int h(x)p(x)dx \quad (6.34c) = aE[X][g(x)] + bE[X][h(x)] \quad (6.34d)$$

◇

对于两个随机变量，我们可能希望描述它们彼此之间的对应关系。协方差直观地表示随机变量彼此依赖程度的概念。

定义 6.5（协方差（单变量））。两个单变量随机变量 $X, Y \in \mathbb{R}$ 之间的协方差由它们偏离各自均值的期望乘积给出，即

$$\text{Cov}[X, Y][x, y] := E[X, Y][(x - EX)(y - E[Y][y])] \quad (6.35)$$

备注。当与期望或协方差相关的随机变量从其参数中明确时，下标通常会被省略（例如， EX 通常写成 Ex ）。◇

通过使用期望的线性性质，定义6.5中的表达式可以重写为乘积的期望值减去期望值的乘积，即

$$\text{Cov}[x, y] = E[xy] - ExE[y] \quad (6.36)$$

变量与自身的协方差 $\text{Cov}[x, x]$ 称为方差，记为 VX 。方差的平方根称为标准差，通常记为 $\sigma(x)$ 。协方差的概念可以推广到多元随机变量。

定义 6.6（协方差（多元））。如果我们考虑两个多元随机变量 X 和 Y ，其状态分别为 $x \in \mathbb{R}^D$ 和 $y \in \mathbb{R}^E$ ，则 X 和 Y 之间的协方差定义为

$$\text{Cov}[x, y] = E[xy^T] - ExE[y]^T = \text{Cov}[y, x]^T \in \mathbb{R}^{(D \times E)} \quad (6.37)$$

定义6.6可以应用于两个参数中的同一个多元随机变量，这产生了一个有用的概念，它直观地捕获了随机变量的“离散程度”。对于多元随机变量，方差描述了随机变量各个维度之间的关系。

定义 6.7（方差）。具有状态 $x \in \mathbb{R}^D$ 和均值向量 $\mu \in \mathbb{R}^D$ 的随机变量 X 的方差定义为

$$VX = \text{Cov}[X][x, x] \quad (6.38a)$$

$$= E[X][(x - \mu)(x - \mu)^T] = E[X][xx^T] - ExEx^T \quad (6.38b)$$

$$= \begin{bmatrix} \text{Cov}[x_1, x_1] & \text{Cov}[x_1, x_2] & \dots & \text{Cov}[x_1, x_D] & \text{Cov}[x_2, x_1] & \text{Cov}[x_2, x_2] & \dots & \text{Cov}[x_2, x_D] & \vdots & \vdots & \ddots & \vdots & \text{Cov}[x_D, x_1] & \dots & \dots & \text{Cov}[x_D, x_D] \end{bmatrix} \quad (6.38c)$$

(6.38c)中的 $D \times D$ 矩阵称为多元随机变量 X 的协方差矩阵。协方差矩阵是对称且正半定的，它告诉我们数据分布的信息。在其对角线上，协方差矩阵包含边际的方差

图6.5



具有相同均值和各轴方差（彩色线）但协方差不同的二维数据集

具有相同均值和各轴方差（彩色线）但协方差不同的二维数据集

a. x 和 y 负相关。(b) x 和 y 正相关。

$$p(x_i) = \int p(x_1, \dots, x_D) dx_{\setminus i}, \quad (6.39)$$

其中“ $\setminus i$ ”表示“除 i 之外的所有变量”。非对角线项是交叉协方差项 $\text{Cov}[x_i, x_j]$ ，其中 $i, j = 1, \dots, D, i \neq j$ 。

备注。在本书中，我们通常假设协方差矩阵是正定的，以便更好地建立直觉。因此我们不讨论导致正半定（低秩）协方差矩阵的特殊情况。◇

当我们想要比较不同随机变量对之间的协方差时，发现每个随机变量的方差会影响协方差的值。协方差的标准化版本称为相关系数。

定义 6.8（相关系数）。两个随机变量 X, Y 之间的相关系数由下式给出

$$\text{corr}[x, y] = \text{Cov}[x, y] / \sqrt{VxVy} \in [-1, 1]. \quad (6.40)$$

相关矩阵是标准化随机变量 $x/\sigma(x)$ 的协方差矩阵。换句话说，在相关矩阵中，每个随机变量都除以其标准差（方差的平方根）。

协方差（和相关系数）表明两个随机变量是如何相关的；见图6.5。正相关 $\text{corr}[x, y]$ 意味着当 x 增长时， y 也期望增长。负相关意味着当 x 增加时， y 减少。

6.4.2 经验均值和协方差

第6.4.1节中的定义通常也称为总体均值和协方差，因为它指的是总体的真实统计量。在机器学习中，我们需要从数据的经验观察中学习。考虑一个随机变量 X 。从总体统计到经验统计实现有两个概念步骤。首先，我们利用有限数据集（大小为 N ）这一事实，构造一个有限数量相同随机变量 $X_1, \dots, X_N$ 的函数的经验统计量。其次，我们观察数据，即查看每个随机变量的实现 $x_1, \dots, x_N$ 并应用经验统计量。

具体来说，对于均值（定义6.4），给定特定数据集，我们可以获得均值的估计，称为经验均值或样本均值。经验协方差也是如此。

定义 6.9（经验均值和协方差）。经验均值向量是每个变量观测值的算术平均，定义为

$$\bar{\mathbf{x}} := \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n, \quad (6.41)$$

其中 $\mathbf{x}_n \in \mathbb{R}^D$ 。

与经验均值类似，经验协方差矩阵是一个 $D \times D$ 矩阵

$$\mathbf{\Sigma} := \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^{\top}. \quad (6.42)$$

为了计算特定数据集的统计量，我们使用实现值（观测值） $\mathbf{x}_1, \dots, \mathbf{x}_N$ 并使用公式 (6.41) 和 (6.42)。经验协方差矩阵是对称的、正半定的（见第 3.2.3 节）。

6.4.3 方差的三种表达式

现在我们专注于单个随机变量 X ，并使用前面的经验公式推导方差的三种可能表达式。以下推导对于总体方差是相同的，只是我们需要处理积分。方差的标准定义对应于协方差的定义（定义 6.5），是随机变量 X 与其期望值 μ 的平方偏差的期望，即

$$\mathbb{V}[X] := \mathbb{E}[(X - \mu)^2]. \quad (6.43)$$

公式 (6.43) 中的期望和均值 $\mu = \mathbb{E}[X]$ 使用公式 (6.32) 计算，取决于 X 是离散还是连续随机变量。公式 (6.43) 表示的方差是新随机变量 $Z := (X - \mu)^2$ 的均值。

当经验估计公式 (6.43) 中的方差时，我们需要使用双遍算法：第一遍遍历数据使用公式 (6.41) 计算均值 μ ，然后第二遍使用这个估计值 $\hat{\mu}$ 计算方差。结果表明，我们可以通过重新排列项来避免两次遍历。公式 (6.43) 可以转换为所谓的原始分数方差公式：

$$\mathbb{V}[X] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2. \quad (6.44)$$

公式 (6.44) 中的表达式可以记忆为“平方的均值减去均值的平方”。它可以在一次数据遍历中经验计算，因为我们可以同时累积 x_i （计算均值）和 x_i^2 ，其中 x_i 是第 i 个观测值。不幸的是，如果以这种方式实现，它可能在数值上不稳定。方差的原始分数版本在机器学习中很有用，例如，在推导偏差-方差分解时（Bishop, 2006）。

理解方差的第三种方式是它是所有观测值对之间成对差异的和。考虑随机变量 X 的实现样本 $x_1, \dots, x_N$ ，我们计算 x_i 和 x_j 对之间的平方差。通过展开平方，我们可以证明 N^2 个成对差异的和是观测值的经验方差：

$$\frac{1}{N^2} \sum_{i,j=1}^N (x_i - x_j)^2 = \frac{2}{N} \sum_{i=1}^N x_i^2 - \frac{2}{N^2} \left(\sum_{i=1}^N x_i \right)^2. \quad (6.45)$$

我们看到公式 (6.45) 是原始分数表达式 (6.44) 的两倍。这意味着我们可以将成对距离的和（其中有 N^2 个）表示为与均值的偏差和（其中有 N 个）。几何上，这意味着成对距离和点集中心的距离之间存在等价性。从计算角度来看，这意味着通过计算均值（求和中的 N 项），然后计算方差（求和中又是 N 项），我们可以得到一个有 N^2 项的表达式（公式 (6.45) 的左边）。

6.4.4 随机变量的和与变换

我们可能想要建模一种无法用教科书分布很好解释的现象（我们将在第 6.5 节和 6.6 节介绍一些），因此可能对随机变量进行简单操作（如两个随机变量相加）。

考虑两个随机变量 X, Y ，状态为 $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$ 。那么：

$$\mathbb{E}[\mathbf{x} + \mathbf{y}] = \mathbb{E}[\mathbf{x}] + \mathbb{E}[\mathbf{y}] \quad (6.46)$$

$$\mathbb{E}[\mathbf{x} - \mathbf{y}] = \mathbb{E}[\mathbf{x}] - \mathbb{E}[\mathbf{y}] \quad (6.47)$$

$$\mathbb{V}[\mathbf{x} + \mathbf{y}] = \mathbb{V}[\mathbf{x}] + \mathbb{V}[\mathbf{y}] + \text{Cov}[\mathbf{x}, \mathbf{y}] + \text{Cov}[\mathbf{y}, \mathbf{x}] \quad (6.48)$$

$$\mathbb{V}[\mathbf{x} - \mathbf{y}] = \mathbb{V}[\mathbf{x}] + \mathbb{V}[\mathbf{y}] - \text{Cov}[\mathbf{x}, \mathbf{y}] - \text{Cov}[\mathbf{y}, \mathbf{x}]. \quad (6.49)$$

当涉及随机变量的仿射变换时，均值和（协）方差表现出一些有用的性质。考虑一个随机变量 $\mathbf{X}$ ，均值为 $\boldsymbol{\mu}$ ，协方差矩阵为 $\boldsymbol{\Sigma}$ ，以及 $\mathbf{x}$ 的（确定性）仿射变换 $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$ 。那么 $\mathbf{y}$ 本身是一个随机变量，其均值向量和协方差矩阵分别为

$$\mathbb{E}[\mathbf{Y}] = \mathbb{E}[\mathbf{A}\mathbf{x} + \mathbf{b}] = \mathbf{A}\mathbb{E}[\mathbf{x}] + \mathbf{b} = \mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \quad (6.50)$$

$$\mathbb{V}[\mathbf{Y}] = \mathbb{V}[\mathbf{A}\mathbf{x} + \mathbf{b}] = \mathbb{V}[\mathbf{A}\mathbf{x}] = \mathbf{A}\mathbb{V}[\mathbf{x}]\mathbf{A}^\top = \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^\top. \quad (6.51)$$

此外，

$$\text{Cov}[\mathbf{x}, \mathbf{y}] = \mathbb{E}[\mathbf{x}(\mathbf{A}\mathbf{x} + \mathbf{b})^\top] - \mathbb{E}[\mathbf{x}]\mathbb{E}[(\mathbf{A}\mathbf{x} + \mathbf{b})^\top] \quad (6.52a)$$

$$= \mathbb{E}[\mathbf{x}]\mathbf{b}^\top + \mathbb{E}[\mathbf{x}\mathbf{x}^\top]\mathbf{A}^\top - \boldsymbol{\mu}\mathbf{b}^\top - \boldsymbol{\mu}\boldsymbol{\mu}^\top\mathbf{A}^\top \quad (6.52b)$$

$$= \boldsymbol{\mu}\mathbf{b}^\top - \boldsymbol{\mu}\mathbf{b}^\top + (\mathbb{E}[\mathbf{x}\mathbf{x}^\top] - \boldsymbol{\mu}\boldsymbol{\mu}^\top)\mathbf{A}^\top \quad (6.52c)$$

$$= \boldsymbol{\Sigma}\mathbf{A}^\top. \quad (6.52d)$$

其中 $\boldsymbol{\Sigma} = \mathbb{E}[\mathbf{x}\mathbf{x}^\top] - \boldsymbol{\mu}\boldsymbol{\mu}^\top$ 是 $\mathbf{X}$ 的协方差。

6.4.5 统计独立性

定义 6.10（独立性）。两个随机变量 X, Y 是统计独立的，当且仅当

$$p(x, y) = p(x)p(y) . \quad (6.53)$$

直观地说，两个随机变量 X 和 Y 是独立的，如果 y 的值（一旦已知）不会提供关于 x 的任何额外信息（反之亦然）。如果 X, Y 是（统计）独立的，那么

$$p(y | x) = p(y) \quad p(x | y) = p(x) \quad V[x + y] = V_x + V_y \quad \text{Cov}[x, y] = 0$$

最后一点的逆命题可能不成立，即两个随机变量可以具有零协方差但不是统计独立的。要理解原因，回想协方差只衡量线性依赖关系。因此，非线性依赖的随机变量可能具有零协方差。

例 6.5

考虑一个零均值的随机变量 X ($E_x = 0$) 并且 $E[x^3] = 0$ 。设 $y = x^2$ （因此， Y 依赖于 X ）并考虑 X 和 Y 之间的协方差 (6.36)。但这给出

$$\text{Cov}[x, y] = E[xy] - E_x E_y = E[x^3] = 0 . \quad (6.54)$$

在机器学习中，我们经常考虑可以建模为独立同分布 (*i.i.d.*) 随机变量的问题， $X_1, \dots, X_n$ 。对于两个以上的随机变量，“独立”一词（定义 6.10）通常指相互独立的随机变量，其中所有子集都是独立的（参见 Pollard (2002, 第4章) 和 Jacod and Protter (2004, 第3章)）。短语“同分布”意味着所有随机变量都来自同一分布。

在机器学习中另一个重要的概念是条件独立性。

定义 6.11（条件独立性）

两个随机变量 X 和 Y 在给定 Z 的条件下条件独立，当且仅当

$$p(x, y | z) = p(x | z)p(y | z) \quad \text{对于所有 } z \in Z , \quad (6.55)$$

其中 Z 是随机变量 Z 的状态集合。我们写 $X \perp\!\!\!\perp Y | Z$ 来表示 X 在给定 Z 的条件下独立于 Y 。

定义 6.11 要求关系式 (6.55) 必须对 z 的每个值都成立。(6.55) 的解释可以理解为“给定关于 z 的知识， x 和 y 的分布可以因式分解”。如果我们写 $X \perp\!\!\!\perp Y | \emptyset$ ，独立性可以看作条件独立性的特殊情况。通过使用概率的乘积法则 (6.22)，我们可以展开 (6.55) 的左边得到

$$p(x, y | z) = p(x | y, z)p(y | z) . \quad (6.56)$$

通过比较 (6.55) 的右边与 (6.56)，我们看到 $p(y | z)$ 出现在两者中，所以

$$p(x | y, z) = p(x | z) . \quad (6.57)$$

方程 (6.57) 提供了条件独立性的另一种定义，即 $X \perp\!\!\!\perp Y | Z$ 。这种替代表示提供了解释“给定我们知道 z ，关于 y 的知识不会改变我们对 x 的知识”。

6.4.6 随机变量的内积

回顾第 3.2 节中内积的定义。我们可以定义随机变量之间的内积，我们在本节中简要描述。如果我们有两个不相关的随机变量 X, Y ，那么

$$V[x + y] = V_x + V[y]. \quad (6.58)$$

由于方差以平方单位测量，这看起来非常像直角三角形的勾股定理 $c^2 = a^2 + b^2$ 。

在下面，我们看看是否可以找到 (6.58) 中不相关随机变量方差关系的几何解释。

图 6.6 随机变量的几何。如果随机变量 X 和 Y 不相关，它们是相应向量空间中的正交向量，勾股定理适用。

随机变量可以被视为向量空间中的向量，我们可以定义内积来获得随机变量的几何性质 (Eaton, 2007)。如果我们定义

$$\langle X, Y \rangle := \text{Cov}[x, y] \quad (6.59)$$

对于零均值随机变量 X 和 Y ，我们得到一个内积。我们看到协方差是对称的、正定的，并且在任一变量中都是线性的。随机变量的长度是

$$\|X\| = \sqrt{\text{Cov}[x, x]} = \sqrt{V_x} = \sigma_x, \quad (6.60)$$

即其标准差。随机变量越“长”，它就越不确定；长度为 0 的随机变量是确定性的。

如果我们查看两个随机变量 X, Y 之间的角度 θ ，我们得到

$$\cos \theta = \langle X, Y \rangle / (\|X\| \|Y\|) = \text{Cov}[x, y] / \sqrt{(V_x V[y])}, \quad (6.61)$$

这是两个随机变量之间的相关性（定义 6.8）。这意味着当我们从几何角度考虑时，我们可以将相关性视为两个随机变量之间角度的余弦。我们从定义 3.7 知道 $X \perp Y \iff \langle X, Y \rangle = 0$ 。在我们的情况下，这意味着 X 和 Y 正交当且仅当 $\text{Cov}[x, y] = 0$ ，即它们不相关。图 6.6 说明了这种关系。

备注。虽然使用欧几里得距离（构造

图6.7

两个随机变量 x_1 和 x_2 的 Gaussian 分布。

从前面关于内积的定义）来比较概率分布，遗憾的是这并不是获得分布间距离的最佳方法。回想一下，概率质量（或密度）是正的，并且需要加和为 1。这些约束意味着分布存在于称为统计流形的东西上。对这个概率分布空间的研究称为信息几何。计算分布间的距离通常使用 Kullback-Leibler 散度，这是考虑统计流形性质的距离的泛化。正如 Euclidean 距离是度量的特例（第 3.3 节），Kullback-Leibler 散度是两个更一般的散度类别的特例，称为 Bregman 散度和 f-散度。散度的研究超出了本书的范围，我们推荐读者参考信息几何领域创始人之一 Amari (2016) 的最新著作获取更多细节。◇

6.5 Gaussian分布

Gaussian分布是连续值随机变量最被广泛研究的概率分布。它也被称为正态分布。其重要性源于它具有许多计算上便利的性质，我们将在下文中讨论这些性质。特别地，我们将使用它来定义线性回归的似然和先验（第9章），并考虑Gaussian混合用于密度估计（第11章）。

机器学习的许多其他领域也受益于使用Gaussian分布，例如Gaussian过程、变分推理和强化学习。它在其他应用领域也被广泛使用，如信号处理（例如Kalman滤波器）、控制（例如线性二次调节器）和统计学（例如假设检验）。

当我们考虑独立且同分布的随机变量的和时，Gaussian分布自然出现。这被称为中心极限定理(Grinstead and Snell, 1997)。

图6.8

Gaussian分布叠加100个样本。(a) 一维情况；(b) 二维情况。

- a. 单变量（一维）Gaussian；红十字表示均值，红线表示方差的范围。
- b. 多变量（二维）Gaussian，从顶部查看。红十字表示均值，彩色线表示密度的等高线。

对于单变量随机变量，Gaussian分布的密度由下式给出

$$p(x | \mu, \sigma^2) = 1/(\sqrt{2\pi\sigma^2}) \exp(-(x-\mu)^2/(2\sigma^2)) \quad (6.62)$$

多变量Gaussian分布完全由均值向量 μ 和协方差矩阵 Σ 特征化，定义为

$$p(x | \mu, \Sigma) = (2\pi)^{-D/2} |\Sigma|^{-1/2} \exp(-1/2(x-\mu)^T \Sigma^{-1}(x-\mu)) \quad (6.63)$$

其中 $x \in \mathbb{R}^D$ 。我们写作 $p(x) = N(x | \mu, \Sigma)$ 或 $X \sim N(\mu, \Sigma)$ 。图6.7显示了二元Gaussian（网格），及其相应的等高线图。图6.8显示了单变量Gaussian和二元Gaussian及相应的样本。Gaussian的特例是零均值和单位协方差，即 $\mu = 0$ 和 $\Sigma = I$ ，被称为标准正态分布。

Gaussian在统计估计和机器学习中被广泛使用，因为它们对边际和条件分布具有闭式表达式。在第9章中，我们将广泛使用这些闭式表达式进行线性回归。使用Gaussian随机变量建模的主要优势是通常不需要变量变换（第6.7节）。由于Gaussian分布完全由其均值和协方差指定，我们通常可以通过将变换应用于随机变量的均值和协方差来获得变换后的分布。

6.5.1 Gaussian的边际和条件分布是Gaussian

在下文中，我们展示多变量随机变量一般情况下的边际化和条件化。如果初次阅读时感到困惑，建议读者考虑两个单变量随机变量。设 X 和 Y 是两个多变量随机变量，可能具有不同的维度。为了考虑应用概率的和规则和条件化的效果，我们明确地用连接状态 $[x^T, y^T]$ 来写Gaussian分布，

$$p(x, y) = N([\mu_x/\mu_y], [\Sigma_{xx} \Sigma_{xy}/\Sigma_{yx} \Sigma_{yy}]) \quad (6.64)$$

其中 $\Sigma_{xx} = \text{Cov}[x, x]$ 和 $\Sigma_{yy} = \text{Cov}[y, y]$ 分别是 x 和 y 的边际协方差矩阵, $\Sigma_{xy} = \text{Cov}[x, y]$ 是 x 和 y 之间的交叉协方差矩阵。

条件分布 $p(x | y)$ 也是 Gaussian 分布 (如图 6.9(c) 所示), 由下式给出 (在 [Bishop, 2006] 第 2.3 节中推导):

$$p(x | y) = \mathcal{N}(x | \mu_x + \Sigma_{xx}^{-1} \Sigma_{xy} (y - \mu_y), \Sigma_{xx} - \Sigma_{xx} \Sigma_{xy}^{-1} \Sigma_{xy}) \quad (6.65)$$

$$\mu_x = \mu_x + \Sigma_{xx}^{-1} \Sigma_{xy} (y - \mu_y) \quad (6.66)$$

$$\Sigma_x = \Sigma_x - \Sigma_{xx} \Sigma_{xy}^{-1} \Sigma_{xy} \quad (6.67)$$

注意, 在 (6.66) 中均值的计算中, y 值是一个观测值, 不再是随机的。

注释: 条件 Gaussian 分布出现在许多地方, 我们对后验分布感兴趣:

Kalman filter [(Kalman, 1960)] 是信号处理中状态估计最核心的算法之一, 它只是计算联合分布的 Gaussian 条件分布 [(Deisenroth and Ohlsson, 2011; Särkkä, 2013)]。

Gaussian processes [(Rasmussen and Williams, 2006)] 是函数分布的实际实现。在 Gaussian process 中, 我们假设随机变量的联合 Gaussianity。通过对观测数据的 (Gaussian) 条件化, 我们可以确定函数的后验分布。

潜在线性 Gaussian 模型 [(Roweis and Ghahramani, 1999; Murphy, 2012)] 包括概率主成分分析 (PPCA) [(Tipping and Bishop, 1999)]。我们将在第 10.7 节中更详细地研究 PPCA。

◇

联合 Gaussian 分布 $p(x, y)$ (见 (6.64)) 的边际分布 $p(x)$ 本身是 Gaussian 的, 通过应用求和法则 (6.20) 计算, 由下式给出:

$$\int p(x) = \int p(x, y) dy = \mathcal{N}(x | \mu_x, \Sigma_x) \quad (6.68)$$

相应的结果对 $p(y)$ 也成立, 通过对 x 积分得到。直观地说, 查看 (6.64) 中的联合分布, 我们忽略 (即积分掉) 我们不感兴趣的所有内容。这在图 6.9(b) 中有所说明。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[200] [Probability and Distributions]

例 6.6

[图 6.9] (a) 二元 [Gaussian]; (b) 联合 [Gaussian 分布的] [边际是] $p(x)$ [Gaussian]; (c) 的 $p(y|x)$ [条件分布] [也是] [Gaussian]。 [4] - [0] 1 x

[(a) 二元 Gaussian.]

[p] [x] [1] [2] [p] [x] [2] [1] [1] [1] [0] [6] 1 [均值] [均值] [1] [0] [2] [2] [0] [0] [8] [0] [4] [0] [6] [0] [4] [0] [2] [0] [2] [0] [0] [0] [0] [0] [5] [0] [0] [0] [0] [5] [1] [0] [1] [5] [0] [5] [0] [0] [0] [0] [5] [1] [0] [1] [5] x x

[(b) 边际分布.] [(c) 条件分布.]

考虑二元 Gaussian 分布（如图 6.9 所示）：

$$p(x_1, x_2) = N(\mu, \Sigma) \quad (6.69)$$

我们可以通过应用 (6.66) 和 (6.67) 分别获得均值和方差，从而计算在 $x_2 = -1$ 条件下的单变量 Gaussian 参数。数值上，这是：

$$\mu_{x_1 | x_2 = -1} = 0 + (-1) \cdot 0.2 \cdot (-1 - 2) = 0.6 \quad (6.70)$$

和

$$\sigma^2_{x_1 | x_2 = -1} = 0.3 - (-1) \cdot 0.2 \cdot (-1) = 0.1 \quad (6.71)$$

因此，条件 Gaussian 由下式给出：

$$p(x_1 | x_2 = -1) = N(0.6, 0.1) \quad (6.72)$$

相比之下，边际分布 $p(x_1)$ 可以通过应用 (6.68) 获得，本质上是使用随机变量 x_1 的均值和方差，给出我们：

$$p(x_1) = N(0, 0.3) \quad (6.73)$$

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>] [6.5 Gaussian Distribution]
[201]

6.5.2 Gaussian 密度的乘积

对于线性回归（第 9 章），我们需要计算 Gaussian likelihood。此外，我们可能希望假设 Gaussian 先验（第 9.3 节）。我们应用 Bayes 定理来计算后验，这导致 likelihood 和先验的乘法，即两个 Gaussian 密度的乘法。两个 Gaussian $\mathbf{N}(\mathbf{x} | \mathbf{a}, \mathbf{A}) \mathbf{N}(\mathbf{x} | \mathbf{b}, \mathbf{B})$ [推导是] 的乘积是一个按 $c \in \mathbb{R}$ 缩放的 Gaussian 分布，由 $\mathbf{c} \mathbf{N}(\mathbf{x} | \mathbf{c}, \mathbf{C})$ 给出，其中 [练习。]

$$\mathbf{C}^{-1} = (\mathbf{A} + \mathbf{B}) \quad (6.74)$$

$$c = C (\mathbf{A} \mathbf{a} + \mathbf{B} \mathbf{b}) \quad (6.75)$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}} |\mathbf{A} + \mathbf{B}| \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{c})^T (\mathbf{A} + \mathbf{B}) (\mathbf{x} - \mathbf{c})\right) \quad (6.76) \quad [2] \quad [2] \quad [2]$$

缩放常数 c 本身可以写成 $\mathbf{a}$ 或 $\mathbf{b}$ 中的 Gaussian 密度形式，具有“膨胀”的协方差矩阵 $\mathbf{A} + \mathbf{B}$ ，即 $\mathbf{c} = \mathbf{N}(\mathbf{a} | \mathbf{b}, \mathbf{A} + \mathbf{B}) = \mathbf{N}(\mathbf{b} | \mathbf{a}, \mathbf{A} + \mathbf{B})$ 。

注释: 为了符号方便，我们有时会使用 $\mathbf{N}(\mathbf{x} | \mathbf{m}, \mathbf{S})$ 来描述 Gaussian 密度的函数形式，即使 $\mathbf{x}$ 不是随机变量。我们在前面的演示中就是这样做的，当我们写：

$$\mathbf{c} = \mathbf{N}(\mathbf{a} | \mathbf{b}, \mathbf{A} + \mathbf{B}) = \mathbf{N}(\mathbf{b} | \mathbf{a}, \mathbf{A} + \mathbf{B}) \quad (6.77)$$

这里， $\mathbf{a}$ 和 $\mathbf{b}$ 都不是随机变量。然而，以这种方式写 $\mathbf{c}$ 比 (6.76) 更紧凑。◇

6.5.3 和及线性变换

如果 X, Y 是独立的 Gaussian 随机变量（即联合分布由 $p(x, y) = p(x)p(y)$ 给出），其中 $p(x) = N(x | \mu, \Sigma)$ 和 x

$p(y) = N(y | \mu, \Sigma)$ ，那么 $x + y$ 也是 Gaussian 分布的，由下式给出： [y] [y]

$$p(x + y) = N(\mu + \mu, \Sigma + \Sigma). \quad (6.78) \quad x[y] \quad x[y]$$

知道 $p(x + y)$ 是 Gaussian 的，均值和协方差矩阵可以使用 (6.46) 到 (6.49) 的结果立即确定。当我们考虑作用于随机变量的 i.i.d. Gaussian 噪声时，这个性质将很重要，如线性回归的情况（第 9 章）。

例 6.7

由于期望是线性运算，我们可以得到独立高斯随机变量的加权和

$$p(ax + by) = N(a\mu_x + b\mu_y, a^2\Sigma_x + b^2\Sigma_y) \quad (6.79)$$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[202] [[概率与分布]]

注记. 在第11章中有一个情况是高斯密度的加权和。这与高斯随机变量的加权和不同。[◇]

在定理6.12中，随机变量 x 来自一个密度，该密度是两个密度 $p_1(x)$ 和 $p_2(x)$ 的混合，由 α 加权。该定理可以推广到多元随机变量的情况，因为期望的线性性也适用于多元随机变量。然而，平方随机变量的概念需要被 $x^T x$ 替代。

定理 6.12. 考虑两个单变量高斯密度的混合

$$p(x) = \alpha p_1(x) + (1 - \alpha)p_2(x), \quad (6.80)$$

其中标量 $0 < \alpha < 1$ 是混合权重， $p_1(x)$ 和 $p_2(x)$ 是具有不同参数的单变量高斯密度（方程 (6.62)），即 $(\mu_1, \sigma_1^2) \neq (\mu_2, \sigma_2^2)$ 。

那么混合密度 $p(x)$ 的均值由每个随机变量均值的加权和给出：

$$Ex = \alpha\mu_1 + (1 - \alpha)\mu_2. \quad (6.81)$$

混合密度 $p(x)$ 的方差由以下给出

$$Vx = \alpha\sigma_1^2 + (1 - \alpha)\sigma_2^2 + \alpha(1 - \alpha)[\mu_1 - \mu_2]^2. \quad (6.82)$$

证明 混合密度 $p(x)$ 的均值由每个随机变量均值的加权和给出。我们应用均值的定义（定义6.4），并代入我们的混合 (6.80)，得到

$$\begin{aligned} Ex &= \int_{-\infty}^{\infty} xp(x)dx \quad (6.83a) = \int_{-\infty}^{\infty} (\alpha xp_1(x) + (1 - \alpha)xp_2(x))dx \quad (6.83b) = \alpha \int_{-\infty}^{\infty} xp_1(x)dx + (1 - \alpha) \int_{-\infty}^{\infty} xp_2(x)dx \quad (6.83c) = \alpha\mu_1 + (1 - \alpha)\mu_2. \quad (6.83d) \end{aligned}$$

为了计算方差，我们可以使用方差的原始分数版本(6.44)，这需要平方随机变量的期望表达式。这里我们使用随机变量函数（平方）的期望定义（定义6.3），

$$E[x^2] = \int_{-\infty}^{\infty} x^2 p(x) dx \quad (6.84a) = \int_{-\infty}^{\infty} (\alpha x^2 p_1(x) + (1 - \alpha) x^2 p_2(x)) dx \quad (6.84b)$$

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] <https://mml-book.com.> [6.5 高斯分布] [203]

$$\begin{aligned} &= \alpha \int_{-\infty}^{\infty} x^2 p_1(x) dx + (1 - \alpha) \int_{-\infty}^{\infty} x^2 p_2(x) dx \quad (6.84c) \\ &= \alpha(\mu_1^2 + \sigma_1^2) + (1 - \alpha)(\mu_2^2 + \sigma_2^2), \quad (6.84d) \end{aligned}$$

其中在最后的等式中，我们再次使用了方差的原始分数版本(6.44)，给出 $\sigma^2 = E[x^2] - \mu^2$ 。这被重新整理，使得平方随机变量的期望是平方均值和方差的和。

因此，方差由(6.84d)减去(6.83d)给出，

$$V_x = E[x^2] - (E_x)^2 \quad (6.85a) = \alpha(\mu_1^2 + \sigma_1^2) + (1 - \alpha)(\mu_2^2 + \sigma_2^2) - (\alpha\mu_1 + (1 - \alpha)\mu_2)^2 \quad (6.85b) = \alpha\sigma_1^2 + (1 - \alpha)\sigma_2^2 + \alpha\mu_1^2 + (1 - \alpha)\mu_2^2 - [\alpha\mu_1 + (1 - \alpha)\mu_2]^2. \quad (6.85c)$$

注记. 前面的推导对任何密度都成立，但由于高斯分布完全由均值和方差确定，混合密度可以以封闭形式确定。◇

对于混合密度，各个成分可以被视为条件分布（以成分身份为条件）。

方程(6.85c)是条件方差公式的一个例子，也称为全方差定律，它一般陈述对于两个随机变量X和Y，有 $V_x = E_Y[V_X[x|y]] + V_Y[E_X[x|y]]$ ，即X的（总）方差是期望条件方差加上条件均值的方差。

我们在例6.17中考虑了一个二元标准高斯随机变量X，并对其执行线性变换Ax。结果是一个均值为零、协方差为AA^T的高斯随机变量。观察到添加一个常数向量将改变分布的均值，而不影响其方差，即随机变量x + μ是均值为μ、协方差为单位矩阵的高斯分布。因此，高斯随机变量的任何线性/仿射变换都是高斯分布的。

考虑一个高斯分布的随机变量 $X \sim N(\mu, \Sigma)$ 。对于给定的适当形状的矩阵A，设Y是一个随机变量，使得 $y = Ax$ 是x的变换版本。我们可以通过利用期望是线性算子(6.50)来计算y的均值：

$$E[y] = E[Ax] = AE_x = A\mu. \quad (6.86)$$

类似地，y的方差可以通过使用(6.51)找到：

$$V[y] = V[Ax] = AV_xA^T = A \Sigma A^T. \quad (6.87)$$

这意味着随机变量y按照以下分布：

$$p(y) = N(y | A\mu, A \Sigma A^T). \quad (6.88)$$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[204] [[概率与分布]]

现在让我们考虑逆变换：当我们知道一个随机变量的均值是另一个随机变量的线性变换时。对于给定的满秩矩阵 $A \in \mathbb{R}^{(M \times N)}$ ，其中 $M \geq N$ ，设 $y \in \mathbb{R}^M$ 是一个高斯随机变量，均值为 Ax ，即

$$p(y) = N(y | Ax, \Sigma). \quad (6.89)$$

相应的概率分布 $p(x)$ 是什么？如果 A 是可逆的，那么我们可以写 $x = A^{-1}y$ 并应用前一段中的变换。然而，一般来说 A 不可逆，我们使用

一种类似于伪逆的方法[(3.57)。]也就是说，我们将两边同时左乘以 $[T][T]A$ ，然后求逆 AA^T ，它是对称且正定的，给出关系式

$$y^T [T]^T [T] Ax = Ax^T [T]^T [T] Ax \iff (A^T A) Ay = x^T [T]^T [T] Ax. \quad (6.90)$$

因此， x 是 y 的线性变换，我们得到

$$p([T]^T [T] Ax | (A^T A) Ay, (A^T A) A^T \Sigma A^T A) = N(x | (A^T A)^{-1} A^T y, (A^T A)^{-1} A^T \Sigma A^T A). \quad (6.91)$$

6.5.4 多元Gaussian分布的采样

我们不会解释在计算机上随机采样的细节，感兴趣的读者可以参考[Gentle (2004)]。对于多元Gaussian，这个过程包含三个阶段：首先，我们需要一个伪随机数源，在区间[0,1]内提供均匀采样；其次，我们使用非线性变换，如Box-Müller变换[(Devroye, 1986)]来获得单变量Gaussian的样本；第三，我们将这些样本的向量组合起来，获得多元标准正态分布 $N(0, I)$ 的样本。

对于一般的多元Gaussian，即均值非零且协方差不是单位矩阵的情况，我们使用Gaussian随机变量线性变换的性质。假设我们想要从具有均值 μ 和协方差矩阵 Σ 的多元[为了计算]Gaussian分布中生成样本 $x[i], i = 1, \dots, n$ 。我们希望[Cholesky][从提供多元标准正态分布 $N(0, I)$ 样本的采样器构造样本。][分解]

[矩阵，需要] [矩阵是]为了从多元正态分布 $N(\mu, \Sigma)$ 获得样本，我们可以使用[对称且][Gaussian随机变量线性变换的性质：]

[正定的] [(第3.2.3节)]。如果 $x \sim N(0, I)$ ，那么 $y = Ax + \mu$ ，其中 $AA^T = \Sigma$ 服从Gaussian分布，均值为 μ ，协方差矩阵为 Σ 。A的一个方便选择[协方差矩阵][具有这一]是使用协方差矩阵 $\Sigma = AA^T$ 的Cholesky分解(第4.3节)。Cholesky分解的好处是A是[性质。]三角矩阵，计算效率高。

6.6 共轭性与指数族

统计学教科书中许多“有名称的”概率分布都是为了建模特定类型的现象而发现的。例如，我们在第6.5节看到了Gaussian分布。这些分布之间也以复杂的方式相互关联[(Leemis and McQueston, 2008)。]对于该领域的初学者来说，弄清楚使用哪种分布可能会让人不知所措。此外，许多这些分布是在统计和计算需要纸笔完成的时代发现的。自然要问的是，在计算时代什么是有意义的概念[(Efron and Hastie, 2016)。]在前一节中，我们看到当分布是Gaussian时，推理所需的许多操作都可以方便地计算。此时值得回顾一下在机器学习背景下操作概率分布的期望条件：

1. 在应用概率规则(如Bayes定理)时存在某种“封闭性质”。封闭性是指应用特定操作返回相同类型的对象。
2. 随着我们收集更多数据，我们不需要更多参数来描述分布。
3. 由于我们对从数据中学习感兴趣，我们希望参数估计表现良好。

事实证明，称为指数族的分布类提供了一般性的正确平衡，同时保持了有利的计算和推理性质。在我们介绍指数族之前，让我们看看“命名”概率分布的另外三个成员：Bernoulli分布(例6.8)、Binomial分布(例6.9)和Beta分布(例6.10)。

例6.8

Bernoulli分布是单个二元随机变量 X 的分布，状态 $x \in \{0, 1\}$ 。它由单个连续参数 $\mu \in [0, 1]$ 控制，表示 $X = 1$ 的概率。Bernoulli分布 $\text{Ber}(\mu)$ 定义为



$$p(x | \mu) = \mu^x (1 - \mu)^{(1-x)}, x \in \{0, 1\}, (6.92)$$

$$E_x = \mu, (6.93)$$

$$V_x = \mu(1 - \mu), (6.94)$$

其中 E_x 和 V_x 是二元随机变量 X 的均值和方差。

Bernoulli分布可以使用的一个例子是当我们想要建模抛硬币时”正面”的概率。



图6.10 Binomial分布的例子，对于 $\mu \in \{0.1, 0.4, 0.75\}$ 和 $N = 15$

图6.10 Binomial分布的例子，对于 $\mu \in \{0.1, 0.4, 0.75\}$ 和 $N = 15$

注释。上面Bernoulli分布的重写，其中我们将布尔变量作为数值0或1使用并在指数中表达它们，是机器学习教科书中经常使用的技巧。另一个出现这种情况的例子是表达Multinomial分布。◇

例6.9 (Binomial分布)

Binomial分布是Bernoulli分布对整数分布的推广(如图6.10所示)。特别地，

Binomial分布可以用来描述在来自Bernoulli分布的 N 个样本集合中观察到 m 次 $X = 1$ 的概率，其中 $p(X = 1) = \mu \in [0, 1]$ 。

Binomial分布 $\text{Bin}(N, \mu)$ 定义为

!

$$p(m | N, \mu) = \binom{N}{m} \mu^m (1 - \mu)^{(N-m)}, (6.95)$$

$$E[m] = N\mu, (6.96)$$

$$V[m] = N\mu(1 - \mu), (6.97)$$

其中 $E[m]$ 和 $V[m]$ 分别是 m 的均值和方差。

一个可以使用Binomial分布的例子是，如果我们想要描述在 N 次抛硬币实验中观察到 m 次”正面”的概率，其中单次实验中观察到正面的概率是 μ 。

例6.10 (Beta分布)

我们可能希望对有限区间上的连续随机变量建模。Beta分布是连续随机变量 $\mu \in [0, 1]$ 上的分布，通常用来表示某些二元事件的概率（例如，控制Bernoulli分布的参数）。Beta分布 $\text{Beta}(\alpha, \beta)$ （如图6.11所示）本身由两个参数 $\alpha > 0, \beta > 0$ 控制，定义为

$$p(\mu | \alpha, \beta) = (\Gamma(\alpha + \beta) / (\Gamma(\alpha) \Gamma(\beta))) \mu^{(\alpha-1)} (1 - \mu)^{(\beta-1)} (6.98)$$

$$E[\mu] = \alpha / (\alpha + \beta), V[\mu] = \alpha\beta / ((\alpha + \beta)^2 (\alpha + \beta + 1)) (6.99)$$

其中 $\Gamma(\cdot)$ 是Gamma函数，定义为

$$\Gamma(t) := \int_0^\infty x^{t-1} \exp(-x) dx, t > 0. \quad (6.100)$$

$$\Gamma(t+1) = t\Gamma(t). \quad (6.101)$$

注意(6.98)中Gamma函数的分数使Beta分布标准化。

图6.11

 Beta分布在不同 α 和 β 值下的示例

直觉上， α 将概率质量向1移动，而 β 将概率质量向0移动。有一些特殊情况：

- 对于 $\alpha = 1 = \beta$ ，我们得到均匀分布 $U[0, 1]$ 。
- 对于 $\alpha, \beta < 1$ ，我们得到在0和1处有尖峰的双峰分布。
- 对于 $\alpha, \beta > 1$ ，分布是单峰的。
- 对于 $\alpha, \beta > 1$ 且 $\alpha = \beta$ ，分布是单峰的、对称的，并且在区间 $[0, 1]$ 中心，即众数/均值在 $1/2$ 处。

备注：有很多有名字分布，它们以不同方式相互关联。值得记住的是，每个命名分布都是为特定原因而创建的，但可能有其他应用。了解特定分布创建背后的原因通常能够洞察如何最好地使用它。我们介绍前面三个分布是为了能够说明共轭性（第6.6.1节）和指数族（第6.6.3节）的概念。

6.6.1 共轭性

根据Bayes定理(6.23)，后验与先验和似然的乘积成正比。先验的设定可能因为两个原因而棘手：首先，先验应该封装我们在看到任何数据之前对问题的知识。这通常很难描述。其次，通常无法解析地计算后验分布。然而，有一些在计算上便利的先验：*共轭先验*。

定义6.13（共轭先验）。 如果后验与先验具有相同的形式/类型，则先验对于似然函数是*共轭的*。

共轭性特别方便，因为我们可以通过更新先验分布的参数来代数地计算后验分布。

备注. 当考虑概率分布的几何时，共轭先验保持与似然相同的距离结构。

为了介绍共轭先验的具体例子，我们在例6.11中描述Binomial分布（定义在离散随机变量上）和Beta分布（定义在连续随机变量上）。

例6.11 (Beta-Binomial共轭性)

考虑一个Binomial随机变量 $x \sim \text{Bin}(N, \mu)$ ，其中

$$p(x \mid N, \mu) = \binom{N}{x} \mu^x (1 - \mu)^{N-x}, x = 0, 1, \dots, N, \quad (6.102)$$

是在N次抛硬币中找到x次”正面”结果的概率，其中 μ 是”正面”的概率。我们对参数 μ 设置Beta先验，即 $\mu \sim \text{Beta}(\alpha, \beta)$ ，其中

$$p(\mu \mid \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)} \mu^{\alpha-1} (1 - \mu)^{\beta-1}. \quad (6.103)$$

如果我们现在观察到某个结果 $x = h$ ，即我们在N次抛硬币中看到h次正面，我们计算 μ 上的后验分布为

$$p(\mu \mid x = h, N, \alpha, \beta) \propto p(x \mid N, \mu) p(\mu \mid \alpha, \beta) \quad (6.104a)$$

$$\propto \mu^h (1 - \mu)^{N-h} \mu^{\alpha-1} (1 - \mu)^{\beta-1} \quad (6.104b)$$

$$= \mu^{h+\alpha-1} (1 - \mu)^{(N-h)+\beta-1} \quad (6.104c)$$

表6.2 常见似然函数的共轭先验示例

似然	共轭先验	后验
Bernoulli	Beta	Beta
Binomial	Beta	Beta
Gaussian	Gaussian/inverse Gamma	Gaussian/inverse Gamma

似然	共轭先验	后验
Gaussian	Gaussian/inverse Wishart	Gaussian/inverse Wishart
Multinomial	Dirichlet	Dirichlet

$$\propto \text{Beta}(h + \alpha, N - h + \beta), \quad (6.104d)$$

即，后验分布是一个Beta分布，就像先验一样，即，Beta先验对于二项似然函数中的参数 μ 是共轭的。

在以下示例中，我们将推导出与Beta-二项共轭结果类似的结果。这里我们将展示Beta分布是伯努利分布的共轭先验。

例6.12 (Beta-伯努利共轭)

设 $x \in \{0, 1\}$ 按照参数为 $\theta \in [0, 1]$ 的伯努利分布分布，即 $p(x = 1 | \theta) = \theta$ 。这也可以表示为 $p(x | \theta) = \theta^x(1 - \theta)^{(1-x)}$ 。设 θ 按照参数为 α, β 的Beta分布分布，即 $p(\theta | \alpha, \beta) \propto \theta^{\alpha-1}(1 - \theta)^{\beta-1}$ 。

将Beta分布和伯努利分布相乘，我们得到

$$p(\theta | x, \alpha, \beta) = p(x | \theta)p(\theta | \alpha, \beta) \quad (6.105a)$$

$$\propto \theta^x(1 - \theta)^{(1-x)} \theta^{\alpha-1}(1 - \theta)^{\beta-1} \quad (6.105b)$$

$$= \theta^{(\alpha+x-1)}(1 - \theta)^{(\beta+(1-x)-1)} \quad (6.105c)$$

$$\propto p(\theta | \alpha + x, \beta + (1 - x)) \quad (6.105d)$$

最后一行是参数为 $(\alpha + x, \beta + (1 - x))$ 的Beta分布。

表6.2列出了概率建模中使用的一些标准似然参数的共轭先验示例。多项式、逆Gamma、逆Wishart和Dirichlet等分布可以在任何统计文本中找到，例如在Bishop (2006)中有描述。Beta分布是二项和伯努利似然中参数 μ 的共轭先验。对于高斯似然函数，我们可以在均值上放置共轭高斯先验。高斯似然在表中出现两次的原因是我们需要区分单变量和多变量情况。在单变量(标量)情况下，逆Gamma是方差的共轭先验。在多变量情况下，我们使用共轭逆Wishart分布作为协方差矩阵的先验。Dirichlet分布是多项式似然函数的共轭先验。更多细节，我们参考Bishop (2006)。

6.6.2 充分统计量

回忆一下，随机变量的统计量是该随机变量的确定性函数。例如，如果 $x = [x_1, \dots, x_n]^T$ 是单变量高斯随机变量的向量，即 $x_n \sim N(\mu, \sigma^2)$ ，那么样本均值 $\hat{\mu} = 1/N(x_1 + \dots + x_n)$ 是一个统计量。罗纳德·费雪爵士发现了充分统计量的概念：有一些统计量包含了从对应于所考虑分布的数据中可以推断出的所有可用信息。换句话说，充分统计量携带了对总体进行推断所需的所有信息，即它们是足以表示分布的统计量。

对于由 θ 参数化的一组分布，设 X 是在未知 θ_0 给定情况下具有分布 $p(x | \theta_0)$ 的随机变量。如果统计量向量 $\varphi(x)$ 包含关于 θ_0 的所有可能信息，则称其为 θ_0 的充分统计量。为了更正式地表述“包含所有可能信息”，这意味着给定 θ 的 x 的概率可以分解为不依赖于 θ 的部分和仅通过 $\varphi(x)$ 依赖于 θ 的部分。Fisher-Neyman分解定理形式化了这个概念，我们在定理6.14中不加证明地陈述这个定理。

定理6.14 (Fisher-Neyman)

[定理6.5在Lehmann and Casella (1998)中] 设 X 具有概率密度函数 $p(x | \theta)$ 。那么统计量 $\varphi(x)$ 对 θ 是充分的，当且仅当 $p(x | \theta)$ 可以写成如下形式

$$p(x | \theta) = h(x)g_{\theta}(\varphi(x)), \quad (6.106)$$

其中 $h(x)$ 是独立于 θ 的分布， g_{θ} 通过充分统计量 $\varphi(x)$ 捕获所有对 θ 的依赖性。

如果 $p(x | \theta)$ 不依赖于 θ ，那么对于任何函数 φ ， $\varphi(x)$ 都是平凡的充分统计量。更有趣的情况是 $p(x | \theta)$ 仅依赖于 $\varphi(x)$ 而不是 x 本身。在这种情况下， $\varphi(x)$ 是 θ 的充分统计量。

在机器学习中，我们考虑来自分布的有限数量样本。人们可以想象，对于简单分布(如例6.8中的伯努利分布)，我们只需要少量样本来估计分布的参数。我们也可以考虑相反的问题：如果我们有一组数据(来自未知分布的样本)，哪个分布给出了最佳拟合？一个自然的问题是，当我们观察到更多数据时，我们是否需要更多参数 θ 来描述分布？事实证明，一般来说答案是肯定的，这在非参数统计学中有研究(Wasserman, 2007)。一个相反的问题是考虑哪类分布具有有限维充分统计量，即描述它们所需的参数数量不会任意增加。答案是指数族分布，在下一节中描述。

6.6.3 指数族

当考虑分布（离散或连续随机变量的分布）时，我们可以有三个可能的抽象层次。在第一层次（频谱的最具体端），我们有一个具有固定参数的特定命名分布，例如零均值和单位方差的单变量Gaussian $N(0,1)$ 。在机器学习中，我们经常使用第二层次的抽象，即我们固定参数形式（单变量Gaussian）并从数据中推断参数。例如，我们假设一个单变量Gaussian $N(\mu, \sigma^2)$ 具有未知均值 μ 和未知方差 σ^2 ，并使用最大似然拟合来确定最佳参数 (μ, σ) 。当我们在第9章考虑线性回归时，我们将看到这个例子。第三层次的抽象是考虑分布族，在本书中，我们考虑指数族。单变量Gaussian是指数族成员的一个例子。许多广泛使用的统计模型，包括表6.2中的所有“命名”模型，都是指数族的成员。它们都可以统一为一个概念(Brown, 1986)。

备注。一个简短的历史轶事：像数学和科学中的许多概念一样，指数族是由不同的研究者在同一时间独立发现的。在1935-1936年间，塔斯马尼亚的Edwin Pitman、巴黎的Georges Darmois和纽约的Bernard Koopman独立证明了指指数族是在重复独立采样下享有有限维充分统计量的唯一族(Lehmann and Casella, 1998)。◇

指数族是一个概率分布族，由 $\theta \in \mathbb{R}^D$ 参数化，形式为

$$p(x | \theta) = h(x) \exp(\langle \theta, \varphi(x) \rangle - A(\theta)), \quad (6.107)$$

其中 $\varphi(x)$ 是充分统计量的向量。一般来说，任何内积（第3.2节）都可以在(6.107)中使用，为了具体起见，我们这里将使用标准点积 $\langle \theta, \varphi(x) \rangle = \theta^T \varphi(x)$ 。注意指数族的形式本质上是Fisher-Neyman定理（定理6.14）中 $g_{\theta}(\varphi(x))$ 的特定表达。

因子 $h(x)$ 可以通过将另一个条目($\log h(x)$)添加到充分统计量向量 $\varphi(x)$ 中，并约束相应的参数 $\theta_0 = 1$ ，从而被吸收到点积项中。项 $A(\theta)$ 是确保分布求和或积分为1的归一化常数，称为**对数配分函数**。通过忽略这两项，可以获得指数族的良好直观概念

212 概率和分布

并将指数族视为以下形式的分布

$$p(x | \theta) \propto \exp(\theta^T \varphi(x)). \quad (6.108)$$

对于这种参数化形式，参数 θ 被称为**自然参数**。乍一看，指数族似乎是通过将指数函数添加到点积结果中的平凡变换。然而，有许多含义允许基于我们可以在 $\varphi(x)$ 中捕获数据信息这一事实进行方便的建模和高效的计算。

例6.13 (Gaussian作为指数族)

考虑单变量Gaussian分布 $N(\mu, \sigma^2)$ 。设 $\varphi(x) = [x, x^2]^T$ 。然后通过使用指数族的定义，

$$p(x | \theta) \propto \exp(\theta_1 x + \theta_2 x^2). \quad (6.109)$$

设置

$$\theta = [\mu/\sigma^2, -1/(2\sigma^2)]^T \quad (6.110)$$

并代入(6.109)，我们得到

$$p(x | \theta) \propto \exp(\mu x / \sigma^2 - x^2 / (2\sigma^2)) \propto \exp(-1/(2\sigma^2)(x - \mu)^2). \quad (6.111)$$

因此，单变量Gaussian分布是指数族的成员，具有充分统计量 $\varphi(x) = [x, x^2]^T$ ，以及由(6.110)中的 θ 给出的自然参数。

例6.14 (Bernoulli作为指数族)

回顾例6.8中的Bernoulli分布

$$p(x | \mu) = \mu^x (1 - \mu)^{1-x}, x \in \{0, 1\}. \quad (6.112)$$

这可以写成指数族形式

$$\begin{aligned} p(x | \mu) &= \exp(\log(\mu^x (1 - \mu)^{1-x})) \quad (6.113a) \\ &= \exp[x \log \mu + (1 - x) \log(1 - \mu)] \quad (6.113b) \\ &= \exp[x \log \mu - x \log(1 - \mu) + \log(1 - \mu)] \\ &= \exp[x \log(\mu/(1-\mu)) + \log(1 - \mu)]. \quad (6.113d) \end{aligned}$$

最后一行(6.113d)可以通过观察以下内容被识别为指数族形式(6.107):

$$h(x) = 1 \quad (6.114)$$

Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>.

6.6 共轭性和指数族 213

$$\theta = \log(\mu/(1-\mu)) \quad (6.115)$$

$$\varphi(x) = x \quad (6.116)$$

$$A(\theta) = -\log(1 - \mu) = \log(1 + \exp(\theta)). \quad (6.117)$$

θ 和 μ 之间的关系是可逆的，因此

$$\mu = 1/(1 + \exp(-\theta)). \quad (6.118)$$

关系(6.118)用于获得(6.117)的右等式。

备注。原始Bernoulli参数 μ 和自然参数 θ 之间的关系被称为**sigmoid**或logistic函数。观察到 $\mu \in (0, 1)$ 但 $\theta \in \mathbf{R}$ ，因此sigmoid函数将实值压缩到范围 $(0, 1)$ 内。这个性质在机器学习中很有用，例如它在logistic回归中使用(Bishop, 2006, 第4.3.2节)，以及作为神经网络中的非线性激活函数(Goodfellow et al., 2016, 第6章)。◇

通常不明显如何找到特定分布的共轭分布的参数形式（例如，表6.2中的那些）。指数族提供了一种方便的方法来找到分布的共轭对。考虑随机变量 X 是指数族(6.107)的成员：

$$p(x | \theta) = h(x) \exp(\langle \theta, \varphi(x) \rangle - A(\theta)). \quad (6.119)$$

指数族的每个成员都有一个共轭先验(Brown,

[1986])

γ

$$p(\gamma | \theta) = h(\gamma) \exp(-A(\gamma)), \quad (6.120)$$

$$[c] \gamma - A(\theta) [2]$$

其中 γ 是具有维度 $\dim(\theta) + 1$ 。共轭先验的充分统计量是 $\gamma[2]$

θ 。通过使用指数族共轭先验的一般形式的知识，我们可以推导出对应于特定分布的共轭先验的函数形式。 $-A(\theta)$

例 6.15

回顾伯努利分布的指数族形式 [(6.113d)]

$$p(x | \mu) = \exp(x \log \mu + \log(1 - \mu)) \cdot 1_{\mu - \mu} \quad (6.121)$$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由剑桥大学出版社出版 (2020).]

[214] [[概率与分布]]

典型共轭先验具有以下形式

$$p(\mu | \alpha, \beta) = \exp(\alpha \log \mu + (\beta + \alpha) \log(1 - \mu) - A(\mu)) \cdot 1_{\mu \in [0, 1]} \quad (6.122)$$

其中我们定义 $[T] \gamma := [\alpha, \beta + \alpha]$ 和 $h(\mu) := \mu/(1 - \mu)$ 。方程[c] [(6.122)] 随后简化为

$$p(\mu | \alpha, \beta) = \exp[(\alpha - 1) \log \mu + (\beta - 1) \log(1 - \mu) - A(\mu)].$$

(6.123)

将其写成非指数族形式得到

$$p(\mu | \alpha, \beta) \propto \mu^\alpha (1 - \mu)^\beta, \quad (6.124)$$

我们将其识别为Beta分布 [(6.98).]。在例 [6.12] 中，我们假设Beta分布是伯努利分布的共轭先验，并证明了它确实是共轭先验。在这个例子中，我们通过查看伯努利分布在指数族形式中的典型共轭先验来推导Beta分布的形式。

如前一节所述，指数族的主要动机是它们具有有限维充分统计量。此外，共轭分布易于写下，共轭分布也来自指数族。从推理角度来看，最大似然估计表现良好，因为充分统计量的经验估计是充分统计量总体值的最优估计（回忆高斯分布的均值和协方差）。从优化角度来看，对数似然函数是凹函数，允许应用高效的优化方法（第7章）。

6.7 变量变换/逆变换

看似有很多已知分布，但实际上我们命名的分布集合相当有限。因此，理解变换随机变量如何分布通常很有用。例如，假设 X 是根据单变量正态分布 $N(0, 1)$ 分布的随机变量， $2X$ 的分布是什么？另一个在机器学习中很常见的例子是，给定 X_1 和 X_2 是单变量标准正态分布， $X_1 + X_2$ 的分布是什么？ $2X_1 + X_2$

计算 $X_1 + X_2$ 分布的一个选择是计算 $2X_1 + X_2$

X_1 和 X_2 的均值和方差，然后将它们结合起来。正如我们在第[6.4.4]节中看到的，当我们考虑随机变量的仿射变换时，我们可以计算结果随机变量的均值和方差。

[草稿 (2023-10-18) “机器学习数学”。反馈：] [https://mml-book.com][.] [6.7 变量变换/逆变换] [215]

然而，我们可能无法在变换下获得分布的函数形式。此外，我们可能对随机变量的非线性变换感兴趣，对此不容易获得闭式表达式。

注释（记号）。在本节中，我们将明确区分随机变量和它们取的值。因此，回忆我们使用大写字母 X, Y 表示随机变量，使用小写字母 x, y 表示随机变量在目标空间 T 中取的值。我们将明确地将离散随机变量 X 的pmf写为 $P(X = x)$ 。对于连续随机变量 X （第[6.2.2]节），pdf写为 $f(x)$ ，cdf写为 $F_X(x)$ 。◇

我们将看两种获得随机变量变换分布的方法：一种是使用累积分布函数定义的直接方法，另一种是使用微积分链式法则的变量变换方法（第[5.2.2]节）。变量变换方法[[矩生成]]被广泛使用，因为它为尝试计算由于变换而产生的分布提供了一个“配方”。我们将[[函数也可以]]解释单变量随机变量的技术，并且只简要[[用于研究]]提供多变量随机变量一般情况的结果。[[随机]][[变量的变换]][[（Casella）]]离散随机变量的变换可以直接理解。假设有一个离散随机变量[[和Berger, 2002,]] X ，其pmf为 $P(X = [第2章]) (x)$ （第[6.2.1]节），以及一个可逆函数 $U(x)$ 。考虑变换后的随机变量 $Y := U(X)$ ，其pmf为 $P(Y = y)$ 。那么

$$P(Y = y) = P(U(X) = y) \text{ 感兴趣的变换 (6.125a)} = P(X = U(y)) \text{ 逆变换 (6.125b)}$$

其中我们可以观察到 $-x = U(y)$ 。因此，对于离散随机变量，变换直接改变个别事件（概率相应地变换）。

6.7.1 分布函数技术

分布函数技术回到第一原理，使用累积分布函数 $F_X(x) = P(X \leq x)$ 的定义以及其微分是pdf $f(x)$ [(Wasserman, 2004,] 第2章)的事实。对于随机变量 X 和函数 U ，我们通过以下步骤找到随机变量 $Y := U(X)$ 的pdf：

1. 找到cdf：

$$F[Y](y) = P(Y \leq y) \text{ (6.126)}$$

2. 对cdf $F[Y](y)$ 求导得到pdf $f(y)$ 。

$$df(y) = F[Y](y) \cdot (6.127) dy$$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由 Cambridge University Press (2020) 出版.]

概率和分布

我们还需要记住，由于变换 U 的作用，随机变量的定义域可能已经发生了变化。

例 6.16

设 X 是一个连续随机变量，在 $0 \leq x \leq 1$ 上有概率密度函数

$$f_X = 3x^2. \quad (6.128)$$

我们感兴趣找到 $Y = X^2$ 的 pdf。

函数 f 是 x 的递增函数，因此得到的 y 值位于区间 $[0, 1]$ 内。我们得到

$F_Y = P(Y \leq y)$ cdf 的定义 (6.129a) = $P(X^2 \leq y)$ 所关注的变换 (6.129b) = $P(X \leq y^{1/2})$ 反函数 (6.129c) = F_X cdf 的定义 (6.129d)

$\int_{[0 \text{ to } y^{1/2}]} 3t^2 dt$ 作为定积分的 cdf (6.129e)

= $[t^3] | [t=0 \text{ to } y^{1/2}]$ 积分结果 (6.129f)

= $(3/2)y, 0 \leq y \leq 1.$ (6.129g)

因此， Y 的 cdf 是

$$F_Y = (3/2)y \quad (6.130)$$

对于 $0 \leq y \leq 1$ 。为了得到 pdf，我们对 cdf 求导

$$f_Y = d/dy F_Y = (3/2)y^{1/2} \quad (6.131)$$

对于 $0 \leq y \leq 1$ 。

在例 6.16 中，我们考虑了一个严格单调递增函数 $f(x) = 3x^2$ 。这意味着我们可以计算一个反函数。一般来说，我们要求所关注的函数 $y = U(x)$ 有一个反函数 $x = U^{-1}(y)$ 。通过考虑随机变量 X 的累积分布函数 F_X ，并将其用作变换 $U(x)$ ，可以得到一个有用的结果。这导致了以下定理。

定理 6.15. [Casella and Berger (2002) 中的定理 2.1.10] 设 X 是一个具有严格单调累积分布函数 F_X 的连续随机变量。那么定义为

$$Y := F_X \quad (6.132)$$

的随机变量 Y 具有均匀分布。

定理 6.15 被称为 **概率积分变换**，它用于通过变换从均匀随机变量采样的结果来导出从分布中采样的算法 (Bishop, 2006)。该算法的工作原理是首先从均匀分布中生成一个样本，然后通过反 cdf（假设这是可用的）对其进行变换，以

获得所需分布的样本。概率积分变换也用于假设检验，以确定样本是否来自特定分布 (Lehmann and Romano, 2005)。cdf 的输出给出均匀分布的想法也构成了 copulas 的基础 (Nelsen, 2006)。

变量变换

第 6.7.1 节中的分布函数技术是从第一性原理推导出来的，基于 cdf 的定义并使用反函数、微分和积分的性质。这个来自第一性原理的论证依赖于两个事实：

1. 我们可以将 Y 的 cdf 变换为 X 的 cdf 的表达式。
2. 我们可以对 cdf 求导以获得 pdf。

让我们逐步分解推理，目标是理解定理 6.16 中更一般的变量变换方法。

备注. “变量变换”这个名称来自于在面对困难积分时改变积分变量的想法。对于一元函数，我们使用积分的替换法则，

$$\int f(g(x))g'(x)dx = \int f(u)du, \text{ 其中 } u = g(x). \quad (6.133)$$

这个法则的推导基于微积分的链式法则 (5.32) 并通过两次应用微积分基本定理。微积分基本定理形式化了积分和微分在某种程度上是彼此“逆运算”的事实。通过考虑方程 $u = g(x)$ 的小变化（微分），即考虑 $\Delta u = g'(x)\Delta x$ 作为 $u = g(x)$ 的微分，可以获得对该法则的直观理解。通过替换 $u = g(x)$ ，(6.133) 右侧积分内的参数变为 $f(g(x))$ 。通过假设项 du 可以用 $du \approx \Delta u = g'(x)\Delta x$ 近似，并且 $dx \approx \Delta x$ ，我们得到 (6.133)。◇

考虑一个一元随机变量 X ，以及一个可逆函数 U ，它给我们另一个随机变量 $Y = U(X)$ 。我们假设随机变量 X 有状态 $x \in [a, b]$ 。根据 cdf 的定义，我们有

$$FY = P(Y \leq y). \quad (6.134)$$

我们对随机变量的函数 U 感兴趣

$$P(Y \leq y) = P(U(X) \leq y), \quad (6.135)$$

其中我们假设函数 U 是可逆的。区间上的可逆函数要么严格递增，要么严格递减。在 U 严格递增的情况下，它的反函数 U^{-1} 也是严格递增的。通过对 $P(U(X) \leq y)$ 的参数应用反函数 U^{-1} ，我们得到

$$P(U(X) \leq y) = P(U^{-1}(U(X)) \leq U^{-1}(y)) = P(X \leq U^{-1}(y)). \quad (6.136)$$

(6.136) 中最右边的项是 X 的 cdf 的表达式。回忆用 pdf 表示的 cdf 定义

$$P(X \leq U^{-1}(y)) = \int [a \text{ to } U^{-1}(y)] fXdx. \quad (6.137)$$

现在我们有用了 x 表示的 Y 的 cdf 的表达式：

$$FY = \int [a \text{ to } U^{-1}(y)] fXdx. \quad (6.138)$$

为了得到 pdf，我们对 (6.138) 关于 y 求导：

$$fY = d/dy FY = d/dy \int [a \text{ to } U^{-1}(y)] fXdx. \quad (6.139)$$

注意右边的积分是关于x的，但我们需要关于y的积分，因为我们要对y求微分。特别地，我们使用[(6.133)]得到替换

$$Z Z']$$

$$f(x) = f(U(y)) U'(y) dy = f(x) dx \text{ 其中 } x = U(y). \quad (6.140)$$

在[(6.139)]的右边使用[(6.140)]给出

$$d = [d] Z [U]$$

$$f(x) = f(U(y)) U'(y) dy = f(x) dx \quad (6.141)$$

$$dx$$

$$[d]$$

然后我们回想起微分是一个线性算子，我们使用下标 x 来提醒自己 $f(U(y))$ 是 x 的函数而不是

x

y的函数。再次调用微积分基本定理给出

$$d$$

$$f(x) = f(U(y)) \cdot U'(y). \quad (6.142)$$

$$x dy$$

回想我们假设U是严格递增函数。对于递减函数，当我们遵循相同的推导时，我们会有一个负号。我们引入微分的绝对值，对递增和递减的U都有相同的表达式：

$$d$$

$$f(x) = f(U(y)) \cdot |U'(y)|. \quad (6.143)$$

$$x dy$$

[Mathematics for Machine Learning草稿 (2023-10-18)。反馈：] [https://mml-book.com][.] [6.7 变量变换/逆变换] [219]

这被称为变量变换技术。[(6.143)]中的项 $|d| = |U'(y)|$ 测量应用U时单位体积的变化量（另见第[5.3]节中Jacobian的定义）。

备注。与[(6.125b)]中的离散情况相比，我们有一个额外的因子 $|d| = |U'(y)|$ 。连续情况需要更加小心，因为对所有y， $P(Y=y)=0$ 。概率密度函数f(y)没有涉及y的事件概率的描述。◇

到目前为止在本节中，我们一直在研究单变量变量变换。多变量随机变量的情况是类似的，但由于绝对值不能用于多变量函数而变得复杂。相反，我们使用Jacobian矩阵的行列式。回想[(5.58)]中Jacobian是偏导数矩阵，非零行列式的

存在表明我们可以求Jacobian的逆。回想第[4.1]节中的讨论，行列式出现是因为我们的微分（体积立方）被Jacobian变换为平行六面体。让我们在以下定理中总结前面的讨论，它给出了多变量变量变换的方法。

定理 6.16. [Billingsley (1995)中的定理17.2] 设 $f(x)$ 是多变量连续随机变量 X 的概率密度值。如果向量值函数 $y = U(x)$ 在 x 的定义域内的所有值都是可微且可逆的，那么对于 y 的相应值， $Y = U(X)$ 的概率密度由下式给出

∂

$$f_Y(y) = f_X(U^{-1}(y)) \cdot |\det U^{-1}(y)|. \quad (6.144)$$

$x \partial y$

这个定理乍看起来令人生畏，但关键点是多变量随机变量的变量变换遵循单变量变量变换的过程。首先我们需要求出逆变换，并将其代入 x 的密度。然后我们计算Jacobian的行列式并将结果相乘。以下例子说明了二元随机变量的情况。

例 6.17

X

考虑一个二元随机变量 X ，状态为 $x = (x_1, x_2)$ 且概率密度函数为

x_1, x_2

$f_X(x) = \frac{1}{2\pi} \exp\left(-\frac{1}{2}(x_1^2 + x_2^2)\right)$

$$f_X(x) = \frac{1}{2\pi} \exp\left(-\frac{1}{2}(x_1^2 + x_2^2)\right). \quad (6.145)$$

x_1, x_2

我们使用定理[6.16]中的变量变换技术推导

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由Cambridge University Press出版 (2020).]

[220] [[概率与分布]]

随机变量线性变换（第[2.7]节）的效果。

考虑定义为的矩阵 $A \in \mathbb{R}^{2 \times 2}$

a, b

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}. \quad (6.146)$$

c, d

我们想要找到变换后的二元随机变量 Y 的概率密度函数，其状态为 $y = Ax$ 。

回想对于变量变换，我们需要 x 作为 y 函数的逆变换。由于我们考虑线性变换，逆变换由矩阵逆（见第[2.2.2]节）给出。对于 2×2 矩阵，我们可以明确写出公式，由下式给出

$$x = \frac{1}{ad - bc} (d - by)$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = A^{-1} \quad (6.147)$$

$$x = \frac{1}{ad - bc} (d - by)$$

观察到 $ad - bc$ 是 A 的行列式（第[4.1]节）。相应的概率密度函数由下式给出

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$f(x) = f(y) \det A^{-1} = \exp\left(-\frac{1}{2} y^T A y\right) \quad (6.148)$$

$$\frac{1}{2} \pi$$

矩阵乘以向量对向量的偏导数就是矩阵本身（第[5.5]节），因此

$$\frac{\partial}{\partial y} = A$$

$$A y = A^{-1} \quad [(6.149)]$$

$$\frac{\partial}{\partial y}$$

回想第[4.1]节中逆矩阵的行列式是行列式的逆，所以Jacobian矩阵的行列式是

$$\frac{\partial}{\partial x}$$

$$\det A^{-1} = \frac{1}{\det A} \quad (6.150)$$

$$\frac{\partial}{\partial y} (ad - bc)$$

现在我们能够通过将[(6.148)]与[(6.150)]相乘来应用定理[6.16]中的变量变换公式，得到

$$\frac{\partial}{\partial x}$$

$$f(x) = f(y) \det A^{-1} \quad (6.151a)$$

$$y \frac{\partial}{\partial y}$$

$$= \frac{1}{2} \pi \exp\left(-\frac{1}{2} y^T A y\right) \quad (6.151b) \quad \frac{1}{2} \pi$$

虽然例[6.17]基于二元随机变量，这使我们能够轻松计算矩阵逆，但前述关系对更高维度也成立。

备注。我们在第[6.5]节中看到[(6.148)]中的密度 $f(x)$ 实际上是标准Gaussian分布，变换后的密度 $f(y)$ 是协方差为 $\Sigma = A^{-1}$ 的二元Gaussian分布。◇

我们将使用本章中的思想来描述概率建模

[《机器学习数学基础》草稿 (2023-10-18) 。反馈：] [<https://mml-book.com>][.] [*6.8 进一步阅读*] [[221]]

在第8.4节中，以及在第8.5节中引入图形语言。我们将在第9章和第11章中看到这些思想在机器学习中的直接应用。

6.8 进一步阅读

本章有时相当简洁。[Grinstead and Snell (1997)]和[Walpole et al. (2011)]提供了更轻松的介绍，适合自学。对概率的哲学方面感兴趣的读者应该考虑[Hacking (2001)]，而更与软件工程相关的方法由[Downey (2014)]提出。指数族的概述可以在[Barndorff-Nielsen (2014)]中找到。我们将在第8章中看到更多关于如何使用概率分布来建模机器学习任务的内容。讽刺的是，最近对神经网络兴趣的激增导致了对概率模型更广泛的认识。例如，normalizing flows的思想[(Jimenez Rezende and Mohamed, 2015)]依赖于变换随机变量的变量变换。[Goodfellow et al. (2016)]一书的第16到20章描述了应用于神经网络的变分推理方法概述。

我们通过避免测度论问题[(Billingsley, 1995; Pollard, 2002)]，以及假设我们有实数和实数上定义集合的方法以及它们适当的出现频率而无需构造，从而回避了连续随机变量中的大部分困难。这些细节确实很重要，例如，在连续随机变量 x, y 的条件概率 $p(y | x)$ 的规范中[(Proschan and Presnell, 1998)]。懒惰的记号隐藏了这样一个事实：我们想要指定 $X = x$ （这是一个测度为零的集合）。此外，我们对 y 的概率密度函数感兴趣。更精确的记号必须说 $E[y][f(y) | \sigma(x)]$ ，其中我们取 y 在测试函数 f 上的期望，条件是 x 的 σ 代数。对概率论细节感兴趣的更技术性读者有很多选择[(Jaynes, 2003; MacKay, 2003; Jacod and Protter, 2004; Grimmett and Welsh, 2014)]，包括一些非常技术性的讨论[(Shiryayev, 1984; Lehmann and Casella, 1998; Dudley, 2002; Bickel and Doksum, 2006; Çinlar, 2011)]。处理概率的另一种方法是从期望的概念开始，然后“向后工作”推导概率空间的必要性质[(Whittle, 2000)]。随着机器学习允许我们在越来越复杂的数据类型上建模更复杂的分布，概率机器学习模型的开发者必须理解这些更技术性的方面。专注于概率建模的机器学习教材包括[MacKay (2003); Bishop (2006); Rasmussen and Williams (2006); Barber (2012); Murphy (2012)]的书籍。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).] [222] [[概率与分布]]

练习

6.1 考虑两个离散随机变量X和Y的以下二元分布 $p(x,y)$ 。

	x_1	x_2	x_3	x_4	x_5
y_1	0.01	0.02	0.03	0.1	0.1
y_2	0.05	0.1	0.05	0.07	0.2
y_3	0.1	0.05	0.03	0.05	0.04

X

计算： a. 边际分布 $p(x)$ 和 $p(y)$ 。 b. 条件分布 $p(x|Y=y_1)$ 和 $p(y|X=x_3)$ 。

6.2 考虑两个Gaussian分布的混合（如图6.4所示）：

$$0.4 N([10/1], [0.8/8.4]) + 0.6 N([0/2], [2.0/1.7/1.7])$$

- a. 计算每个维度的边际分布。
- b. 计算每个边际分布的均值、众数和中位数。
- c. 计算二维分布的均值和众数。

6.3 你写了一个计算机程序，有时编译成功，有时不成功（代码不变）。你决定使用参数为 μ 的Bernoulli分布来建模编译器的明显随机性（成功vs不成功） x ：

$$p(x|\mu) = \mu^x(1-\mu)^{1-x}, x \in \{0,1\}。$$

为Bernoulli似然选择一个共轭先验，并计算后验分布 $p(\mu|x_1, \dots, x_n)$ 。

6.4 有两个袋子。第一个袋子装有四个芒果和两个苹果；第二个袋子装有四个芒果和四个苹果。

我们还有一个有偏的硬币，显示”正面”的概率是0.6，显示”反面”的概率是0.4。如果硬币显示”正面”，我们从袋子1中随机选择一个水果；否则我们从袋子2中随机选择一个水果。

你的朋友抛硬币（你看不到结果），从相应的袋子中随机选择一个水果，然后给你一个芒果。

芒果是从袋子2中选出的概率是多少？

提示：使用Bayes定理。

6.5 考虑时间序列模型

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{w}, \mathbf{w} \sim N(0, Q) \quad \mathbf{y}_t = C\mathbf{x}_t + \mathbf{v}, \mathbf{v}_t \sim N(0, R),$$

其中 $\mathbf{w}, \mathbf{v}$ 是i.i.d. Gaussian噪声变量。进一步假设 $p(\mathbf{x}_0) = N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ 。

[《机器学习数学基础》草稿 (2023-10-18)。反馈:] [<https://mml-book.com>][.] [练习] [[223]]

a. $p(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_t)$ 的形式是什么? 证明你的答案 (你不必显式计算联合分布)。

b. 假设 $p(\mathbf{x}_{-t} | \mathbf{y}_{-1}, \dots, \mathbf{y}_{-t}) = N(\boldsymbol{\mu}_{-t}, \boldsymbol{\Sigma}_{-t})$ 。

1. 计算 $p(\mathbf{x}_{-t+1} | \mathbf{y}_{-1}, \dots, \mathbf{y}_{-t})$ 。

2. 计算 $p(\mathbf{x}_{-t+1}, \mathbf{y}_{-t+1} | \mathbf{y}_{-1}, \dots, \mathbf{y}_{-t})$ 。

3. 在时间 $t+1$, 我们观察到值 $\mathbf{y} = \hat{\mathbf{y}}_{-t+1}$ 。计算条件分布 $p(\mathbf{x}_{-t+1} | \mathbf{y}_{-1}, \dots, \mathbf{y}_{-t+1})$ 。

6.6 证明(6.44)中的关系, 该关系将方差的标准定义与方差的原始分数表达式联系起来。

6.7 证明(6.45)中的关系, 该关系将数据集中样本之间的成对差异与方差的原始分数表达式联系起来。

6.8 将Bernoulli分布表示为指数族的自然参数形式, 见(6.107)。

6.9 将二项分布表示为指数族分布。同时将Beta分布表示为指数族分布。证明Beta分布和二项分布的乘积也是指数族的成员。

6.10 用两种方法推导第6.5.2节中的关系:

a. 通过配方

b. 通过将Gaussian表示为其指数族形式

两个Gaussians $N(\mathbf{x} | \mathbf{a}, A) N(\mathbf{x} | \mathbf{b}, B)$ 的乘积是一个未归一化的Gaussian分布 $c N(\mathbf{x} | \mathbf{c}, C)$, 其中

$$C^{-1} = A^{-1} + B^{-1} \quad c = C(A^{-1}\mathbf{a} + B^{-1}\mathbf{b})$$

$$c = (2\pi)^{D/2} |A + B|^{-1/2} \exp[-1/2(\mathbf{a} - \mathbf{b})^T(A + B)^{-1}(\mathbf{a} - \mathbf{b})]$$

注意归一化常数 c 本身可以被认为是 $\mathbf{a}$ 或 $\mathbf{b}$ 的(归一化)Gaussian分布, 具有”膨胀的”协方差矩阵 $A + B$, 即, $c = N(\mathbf{a} | \mathbf{b}, A + B) = N(\mathbf{b} | \mathbf{a}, A + B)$ 。

6.11 迭代期望

考虑两个随机变量 $\mathbf{x}, \mathbf{y}$, 具有联合分布 $p(\mathbf{x}, \mathbf{y})$ 。证明

$$E_{\mathbf{X}}\mathbf{x} = E_{\mathbf{Y}}[E_{\mathbf{X}}[\mathbf{x} | \mathbf{y}]]$$

这里, $E_{\mathbf{X}}[\mathbf{x} | \mathbf{y}]$ 表示在条件分布 $p(\mathbf{x} | \mathbf{y})$ 下 $\mathbf{x}$ 的期望值。

6.12 Gaussian随机变量的操作

考虑 Gaussian 随机变量 $x \sim N(x \mid \mu_x, \Sigma_x)$, 其中 $x \in \mathbb{R}^D$ 。

此外, 我们有

$$y = Ax + b + w,$$

其中 $y \in \mathbb{R}^E$, $A \in \mathbb{R}^{E \times D}$, $b \in \mathbb{R}^E$, $w \sim N(w \mid 0, Q)$ 是独立的 Gaussian 噪声。“独立”意味着 x 和 w 是独立随机变量, 并且 Q 是对角的。

a. 写出似然 $p(y \mid x)$ 。

b. 分布 $p(y) = \int p(y \mid x)p(x)dx$ 是 Gaussian。计算均值 μ_y 和协方差 Σ_y 。详细推导你的结果。

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

224 概率与分布

c. 随机变量 y 根据测量映射进行变换

$$z = Cy + v,$$

其中 $z \in \mathbb{R}^F$, $C \in \mathbb{R}^{F \times E}$, $v \sim N(v \mid 0, R)$ 是独立的 Gaussian (测量) 噪声。

写出 $p(z \mid y)$ 。计算 $p(z)$, 即均值 μ_z 和协方差 Σ_z 。详细推导你的结果。

d. 现在, 测量到一个值 $\hat{y}$ 。计算后验分布 $p(x \mid \hat{y})$ 。

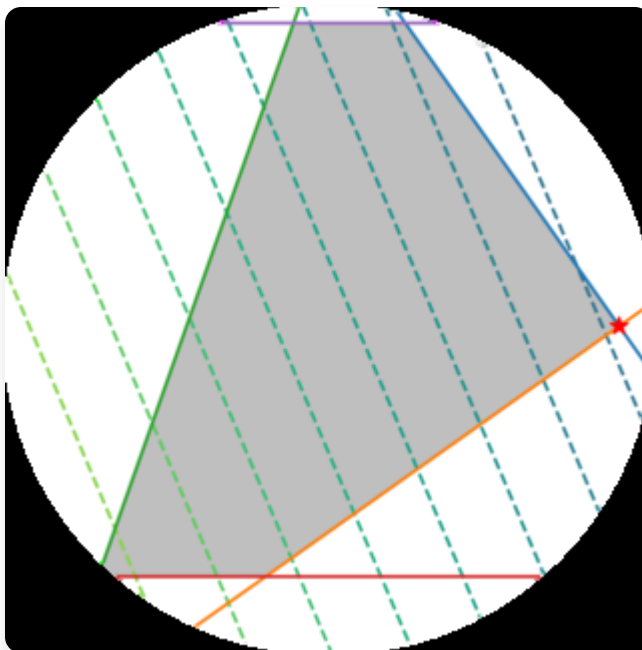
解答提示: 这个后验也是 Gaussian, 即我们只需要确定其均值和协方差矩阵。从显式计算联合 Gaussian $p(x, y)$ 开始。这也需要我们计算交叉协方差 $\text{Cov}[x, y]$ 和 $\text{Cov}[y, x]$ 。然后应用 Gaussian 条件化的规则。

6.13 概率积分变换

给定连续随机变量 X , 具有 cdf $F_X(x)$, 证明随机变量 $Y := F_X(X)$ 是均匀分布的 (定理 6.15)。

Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>.

连续优化



由于机器学习算法是在计算机上实现的，数学公式被表达为数值优化方法。本章描述了训练机器学习模型的基本数值方法。训练机器学习模型通常归结为找到一组好的参数。“好”的概念由目标函数或概率模型确定，我们将在本书第二部分看到相关示例。给定目标函数，使用优化算法找到最佳值。由于我们考虑 $\mathbb{R}^D$ 中的数据和模型，我们面临的问题是连续优化问题，而不是组合优化问题。

本章涵盖连续优化的两个主要分支(图7.1)：无约束优化和约束优化。我们在本章中假设我们的目标函数是可微的(见第5章)，因此我们可以在空间中的每个位置访问梯度来帮助我们找到最优值。按照惯例，机器学习中的大多数目标函数都是要被最小化的，即最佳值是最小值。直观地说，找到最佳值就像找到最小值一样。

目标函数的谷，梯度指向我们上坡的方向。优化的思想是向下坡方向移动（与梯度相反）并希望找到最深的点。对于无约束优化，这是我们需要的唯一概念，但有几种设计选择，我们将在第7.1节中讨论。对于约束优化，我们需要引入其他概念来管理约束（第7.2节）。我们还将介绍一类特殊的问题（第7.3节中的凸优化问题），在这类问题中我们可以对达到全局最优做出陈述。

考虑图7.2中的函数。该函数在 $x = -4.5$ 附近有一个**全局最小值**，函数值约为 -47 。由于函数是“光滑的”，梯度可以通过指示我们应该向右还是向左迈步来帮助找到最小值。这假设我们在正确的谷中，因为在 $x = 0.7$ 附近还存在另一个**局部最小值**。回想一下，我们可以通过计算函数的导数并将其设为零来求解函数的所有驻点。

对于

$$\ell(x) = x^4 + 7x^3 + 5x^2 - 17x + 3, (7.1)$$

我们得到相应的梯度为

$$d\ell(x)/dx = 4x^3 + 21x^2 + 10x - 17. \quad (7.2)$$

图7.1 本章中与优化相关概念的思维导图。有两个主要思想：梯度下降和凸优化。

[图7.1显示了一个思维导图，包含以下概念分支：- 连续优化 - 无约束优化 - 梯度下降 - 步长 - 动量 - 随机梯度下降 - 约束优化 - 拉格朗日乘数 - 凸优化 - 凸优化与对偶 - 凸共轭 - 线性规划 - 二次规划 - 连接到第10章（降维） - 连接到第11章（密度估计） - 连接到第12章（分类）]

由于这是一个三次方程，当设为零时通常有三个解。在这个例子中，其中两个是最小值，一个是最大值（在 $x = -1.4$ 附近）。要检查一个驻点是最小值还是最大值，我们需要对其求二阶导数，并检查二阶导数在驻点处是正还是负。在我们的情况下，二阶导数是

$$d^2\ell(x)/dx^2 = 12x^2 + 42x + 10. \quad (7.3)$$

通过代入我们目测估计的值 $x = -4.5, -1.4, 0.7$ ，我们将观察到正如预期的那样，中间的点是一个最大值（ $d^2\ell(x)/dx^2 < 0$ ），其他两个驻点是最小值。

注意，在前面的讨论中我们避免了解析求解 x 的值，尽管对于像前面这样的低阶多项式我们可以这样做。一般来说，我们无法找到解析解，因此我们需要从某个值开始，比如 $x_0 = -6$ ，然后沿着负梯度方向前进。负梯度表明我们应该

图7.2 示例目标函数。负梯度用箭头表示，全局最小值用蓝色虚线表示。

[图7.2显示了函数 $\ell(x) = x^4 + 7x^3 + 5x^2 - 17x + 3$ 的图像， x 轴从-6到2， y 轴从-60到60，显示了函数的形状和梯度方向]

向右走，但不知道走多远（这被称为步长）。此外，如果我们从右侧开始（例如， $x_0 = 0$ ），负梯度会将我们引导到错误的局部最小值。图7.2说明了这样一个事实：对于 $x > -1$ ，负梯度指向图中右侧的最小值，该最小值具有更大的目标值。

根据Abel-Ruffini定理，一般来说5次或更高次的多项式没有代数解（Abel, 1826）。

在第7.3节中，我们将学习一类称为凸函数的函数，这类函数不会表现出这种对优化算法起始点的复杂依赖性。对于凸函数，所有局部最小值都是全局最小值。事实证明，许多机器学习目标函数都设计成凸函数，我们将在第12章中看到例子。

本章到目前为止的讨论是关于一维函数的，在这里我们能够可视化梯度、下降方向和最优值的概念。在本章的其余部分，我们将在高维中发展相同的思想。不幸的是，我们只能在一维中可视化这些概念，但有些概念不能直接推广到高维，因此在阅读时需要小心。

7.1 使用梯度下降的优化

我们现在考虑求解实值函数最小值的问题

$$\min_{\mathbf{x}} f(\mathbf{x}), \quad (7.4)$$

其中 $f: \mathbb{R}^d \rightarrow \mathbb{R}$ 是一个目标函数，它捕获了机器

learning problem at hand. 我们假设函数 f 是可微的，并且无法通过解析方法找到闭式解。

梯度下降是一种一阶优化算法。为了使用梯度下降找到函数的局部最小值，我们沿着当前点处函数梯度的负方向按比例步进。

[我们使用] [回顾第5.1]节，梯度指向[行向量的] [最陡上升方向。另一个有用的直觉是考虑函数值为某个常数 ($f(\mathbf{x}) = c$, 其中 $c \in \mathbb{R}$) 的等高线集合，] [约定来表示] [这些线被称为等高线。梯度指向垂直于我们希望优化的函数等高线的方向。] [梯度。]

让我们考虑多变量函数。想象一个由函数 $f(\mathbf{x})$ 描述的曲面，上面有一个球从特定位置 $\mathbf{x}[0]$ 开始。当球被释放时，它会沿最陡下降方向向下滚动。梯度下降利用了这样一个事实：如果从 $\mathbf{x}[0]$ 沿负梯度方向 $-(\nabla f)(\mathbf{x}[0])$ 移动， $f(\mathbf{x}[0])$ 下降最快。我们在本书中假设函数是可微的，并在第7.4节中介绍更一般的情况。那么，如果

$$\mathbf{x}[1][T] = \mathbf{x}[0] - \gamma (\nabla f)(\mathbf{x}[0]) T \quad (7.5)$$

对于小的步长 $\gamma \geq 0$ ，则 $f(\mathbf{x}[1]) \leq f(\mathbf{x}[0])$ 。注意我们对梯度使用转置，否则维度不匹配。

这个观察使我们能够定义一个简单的梯度下降算法：如果我们要找到函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ， $\mathbf{x} \mapsto f(\mathbf{x})$ 的局部最优解 $\mathbf{x}^*$ ，我们从要优化参数的初始猜测 $\mathbf{x}[0]$ 开始，然后按照以下方式迭代：

$$\mathbf{x}[i+1][T] = \mathbf{x}[i] - \gamma \nabla f(\mathbf{x}[i]) \quad (7.6)$$

对于合适的步长 $\gamma[i]$ ，序列 $f(\mathbf{x}[0]) \geq f(\mathbf{x}[1]) \geq \dots$ 收敛到局部最小值。

例 7.1

考虑二维二次函数

$$f(\begin{bmatrix} x_1/x[2] \end{bmatrix}) = \begin{bmatrix} x_1 & x[2] \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 20 \end{bmatrix} \begin{bmatrix} x_1/x[2] \end{bmatrix} - 5 \begin{bmatrix} x_1/x[2] \end{bmatrix} + 3 \quad (7.7)$$

其梯度为

$$\nabla f(\begin{bmatrix} x_1/x[2] \end{bmatrix}) = \begin{bmatrix} x_1 & x[2] \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 20 \end{bmatrix} - 5 \quad (7.8)$$

从初始位置 $\mathbf{x}[0][T] = [-3, -1]$ 开始，我们迭代应用(7.6)得到收敛到最小值的估计序列



图7.3 在二维二次曲面上的梯度下降（以热图显示）。参见例7.1的描述。

图7.3 在二维二次曲面上的梯度下降（以热图显示）。参见例7.1的描述。

（如图7.3所示）。我们可以看到（从图中和将 $x[0]$ 代入(7.8)， $\gamma = 0.085$ ）， $x[0]$ 处的负梯度指向北方和东方，导致 $x[1][T] = [-1.98, 1.21]$ 。重复这个过程得到 $x[2][T] = [-1.32, -0.42]$ ，依此类推。

备注。梯度下降在接近最小值时可能相对较慢：其渐近收敛速度不如许多其他方法。使用球滚下山的类比，当曲面是一个狭长的山谷时，问题是病态的(Trefethen and Bau III, 1997)。对于病态凸问题，梯度下降逐渐“之字形”前进，因为梯度几乎垂直于到最小值点的最短方向；见图7.3。◇

7.1.1 步长

如前所述，在梯度下降中选择合适的步长很重要。如果步长太小，梯度下降可能很慢。如果步长选择得太大，梯度下降可能会超调、无法收敛，甚至发散。我们将在下一节讨论momentum的使用。这是一种平滑梯度更新不稳定行为并阻尼震荡的方法。

自适应梯度方法根据函数的局部性质在每次迭代时重新调整步长。有两个简单的启发式方法(Toussaint, 2012):

当梯度步进后函数值增加时，步长太大。撤销该步并减小步长。

当函数值减少时，步长本可以更大。尝试增加步长。

虽然”撤销”步骤似乎是资源浪费，但使用这种启发式方法保证单调收敛。

例 7.2（求解线性方程组）

当我们求解形式为 $Ax = b$ 的线性方程时，在实践中我们通过找到最小化平方误差的 x^* 来近似求解 $Ax - b = 0$

$$\|Ax - b\|^2 = (Ax - b)^T(Ax - b) \quad (7.9)$$

如果我们使用欧几里得范数。(7.9)关于 x 的梯度是

$$\nabla_x = 2(Ax - b)^T A \quad (7.10)$$

我们可以在梯度下降算法中直接使用这个梯度。然而，对于这个特殊情况，事实证明存在解析解，可以通过将梯度设为零来找到。我们将在第9章看到更多关于求解平方误差问题的内容。

备注。当应用于求解线性方程组 $Ax = b$ 时，梯度下降可能收敛缓慢。梯度下降的收敛速度依赖于条件数 $\kappa = \sigma_{\max}(A) / \sigma_{\min}(A)$ ，这是 A 的最大奇异值与最小奇异值的比值（第4.5节）。条件数本质上度量了最

曲线方向与最不曲线方向相比，这对应于我们对病态问题的形象化理解，即它们是长而细的山谷：它们在一个方向上非常弯曲，但在另一个方向上非常平坦。与其直接求解 $Ax = b$ ，不如求解 $P(Ax - b) = 0$ ，其中 P^{-1} 称为预条件器(preconditioner)。目标是设计 P 使得 PA 具有更好的条件数，但同时 P^{-1} 易于计算。有关梯度下降、预条件和收敛的更多信息，我们参考 [Boyd and Vandenberghe (2004), 第9章]。◇

7.1.2 带动量的梯度下降

如图 7.3 所示，如果优化表面的曲率使得存在缩放不良的区域，梯度下降的收敛可能非常缓慢。曲率使得梯度下降步骤在山谷壁之间跳跃，并以小步长接近最优值。为改善收敛而提出的调整是给梯度下降一些记忆。

[Goh (2017) 写了一篇关于带动量梯度下降的直观博客文章] 带动量的梯度下降 [(Rumelhart et al., 1986)] 是一种引入额外项来记住前一次迭代中发生了什么的方法。这种记忆抑制振荡并平滑梯度更新。继续使用球的类比，动量项模拟了一个不愿改变方向的重球现象。其思想是通过记忆来实现梯度更新，以实现移动平均。基于动量的方法记住每次迭代 i 时的更新 Δx_i ，并将下一次更新确定为当前和先前梯度的线性组合

$$x_{i+1} = x_i - \gamma(\nabla f(x_i)) + \alpha \Delta x_i \quad (7.11)$$

$$\Delta x_i = x_i - x_{i-1} = \alpha \Delta x_{i-1} - \gamma(\nabla f(x_{i-1})), \quad (7.12)$$

其中 $\alpha \in [0, 1]$ 。有时我们只能近似知道梯度。在这种情况下，动量项很有用，因为它平均了梯度的不同噪声估计。获得近似梯度的一种特别有用的方法是使用随机近似，我们接下来讨论这个问题。

7.1.3 随机梯度下降

计算梯度可能非常耗时。然而，通常可以找到梯度的“便宜”近似。只要近似梯度大致指向与真实梯度相同的方向，近似梯度仍然有用。

随机梯度下降(stochastic gradient descent) (通常简称为SGD) 是梯度下降方法的随机近似，用于最小化写成可微函数和的目标函数。这里的随机指的是我们承认我们不能精确知道梯度，而只知道它的噪声近似。通过约束近似梯度的概率分布，我们仍然可以在理论上保证SGD会收敛。

在机器学习中，给定 $n = 1, \dots, N$ 个数据点，我们通常考虑目标函数是每个样本 n 产生的损失 L_n 的和。在数学记号中，我们有形式

$$L(\theta) = \sum_{n=1}^N L_n(\theta), \quad (7.13)$$

其中 θ 是感兴趣的参数向量，即我们想找到最小化 L 的 θ 。回归（第9章）中的一个例子是负对数似然，它表示为各个样本对数似然的和

$$L(\theta) = -\sum_{n=1}^N \log p(y_n | x_n, \theta), \quad (7.14)$$

其中 $x_n \in \mathbb{R}^D$ 是训练输入， y_n 是训练目标， θ 是回归模型的参数。

如前所述，标准梯度下降是一种“批量”优化方法，即使用完整训练集进行优化，通过以下方式更新参数向量

$$\theta_{i+1} = \theta_i - \gamma_i (\nabla L(\theta_i)) = \theta_i - \gamma_i \sum_{n=1}^N (\nabla L_n(\theta_i)) \quad (7.15)$$

对于合适的步长参数 γ_i 。评估和梯度可能需要昂贵地评估所有单个函数 L_n 的梯度。当训练集庞大和/或不存在简单公式时，评估梯度和变得非常昂贵。

考虑 (7.15) 中的项 $\sum_{n=1}^N (\nabla L_n(\theta_i))$ 。我们可以通过对较小的 L_n 集合求和来减少计算量。与使用所有 $n = 1, \dots, N$ 的 L_n 的批量梯度下降相比，我们随机选择 L_n 的子集进行小批量梯度下降。在极端情况下，我们随机选择只有一个 L_n 来估计梯度。

为什么取数据子集是合理的，关键洞察是意识到为了梯度下降收敛，我们只需要梯度是真实梯度的无偏估计。实际上，(7.15) 中的项 $\sum_{n=1}^N (\nabla L_n(\theta_i))$ 是梯度期望值（第6.4.1节）的经验估计。因此，期望值的任何其他无偏经验估计，例如使用数据的任何子样本，都足以保证梯度下降的收敛。

备注。当学习率以适当的速度递减，并且在相对温和的假设下，随机梯度下降几乎必然收敛到局部最小值 [(Bottou, 1998)]。◇

为什么要考虑使用近似梯度？一个主要原因是实际实现约束，例如中央

处理单元 (CPU)/图形处理单元 (GPU) 内存或计算时间的限制。我们可以用与估算经验均值时考虑样本大小相同的方式来思考用于估计梯度的子集大小（第6.4.1节）。大的mini-batch大小将提供准确的梯度估计，减少参数更新中的方差。此外，大的mini-batch利用了成本和梯度向量化实现中高度优化的矩阵运算。方差的减少导致更稳定的收敛，但每次梯度计算将更加昂贵。

相比之下，小的mini-batch估计速度快。如果我们保持mini-batch大小较小，梯度估计中的噪声将允许我们摆脱一些糟糕的局部最优解，否则我们可能会陷入其中。

在机器学习中，优化方法通过最小化训练数据上的目标函数来用于训练，但总体目标是提高泛化性能（第8章）。由于机器学习中的目标不一定需要目标函数最小值的精确估计，使用mini-batch方法的近似梯度已被广泛使用。随机梯度下降在大规模机器学习问题中非常有效 (Bottou et al., 2018)，

图7.4 约束优化的说明。无约束问题（由等高线表示）在右侧有一个最小值（由圆圈表示）。边界约束（ $-1 \leq x \leq 1$ 和 $-1 \leq y \leq 1$ ）要求最优解在边界框内，导致最优值由星号表示。

例如在数百万图像上训练深度神经网络 (Dean et al., 2012)，主题模型 (Hoffman et al., 2013)，强化学习 (Mnih et al., 2015)，或大规模Gaussian过程模型的训练 (Hensman et al., 2013; Gal et al., 2014)。

7.2 约束优化和Lagrange乘数

在前一节中，我们考虑了求解函数最小值的问题

$$\min f(x), \quad (7.16)$$

其中 $f: \mathbb{R}^D \rightarrow \mathbb{R}$ 。

在本节中，我们有额外的约束。也就是说，对于实值函数 $g_i: \mathbb{R}^D \rightarrow \mathbb{R}$, $i = 1, \dots, m$ ，我们考虑约束优化问题（见图7.4的说明）

$$\min f(x) \quad (7.17) \text{ subject to } g_i(x) \leq 0 \text{ for all } i = 1, \dots, m.$$

值得指出的是，函数 f 和 g_i 一般情况下可能是非凸的，我们将在下一节考虑凸情况。

将约束问题(7.17)转换为无约束问题的一种显而易见但不太实用的方法是使用指示函数

$$J(x) = f(x) + \sum_{i=1}^m 1(g_i(x)), \quad (7.18)$$

其中 $1(z)$ 是无限阶跃函数

$$1(z) = \begin{cases} 0 & \text{if } z \leq 0 \\ \infty & \text{otherwise} \end{cases} \quad (7.19)$$

如果不满足约束条件，这将给出无限惩罚，因此将提供相同的解。然而，这个无限阶跃函数同样难以优化。我们可以通过引入Lagrange乘数来克服这个困难。Lagrange乘数的思想是用线性函数替换阶跃函数。

我们通过引入对应于每个不等式约束的Lagrange乘数 $\lambda_i \geq 0$ ，将问题(7.17)关联到Lagrangian (Boyd and Vandenberghe, 2004, 第4章)，使得

$$L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) \quad (7.20a) = f(x) + \lambda^T g(x), \quad (7.20b)$$

其中在最后一行中，我们将所有约束 $g_i(x)$ 连接成向量 $g(x)$ ，将所有Lagrange乘数连接成向量 $\lambda \in \mathbb{R}^m$ 。

我们现在引入Lagrangian对偶性的概念。一般来说，优化中的对偶性是将一组变量 x （称为原始变量）中的优化问题转换为另一组不同变量 λ （称为对偶变量）中的另一个优化问题的思想。我们介绍两种不同的对偶性方法：在本节中，我们讨论Lagrangian对偶性；在第7.3.3节中，我们讨论Legendre-Fenchel对偶性。

定义7.1. 问题(7.17)中的

$$\min f(x) \quad (7.21) \text{ subject to } g_i(x) \leq 0 \text{ for all } i = 1, \dots, m$$

被称为原始问题，对应于原始变量 x 。相关的Lagrangian对偶问题为

$$\max D(\lambda) \quad (7.22) \lambda \in \mathbb{R}^m \text{ subject to } \lambda \geq 0,$$

其中 λ 是对偶变量， $D(\lambda) = \min_{x \in \mathbb{R}^D} L(x, \lambda)$ 。

备注. 在定义7.1的讨论中，我们使用了两个独立感兴趣的概念 (Boyd and Vandenberghe, 2004)。首先是minimax不等式，它说对于任何具有两个参数的函数 $\varphi(x, y)$ ，maximin小于minimax，即

$$\max_x \min_y \varphi(x, y) \leq \min_y \max_x \varphi(x, y). \quad (7.23)$$

这个不等式可以通过考虑不等式来证明

$$\text{对于所有 } x, y: \min_y \varphi(x, y) \leq \max_x \varphi(x, y). \quad (7.24)$$

注意，对[(7.24)]左侧取 y 的最大值保持了不等式成立，因为该不等式对所有 y 都成立。类似地，我们可以对[(7.24)]右侧取 x 的最小值来得到[(7.23)]。

第二个概念是弱对偶性，它使用[(7.23)]来表明[[weak duality]]原问题的值总是大于或等于对偶问题的值。这在[(7.27)]中有更详细的描述。◇

回忆[(7.18)]中的 $J(x)$ 和[(7.20b)]中的拉格朗日函数的区别在于我们将指示函数松弛为线性函数。因此，当 $\lambda \geq 0$ 时，拉格朗日函数 $L(x, \lambda)$ 是 $J(x)$ 的下界。因此， $L(x, \lambda)$ 关于 λ 的最大值为

$$J(x) = \max_{\lambda \geq 0} L(x, \lambda). \quad (7.25)$$

回忆原问题是最小化 $J(x)$ ，

$$\min_{x \in [R][d]} \max_{\lambda \geq 0} L(x, \lambda). \quad (7.26)$$

根据minimax不等式[(7.23)]，交换最小值和最大值的顺序会得到更小的值，即

$$\min_{x \in [R][d]} \max_{\lambda \geq 0} L(x, \lambda) \geq \max_{\lambda \geq 0} \min_{x \in [R][d]} L(x, \lambda). \quad (7.27)$$

这也被称为弱对偶性。注意右侧的内部是对偶目标函数 $D(\lambda)$ ，定义如下。[[weak duality]]

与有约束的原优化问题相比， $\min_{x \in [R][d]} L(x, \lambda)$ 对于给定的 λ 值是一个无约束优化问题。如果求解 $\min_{x \in [R][d]} L(x, \lambda)$ 很容易，那么整个问题就容易求解。我们可以从[(7.20b)]观察到 $L(x, \lambda)$ 关于 λ 是仿射的。因此 $\min_{x \in [R][d]} L(x, \lambda)$ 是 λ 的仿射函数的逐点最小值，因此 $D(\lambda)$ 是凹函数，即使 $f(\cdot)$ 和 g_i 可能是非凸的。外层问题，关于 λ 的最大化，是凹函数的最大值，可以高效计算。

假设 $f(\cdot)$ 和 g_i 是可微的，我们通过对拉格朗日函数关于 x 求微分，设微分为零，并求解最优值来找到拉格朗日对偶问题。我们将在第[7.3.1]节和第[7.3.2]节中讨论两个具体例子，其中 $f(\cdot)$ 和 g_i 是凸函数。

备注（等式约束）。 考虑[(7.17)]带有额外等式约束

$$\min f(x) \quad x \in \{x \mid \text{calibre66}\}$$

$$\text{subject to } (7.28) \quad g_i \leq 0 \text{ for all } i = 1, \dots, m \quad h_j = 0 \text{ for all } j = 1, \dots, n.$$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[236] [[连续优化]]

我们可以通过将等式约束替换为两个不等式约束来建模等式约束。也就是说，对于每个等式约束 $h_j = 0$ ，我们等价地将其替换为两个约束 $h_j \leq 0$ 和 $h_j \geq 0$ 。结果表明，相应的拉格朗日乘数是无约束的。

因此，我们约束[(7.28)]中对应不等式约束的拉格朗日乘数为非负，而让对应等式约束的拉格朗日乘数保持无约束。◇

7.3 凸优化

我们将注意力集中在一类特别有用的优化问题上，在该类问题中我们可以保证全局最优性。当 $f(\cdot)$ 是凸函数，且涉及 $g(\cdot)$ 和 $h(\cdot)$ 的约束是凸集时，[convex optimization]这被称为**凸优化问题**。在这种情况下，我们有**强对偶性**：对偶问题的最优解与[problem]原问题的最优解相同。凸函数和凸集之间的区别在机器学习文献中通常没有严格[strong duality]表述，但人们往往可以从上下文推断出隐含的含义。

定义7.2. 集合 C 是**凸集**，如果对于任意 $x, y \in C$ 和任意标量 θ ，其中 $0 \leq \theta \leq 1$ ，我们有[convex set]

$$\theta x + (1 - \theta)y \in C. \quad [(7.29)]$$

图7.5 凸集的例子。凸集是这样的集合：连接集合中任意两个元素的直线都位于集合内部。图[7.5]和[7.6]分别说明了凸集和非凸集。

凸函数是这样的函数：函数上任意两点之间的直线都位于函数上方。图[7.2]显示了一个非凸函数，图[7.3]显示了一个凸函数。另一个凸函数显示在图[7.7]中。

图7.6 非凸集的例子。

定义7.3. 设函数 $f: \mathbb{R}^D \rightarrow \mathbb{R}$ 是一个定义域为凸集的函数。如果对于 f 定义域中的所有 x, y ，以及任意标量 θ ，其中 $0 \leq \theta \leq 1$ ，我们有

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y). \quad [(7.30)]$$

则函数 f 是**凸函数**。

备注。凹函数是凸函数的负值。◇ [convex function]

[(7.28)]中涉及 $g(\cdot)$ 和 $h(\cdot)$ 的约束在标量值处截断函数，产生集合。凸函数和[concave function]凸集之间的另一个关系是考虑通过”填充”凸函数得到的集合。凸函数是一个碗状物体，我们想象向其中倒水将其填满。这个填充后的集合，称为凸函数的**上图**，是一个凸集。[epigraph]

如果函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 是可微的，我们可以用

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>.] [7.3 凸优化] [237]

图7.7 凸函数的例子。

$$y = 3x^5 - x + 2$$



图表显示凸函数曲线

图表显示凸函数曲线

其梯度 $\nabla_x f(x)$ （第[5.2]节）来刻画凸性。函数 $f(x)$ 是凸的当且仅当对于任意两点 x, y 都有

$$f(y) \geq f(x) + \nabla f(x)^T(y - x). \quad (7.31)$$

如果我们进一步知道函数 $f(x)$ 是二次可微的，即 Hessian $\nabla^2 f(x)$ 对于 x 域中的所有值都存在，那么函数 $f(x)$ 是凸函数当且仅当 $\nabla^2 f(x)$ 是正半定的 [Boyd and Vandenberghe, 2004].

示例 7.3

负熵函数 $f(x) = x \log_2 x$ 在 $x > 0$ 时是凸函数。该函数的可视化如图 [7.8] 所示，我们可以看到该函数是凸的。为了说明凸性的先前定义，让我们检查两个点 $x = 2$ 和 $x = 4$ 的计算。注意，要证明 $f(x)$ 的凸性，我们需要检查所有点 $x \in \mathbb{R}$ 。

回顾定义 [7.3]。考虑两点中间的一点（即 $\theta = 0.5$ ）；那么左边是 $f(0.5 \cdot 2 + 0.5 \cdot 4) = 3 \log_2 3 \approx 4.75$ 。右边是 $0.5(2 \log_2 2) + 0.5(4 \log_2 4) = 1 + 4 = 5$ 。因此满足定义。

由于 $f(x)$ 是可微的，我们可以另外使用 [(7.31)]。计算 $f(x)$ 的导数，我们得到

$$\nabla f(x) = 1 - \log_2 x \quad (7.32)$$

$$\nabla f(2) = 1 - \log_2 2 = 0$$

使用相同的两个测试点 $x = 2$ 和 $x = 4$ ，[(7.31)] 的左边由 $f(3) \approx 4.75$ 给出。右边是

$$f(2) + \nabla f(2)^T(4 - 2) = 1 + 0 = 1 \quad (7.33a)$$

$$= 1 + 0 = 1 < 4.75 \quad (7.33b)$$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

图 7.8 负熵函数 $x \log_2 x$ （该函数是凸的）及其在 $x = 2$ 处的切线。

我们可以通过回顾定义从第一原理检查函数或集合是否为凸。在实践中，我们经常依靠保持凸性的操作来检查特定函数或集合是否为凸。尽管细节差异很大，这又是我们在第 [2] 章中为向量空间引入的闭包思想。

示例 7.4

凸函数的非负加权和是凸的。观察到如果 f 是凸函数， $\alpha \geq 0$ 是非负标量，那么函数 αf 是凸的。我们可以通过将 α 乘以定义 [7.3] 中等式的两边来看到这一点，并回忆乘以非负数不会改变不等式。

如果 f_1 和 f_2 是凸函数，那么根据定义我们有

$$f_1(\theta x + (1 - \theta)y) \leq \theta f_1(x) + (1 - \theta)f_1(y) \quad (7.34)$$

将两边相加得到

$$f_1 + f_2 \leq \theta(f_1(x) + f_1(y)) + (1 - \theta)(f_1(x) + f_1(y)) \quad (7.35)$$

其中右边可以重新排列为

$$\theta(f_1 + f_2)(x) + (1 - \theta)(f_1 + f_2)(y) \quad (7.36)$$

完成了凸函数之和是凸的证明。

结合前面的两个事实，我们看到对于 $\alpha, \beta \geq 0$, $\alpha f_1 + \beta f_2$ 是凸的。这个闭包性质可以使用类似的论证扩展到两个以上凸函数的非负加权和。

备注。[(7.30)] 中的不等式有时称为 *Jensen* 不等式。实际上，一整类关于凸函数非负加权和的不等式都称为 Jensen 不等式。◇

总之，如果约束优化问题被称为 *凸优化问题*

$\min f(x) \quad x$

约束条件为 (7.38) $g_i(x) \leq 0$ 对于所有 $i = 1, \dots, m$ $h_j(x) = 0$ 对于所有 $j = 1, \dots, n$,

其中所有函数 $f(x)$ 和 g_i 都是凸函数，所有 $h_j = 0$ 都是凸集合。下面，我们将描述两类广泛使用且被充分理解的凸优化问题。

7.3.1 线性规划

考虑当所有前述函数都是线性的特殊情况，即

$$\min_{x \in \mathbb{R}^d} c^T x \quad (7.39)$$

约束条件为 $Ax \leq b$ ，

其中 $A \in \mathbb{R}^{m \times d}$ 和 $b \in \mathbb{R}^m$ 。这被称为 *线性规划*。它有 d 个变量和 m 个线性约束。Lagrangian 由以下给出

$$L(x, \lambda) = c^T x + \lambda^T (Ax - b), \quad (7.40)$$

其中 $\lambda \in \mathbb{R}^m$ 是非负 Lagrange 乘数向量。重新排列对应于 x 的项得到

$$L(x, \lambda) = (c + A^T \lambda)^T x - \lambda^T b. \quad (7.41)$$

对 $L(x, \lambda)$ 关于 x 求导并设为零得到

$$c + A^T \lambda = 0. \quad (7.42)$$

因此，对偶 Lagrangian 是 $D(\lambda) = -\lambda^T b$ 。回忆我们想要最大化 $D(\lambda)$ 。除了由于 $L(x, \lambda)$ 的导数为零的约束外，我们还有 $\lambda \geq 0$ 的事实，产生以下对偶优化问题

$$\max_{\lambda \geq 0} -\lambda^T b \quad (7.43)$$

约束条件为 $c + A^T \lambda = 0, \lambda \geq 0$ 。

这也是一个线性规划，但有 m 个变量。我们可以选择求解原问题 [(7.39)] 或对偶问题 [(7.43)]，这取决于 m 或 d 哪个更大。回忆 d 是变量数， m 是原线性规划中的约束数。

示例 7.5 (线性规划)

考虑线性规划

$$\min_{x \in \mathbb{R}^2} 5x_1 - 3x_2 \quad (7.44)$$

约束条件为 $2x_1 - 4x_2 \leq 5, -x_1 + x_2 \leq 0, -1 \leq x_1 \leq 1, -8 \leq x_2 \leq 8$

使用两个变量。该 program 也显示在图 [7.9] 中。目标

函数是线性的，产生线性等高线。标准形式的约束集在

图例中被翻译。最优值必须位于

阴影（可行）区域内，并用星号表示。

图7.9 线性program的图解。无约束问题（由等高线表示）在右侧有一个最小值。

给定约束的
最优值用星号
显示。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>] 7.3 Convex Optimization [241]

7.3.2 二次规划

考虑凸二次目标函数的情况，其中约束是仿射的，即

$$\min_{x \in \mathbb{R}^d} \frac{1}{2} x^T Q x + c^T x \quad (7.45)$$

受约束 $Ax \leq b$,

其中 $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, 和 $c \in \mathbb{R}^d$. 对称方阵 $Q \in \mathbb{R}^{d \times d}$ 是正定的，因此目标函数是凸的。这被称为二次program。注意它有d个变量和m个线性约束。

例7.6 (二次program)

考虑二次program

$$\min_{x \in \mathbb{R}^2} \frac{1}{2} x^T \begin{bmatrix} 2 & 1 \\ 1 & 5 \end{bmatrix} x + (7.46) \quad \text{s.t.} \quad \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix} x \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

受约束 $\leq (7.47)$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

包含两个变量。该program也在图[7.4]中有说明。目标函数是带有正半定矩阵Q的二次函数，产生椭圆等高线。最优值必须位于阴影（可行）区域内，并用星号表示。

Lagrangian给出为

$$L(x, \lambda) = \frac{1}{2} x^T Q x + c^T x + \lambda^T (Ax - b) \quad (7.48a)$$

$$= \frac{1}{2} x^T Q x + (c + A^T \lambda)^T x - \lambda^T b, \quad (7.48b)$$

其中我们再次重新排列了项。对L(x, λ)关于x求导并设为零，得到

$$Qx + (c + A^T \lambda) = 0. \quad (7.49)$$

假设Q是可逆的，我们得到

$$x = -Q^{-1} (c + A^T \lambda). \quad (7.50)$$

将[(7.50)]代入原始Lagrangian L(x, λ)，我们得到对偶Lagrangian

$$D(\lambda) = -\frac{1}{2} (c + A^T \lambda)^T Q^{-1} (c + A^T \lambda) - \lambda^T b. \quad (7.51)$$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[连续优化]

因此，对偶优化问题给出为

$$[\lambda] \max \frac{1}{2} [\tau] [\tau] - [\tau] [\tau] (-c + A\lambda) Q (c + A\lambda) [\epsilon] [m] - \lambda b [R] 2 \quad (7.52)$$

受约束 $\lambda \geq 0$.

我们将在第[12]章看到二次规划在机器学习中的应用。

7.3.3 Legendre-Fenchel变换和凸共轭

让我们重新审视第[7.2]节中的对偶思想，不考虑约束。关于凸集的一个有用事实是，它可以通过其支撑超平面等价地描述。如果一个超平面与凸集相交，且凸集包含在超平面的一侧，则称该超平面为凸集的**支撑超平面**。回想一下，我们可以填充一个凸函数得到上图(epigraph)，它是一个凸集。因此，我们也可以用支撑超平面来描述凸函数。此外，观察到支撑超平面刚好与凸函数相切，实际上是函数在该点的切线。回想一下，函数 $f(x)$ 在给定 x_0 处的切线是该函数在该点的梯度的计算值 $d f [x_0]$ 。总结来说，因为凸集可以通过其支撑超平面等价地描述，凸函数可以通过其梯度的函数等价地描述。*Legendre*变换将这个概念形式化。

d

$[x] = [x]_0$

我们从最一般的定义开始，遗憾的是它有一个反直觉的形式，然后看特殊情况将定义与前段描述的直觉联系起来。*Legendre-Fenchel*变换是从凸可微函数 $f(x)$ 到依赖于切线 $s(x) = \nabla f(x)$ 的函数的变换（在Fourier变换的意义上）。值得强调的是，这是函数 $f(\cdot)$ 的变换，而不是变量 x 或在 x 处求值的函数的变换。

x

.

Legendre-Fenchel变换也被称为**凸共轭**（原因我们很快会看到），并且与对偶密切相关([Hiriart-Urruty和Lemar´echal, 2001,]第5章)。

[D] **定义7.4.** 函数 $f: \mathbb{R} \rightarrow \mathbb{R}$ 的**凸共轭**是一个函数 $[*]f$ ，定义为

$$f^* = \sup \langle s, x \rangle - f(x). \quad (7.53)$$

$x \in \mathbb{R}$

注意前面的凸共轭定义不需要函数 f 是凸的或可微的。在定义[7.4]中，我们使用了一般内积（第[3.2]节），但在本节的其余部分我们

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] <https://mml-book.com>[.] 7.3 Convex Optimization

将考虑有限维向量之间的标准点积($\langle s, x \rangle = sx$)以避免过多的技术细节。

为了以几何方式理解定义[7.4]，考虑一个良好的

简单的一维凸可微函数，例如最容易理解的 $f(x) = x^2$ 。注意由于我们考虑的是一维问题，超平面简化为一条直线。考虑直线 $y = sx + c$ 。回想一下，我们能够通过支撑超平面来描述凸函数，所以让我们尝试通过支撑直线来描述这个函数 $f(x)$ 。固定直线的梯度 $s \in \mathbb{R}$ ，对于 f 图像上的每个点 $(x_0, f(x_0))$ ，找到使直线仍然与 $(x_0, f(x_0))$ 相交的 c 的最小值。注意 c 的最小值是斜率为 s 的直线“刚好接触”函数 $f(x) = x^2$ 的位置。通过 $(x_0, f(x_0))$ 且梯度为 s 的直线由下式给出：

$$y - f(x_0) = s(x - x_0). \quad (7.54)$$

这条直线的 y 截距是 $-sx_0 + f(x_0)$ 。因此，使 $y = sx + c$ 与 f 的图像相交的 c 的最小值为

$$\inf_{x_0} \{-sx_0 + f(x_0)\}. \quad (7.55)$$

按照惯例，上述凸共轭定义为这个值的负数。本段的推理并不依赖于我们选择了一维凸可微函数这个事实，对于 $f: \mathbb{R}^D \rightarrow \mathbb{R}$ 的非凸和不可微函数同样成立。

备注. 凸可微函数如例子 $f(x) = x^2$ 是一个很好的特殊情况，这里不需要上确界，并且函数与其 Legendre 变换之间存在一一对应关系。让我们从第一性原理推导这一点。对于凸可微函数，我们知道在 x_0 处切线接触 $f(x_0)$ ，使得

$$f(x_0) = sx_0 + c. \quad (7.56)$$

回想我们想要用梯度 $\nabla_x f(x)$ 来描述凸函数 $f(x)$ ，且 $s = \nabla_x f(x_0)$ 。我们重新整理得到 $-c$ 的表达式：

$$-c = sx_0 - f(x_0). \quad (7.57)$$

注意 $-c$ 随 x_0 变化，因此也随 s 变化，这就是为什么我们可以将其视为 s 的函数，我们称之为

$$f^*(s) := sx_0 - f(x_0). \quad (7.58)$$

比较 (7.58) 与定义 7.4，我们看到 (7.58) 是一个特殊情况（没有上确界）。◇

共轭函数具有良好的性质；例如，对于凸函数，再次应用 Legendre 变换会让我们回到原函数。就像 $f(x)$ 的斜率是 s 一样， $f^*(s)$ 的斜率是 x 。以下两个例子展示了凸共轭在机器学习中的常见用途。

Example 7.7（凸共轭）

为了说明凸共轭的应用，考虑二次函数

$$f_\lambda(y) = (1/2)y^T K^{-1} y \quad (7.59)$$

基于正定矩阵 $K \in \mathbb{R}^{n \times n}$ 。我们将原变量记为 $y \in \mathbb{R}^n$ ，对偶变量记为 $\alpha \in \mathbb{R}^n$ 。

应用定义 7.4，我们得到函数

$$f^*_\lambda(\alpha) = \sup_{y \in \mathbb{R}^n} \langle y, \alpha \rangle - (1/2)y^T K^{-1} y. \quad (7.60)$$

由于函数是可微的，我们可以通过求导并将其设为零来找到最大值。

$$\partial/\partial y [\langle y, \alpha \rangle - (1/2)y^T K^{-1} y] = \alpha - \lambda K^{-1} y \quad (7.61)$$

因此，当梯度为零时，我们有 $y = (1/\lambda)K\alpha$ 。代入 (7.60) 得到

$$f^*_\lambda(\alpha) = \alpha^T K \alpha / \lambda - (1/2\lambda^2)\alpha^T K K^{-1} K \alpha = (1/2\lambda)\alpha^T K \alpha. \quad (7.62)$$

Example 7.8

在机器学习中，我们经常使用函数的和；例如，训练集的目标函数包括训练集中每个样本的损失之和。在下文中，我们推导损失和 $L(t) = \sum_{i=1}^n \ell_i(t_i)$ 的凸共轭，其中 $\ell: \mathbb{R} \rightarrow \mathbb{R}$ 。这也说明了凸共轭在向量情况下的应用。那么，

$$L^*(z) = \sup_{\mathbf{t} \in \mathbb{R}^n} \langle \mathbf{z}, \mathbf{t} \rangle - \sum_{i=1}^n \ell_i(t_i) \quad (7.63a)$$

$$= \sup_{\mathbf{t} \in \mathbb{R}^n} \sum_{i=1}^n z_i t_i - \ell_i(t_i) \quad \text{点积的定义} \quad (7.63b)$$

$$= \sum_{i=1}^n \sup_{t_i \in \mathbb{R}} \{z_i t_i - \ell_i(t_i)\} \quad (7.63c)$$

$$= \sum_{i=1}^n \ell_i^*(z_i). \quad \text{共轭的定义} \quad (7.63d)$$

回想在第 7.2 节中，我们使用 Lagrange 乘数推导了对偶优化问题。此外，对于凸优化问题，我们有强对偶性，即原问题和对偶问题的解匹配。这里描述的 Legendre-Fenchel 变换也可以用来推导对偶优化问题。此外，当函数是凸且可微时，上确界是唯一的。为了进一步研究这两种方法之间的关系，让我们考虑一个线性等式约束凸优化问题。

Example 7.9

设 $f(y)$ 和 $g(x)$ 是凸函数， A 是适当维数的实矩阵，使得 $Ax = y$ 。那么

$$\min_x f(Ax) + g(x) = \min_{Ax=y} f(y) + g(x). \quad (7.64)$$

通过为约束 $Ax = y$ 引入 Lagrange 乘数 u ,

$$\min_{Ax=y} f(y) + g(x) = \min_{x,y} \max_u f(y) + g(x) + (Ax - y)^T u \quad (7.65a)$$

$$= \max_u \min_{x,y} f(y) + g(x) + (Ax - y)^T u, \quad (7.65b)$$

其中交换 $\max$ 和 $\min$ 的最后一步是由于 $f(y)$ 和 $g(x)$ 是凸函数。通过分解点积项并收集 x 和 y ,

$$\max_u \min_{x,y} f(y) + g(x) + (Ax - y)^T u \quad (7.66a)$$

$$= \max_u [\min_y \{-y^T u + f(y)\} + \min_x \{(Ax)^T u + g(x)\}] \quad (7.66b)$$

$$= \max_u [\min_y \{-y^T u + f(y)\} + \min_x \{x^T A^T u + g(x)\}] \quad (7.66c)$$

回忆凸共轭的定义（定义 7.4）以及内积的对称性，

$$\max_u u^T y + f^*(y) + \min_x x^T A^T u + g(x) \quad (7.67a) \quad u, y, x$$

$$= \max_u -f(u) - g(-A^T u). \quad (7.67b) \quad u$$

因此，我们已经证明了

$$\min f(Ax) + g(x) = \max_u -f(u) - g(-A^T u). \quad (7.68) \quad x, u$$

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由 Cambridge University Press 出版 (2020).

246 连续优化

Legendre-Fenchel 共轭对于可以表达为凸优化问题的机器学习问题非常有用。特别是对于独立应用于每个样本的凸损失函数，共轭损失是推导对偶问题的便捷方法。

7.4 延伸阅读

连续优化是一个活跃的研究领域，我们并不试图提供最新进展的全面描述。

从梯度下降的角度来看，有两个主要缺陷，每个都有自己的一套文献。第一个挑战是梯度下降是一阶算法，不使用关于曲面曲率的信息。当存在长谷时，梯度垂直于感兴趣的方向。动量的想法可以推广到一般的加速方法类别(Nesterov, 2018)。共轭梯度方法通过考虑之前的方向来避免梯度下降面临的问题(Shewchuk, 1994)。二阶方法如Newton方法使用Hessian来提供关于曲率的信息。许多选择步长和动量等想法的选择都是通过考虑目标函数的曲率而产生的(Goh, 2017; Bottou et al., 2018)。准Newton方法如L-BFGS尝试使用更便宜的计算方法来近似Hessian(Nocedal and Wright, 2006)。最近人们对计算下降方向的其他度量产生了兴趣，导致了诸如镜像下降(Beck and Teboulle, 2003)和自然梯度(Toussaint, 2012)等方法。

第二个挑战是处理不可微函数。当函数中存在折痕时，梯度方法没有很好的定义。在这些情况下，可以使用子梯度方法(Shor, 1985)。有关优化不可微函数的更多信息和算法，我们参考Bertsekas (1999)的书。关于数值求解连续优化问题的不同方法有大量文献，包括约束优化问题的算法。欣赏这些文献的良好起点是Luenberger (1969)和Bonnans et al. (2006)的书。Bubeck (2015)提供了连续优化的最新综述。机器学习的现代应用通常意味着数据集的大小禁止使用批量梯度下降，因此随机梯度下降是大规模机器学习方法的当前主力。最近的文献综述包括Hazan (2015)和Bottou et al. (2018)。

对于对偶性和凸优化，Boyd and Vandenberghe (2004)的书包括在线讲座和幻灯片。Bertsekas (2009)提供了更数学化的处理，优化领域关键研究者之一的最新著作是Nesterov (2018)。凸优化基于凸分析，对凸函数基础结果感兴趣的读者可参考Rockafellar (1970)、Hiriart-Urruty and Lemaréchal (2001)和Borwein and Lewis (2006)。Legendre-Fenchel变换也在上述凸分析书籍中涵盖，但Zia et al. (2009)提供了更适合初学者的介绍。Legendre-Fenchel变换在凸优化算法分析中的作用在Polyak (2016)中进行了综述。

草稿 (2023-10-18) “机器学习数学”。反馈: <https://mml-book.com>。练习 247

练习

7.1 考虑单变量函数

$$f(x) = x^3 + 6x^2 - 3x - 5.$$

找出其驻点并指出它们是最大值、最小值还是鞍点。

7.2 考虑随机梯度下降的更新方程（方程(7.15)）。

写出当我们使用大小为1的mini-batch时的更新。

7.3 考虑以下陈述是否为真或假：

- a. 任意两个凸集的交集是凸的。
- b. 任意两个凸集的并集是凸的。
- c. 一个凸集A与另一个凸集B的差是凸的。

7.4 考虑以下陈述是否为真或假：

- a. 任意两个凸函数的和是凸的。
- b. 任意两个凸函数的差是凸的。
- c. 任意两个凸函数的积是凸的。
- d. 任意两个凸函数的最大值是凸的。

7.5 将以下优化问题表达为矩阵记号的标准线性规划

$$\max p^T x + \xi \quad x \in \mathbb{R}^2, \xi \in \mathbb{R}$$

受约束条件 $\xi \geq 0$, $x_0 \leq 0$ 和 $x_1 \leq 3$ 。

7.6 考虑图7.9中所示的线性规划，

$$\min -5/3 x_1 \quad x \in \mathbb{R}^2$$

受约束条件 $[-2 \ 1] x \leq [8] \quad [2 \ -4] x \leq [5]$

使用Lagrange对偶性推导对偶线性规划。

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 剑桥大学出版社出版 (2020)。

连续优化

7.7 考虑图7.4中所示的二次规划问题，

$$\min_{\mathbf{x} \in \mathbb{R}^2} \frac{1}{2} \begin{bmatrix} x_1 & x_2 \end{bmatrix}^{\top} \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 5 \\ 3 \end{bmatrix}^{\top} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\text{subject to } \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq 1$$

使用Lagrange对偶性推导对偶二次规划。

7.8 考虑以下凸优化问题

$$\min_{\mathbf{w} \in \mathbb{R}^D} \frac{1}{2} \mathbf{w}^{\top} \mathbf{w}$$

$$\text{subject to } \mathbf{w}^{\top} \mathbf{x} \geq 1$$

通过引入Lagrange乘数 λ 推导Lagrangian对偶。

7.9 考虑 $\mathbf{x} \in \mathbb{R}^D$ 的负熵，

$$f(\mathbf{x}) = \sum_{d=1}^D x_d \log x_d$$

通过假设标准点积，推导凸共轭函数 $f^*(s)$ 。

提示：对适当的函数求梯度并令梯度为零。

7.10 考虑函数

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^{\top} \mathbf{A} \mathbf{x} + \mathbf{b}^{\top} \mathbf{x} + c$$

其中 $\mathbf{A}$ 是严格正定的，这意味着它是可逆的。推导 $f(\mathbf{x})$ 的凸共轭。提示：对适当的函数求梯度并令梯度为零。

7.11 hinge损失（支持向量机使用的损失）由下式给出

$$L(\alpha) = \max\{0, 1 - \alpha\}$$

如果我们有兴趣应用梯度方法如L-BFGS，并且不想诉诸于次梯度方法，我们需要平滑hinge损失中的拐点。计算hinge损失的凸共轭 $L^*(\beta)$ ，其中 β 是对偶变量。添加一个 ℓ_2 近似项，并计算结果函数的共轭

$$L^*(\beta) + \frac{\gamma}{2} \beta^2$$

其中 γ 是给定的超参数。

草稿 (2023-10-18) “机器学习数学”。反馈：<https://mml-book.com>

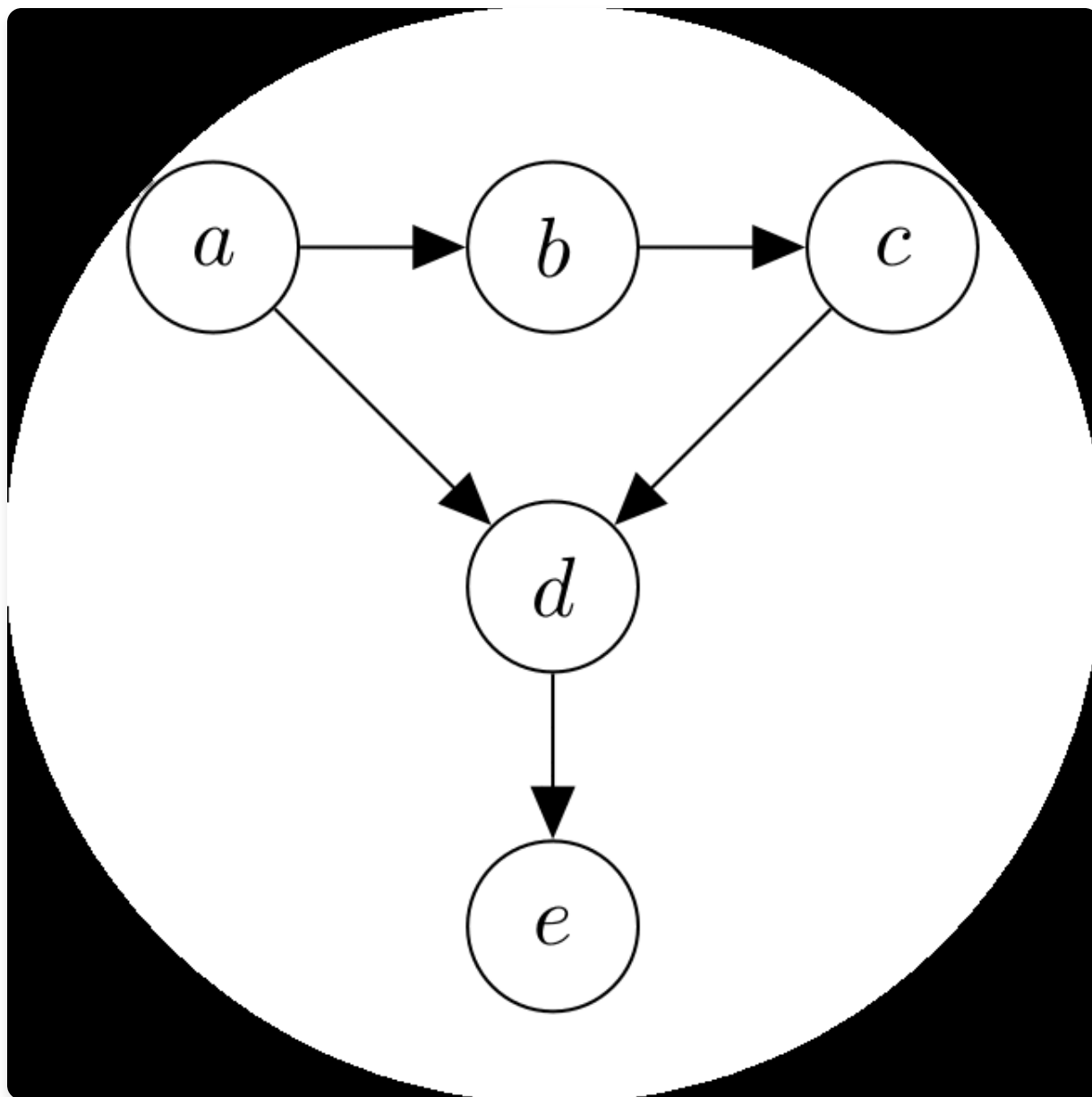
第二部分

机器学习核心问题

这些材料由剑桥大学出版社作为《机器学习数学》出版，作者Marc Peter Deisenroth, A. Aldo Faisal, 和 Cheng Soon Ong (2020)。此版本免费供个人查看和下载。不得重新分发、转售或用于衍生作品。 ©by M. P. Deisenroth, A. A. Faisal, and C. S. Ong, 2023. <https://mml-book.com>

第8章

当模型遇到数据



在本书的第一部分中，我们介绍了构成许多机器学习方法基础的数学。希望读者能够从第一部分学习数学语言的基本形式，我们现在将使用这些来描述和讨论机器学习。本书的第二部分介绍了机器学习的四大支柱：

回归（第9章）

降维（第10章）

密度估计（第11章）

分类（第12章）

本书这一部分的主要目标是说明第一部分介绍的数学概念如何用于设计机器学习算法，这些算法可以用来解决四大支柱范围内的任务。我们打算介绍高级机器学习概念，而是提供一套实用方法，让读者能够应用从第一部分获得的知识。它还为已经熟悉数学的读者提供了通往更广泛机器学习文献的门户。

8.1 数据、模型和学习

此时值得停下来考虑机器学习算法要解决的问题。如第1章所讨论的，机器学习系统有三个主要组成部分：数据、模型和学习。机器学习的主要问题是”我们所说的好模型是什么意思？“。模型这个词有很多微妙之处，我们将在本章中多次重新审视它。如何客观地定义”好”这个词也不是完全显而易见的。机器学习的指导原则之一是好的模型应该在未见过的数据上表现良好。这要求我们定义一些性能指标，如准确性或与真实值的距离，以及找出在这些性能指标下表现良好的方法。本章涵盖了一些必要的数学和统计语言片段，这些通常用于

表8.1 来自虚构人力资源数据库的示例数据，不是数值格式

姓名	性别	学位	邮编	年龄	年薪
Aditya	M	MSc	W21BG	36	89563
Bob	M	PhD	EC1A1BA	47	123543
Chloé	F	BEcon	SW1A1BH	26	23989
Daisuke	M	BSc	SE207AT	68	138769
Elisabeth	F	MBA	SE10AA	33	113888

用于讨论机器学习模型。通过这样做，我们简要概述了训练模型的当前最佳实践，使得结果预测器在我们尚未见过的数据上表现良好。

当模型遇到数据

这些材料由剑桥大学出版社作为《机器学习数学》出版，作者Marc Peter Deisenroth, A. Aldo Faisal, 和 Cheng Soon Ong (2020)。此版本免费供个人查看和下载。不得重新分发、转售或用于衍生作品。 ©by M. P. Deisenroth, A. A. Faisal, and C. S. Ong, 2023. <https://mml-book.com>

正如第1章所述，我们使用“机器学习算法”这个短语有两种不同的含义：训练和预测。我们将在本章中描述这些概念，以及在不同模型中进行选择的思想。我们将在第[8.2]节介绍经验风险最小化框架，在第[8.3]节介绍最大似然原理，在第[8.4]节介绍概率模型的概念。我们在第[8.5]节简要概述了用于指定概率模型的图形语言，最后在第[8.6]节讨论模型选择。本节的其余部分将详细阐述机器学习的三个主要组成部分：数据、模型和学习。

8.1.1 数据作为向量

我们假设我们的数据可以被计算机读取，并且能够在数值格式中得到充分表示。数据被假设为表格形式（图[8.1]），其中我们将表格的每一行视为表示特定实例或样本，每一列为特定特征。近年来，机器学习已被应用于许多类型的数据，这些数据并不明显以表格数值格式出现，例如基因组序列、文本和网页的图像内容，以及社交媒体图谱。

我们不讨论识别良好特征的重要且具有挑战性的方面。这些方面中的许多取决于领域专业知识并需要仔细的工程设计，近年来，它们被归入数据科学的范畴[Stray, 2016; Adhikari and DeNero, 2018]。

即使当我们有表格格式的数据时，仍然需要做出选择来获得数值表示。例如，在表[8.1]中，性别列（分类变量）可以转换为数字0表示“男性”，1表示“女性”。或者，性别可以分别用数字-1、+1表示（如表[8.2]所示）。此外，在构建表示时使用领域知识通常很重要，例如知道大学学位从学士到硕士再到博士的递进关系，或者意识到提供的邮政编码不仅仅是字符串，而是实际编码了伦敦的一个区域。在表[8.2]中，我们将表[8.1]的数据转换为数值格式，每个邮政编码表示为两个数字，

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] [<https://mml-book.com>] [8.1 Data, Models, and Learning] [253]

[Gender ID] [Degree] [Latitude] [Longitude] [Age] [Annual Salary] [表8.2] [来自虚构人力] [(纬度)] [(经度)] [(以千为单位)] [资源数据库的] [-1] [2] [51.5073] [0.1290] [36] [89.563] [示例数据] [-1] [3] [51.5074] [0.1275] [47] [123.543] [（见表[8.1]），] [+1] [1] [51.5071] [0.1278] [26] [23.989] [转换为数值格式。] [-1] [1] [51.5075] [0.1281] [68] [138.769] [+1] [2] [51.5074] [0.1278] [33] [113.888]

纬度和经度。即使是可能可以直接读入机器学习算法的数值数据，也应该仔细考虑单位、缩放和约束。在没有额外信息的情况下，应该对数据集的所有列进行平移和缩放，使它们的经验均值为0，经验方差为1。为了本书的目的，我们假设领域专家已经适当地转换了数据，即每个输入 $x[n]$ 是一个D维实数向量，这些被称为特征、属性或协变量。我们认为数据集的形式如表[8.2]所示。注意我们在新的数值表示中删除了表[8.1]的姓名列。这样做有两个主要原因：(1)我们不期望标识符（姓名）对机器学习任务具有信息性；(2)我们可能希望匿名化数据以帮助保护员工的隐私。

在本书的这一部分中，我们将使用N来表示数据集中样本的数量，并用小写 $n = 1, \dots, N$ 来索引样本。我们假设给定一组数值数据，表示为向量数组（表[8.2]）。每一行是一个特定的个体 $x[n]$ ，在机器学习中通常被称为样本或数据点。下标n指的是这是总共N个样本中的第n个样本。每一列表示关于样本的特定关注特征，我们将特征索引为 $d = 1, \dots$,

D。回想一下数据表示为向量，这意味着每个样本（每个数据点）是一个D维向量。表格的方向来源于数据库社区，但对于一些机器学习算法（例如，在第[10]章中），将样本表示为列向量更方便。

让我们考虑基于表[8.2]中的数据从年龄预测年薪的问题。这被称为监督学习问题，其中我们有一个与每个样本 $x[n]$ （年龄）相关联的标签 $y[n]$ （薪水）。标签 $y[n]$ 有各种其他名称，包括目标、响应变量和注释。数据集写作一组样本-标签对 $\{(x_1, y_1), \dots, (x[n], y[n]), \dots, (x[N], y[N])\}$ 。样本表 $\{x_1, \dots, x[N]\}$ 经常被连接，写作 $X \in \mathbb{R}[N \times D]$ 。图[8.1]展示了由表[8.2]最右两列组成的数据集，其中 x = 年龄， y = 薪水。

我们使用本书第一部分介绍的概念来形式化

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[254] [*When Models Meet Data*]

[图8.1] [玩具数据]

线性回归的训练数据以 (x_n, y_n) 配对形式来自表8.2最右边两列。我们对一个60岁的人($x = 60$)的工资 y 感兴趣，如垂直红色虚线所示，这不是训练数据的一部分。

x

像前一段落中的机器学习问题。

将数据表示为向量 x_n 允许我们使用线性代数的概念(在第2章介绍)。在许多机器学习算法中，我们还需要能够比较两个向量。正如我们将在第9章和第12章中看到的，计算两个示例之间的相似性或距离使我们能够形式化这样的直觉：具有相似特征的示例应该具有相似的标签。两个向量的比较需要我们构建一个几何(在第3章解释)，并允许我们使用第7章的技术优化所得的学习问题。

由于我们有数据的向量表示，我们可以操作数据来找到它的潜在更好表示。我们将讨论以两种方式寻找好的表示：找到原始特征向量的低维近似，以及使用原始特征向量的非线性高维组合。在第10章中，我们将看到通过找到主成分来寻找原始数据空间的低维近似的例子。寻找主成分与第4章介绍的特征值和奇异值分解概念密切相关。对于高维表示，我们将看到一个明确的特征映射 $\varphi(\cdot)$ ，它允许我们使用高维表示 $\varphi(x_n)$ 来表示输入 x_n 。高维表示的主要动机是我们可以构造新特征作为原始特征的非线性组合，这进而可能使学习问题变得更容易。我们将在第9.2节讨论特征映射，并在第12.4节展示这个特征映射如何导致核。近年来，深度学习方法(Goodfellow et al., 2016)在使用数据本身学习新的好特征方面显示出前景，并在计算机视觉、语音识别和自然语言处理等领域非常成功。我们不会在本书的这一部分涵盖神经网络，但读者可以参考

图8.2 示例函数(黑色实对角线)及其在 $x = 60$ 处的预测，即 $f(60) = 100$ 。

x

第5.6节了解反向传播的数学描述，这是训练神经网络的关键概念。

8.1.2 作为函数的模型

一旦我们有了合适的向量表示的数据，我们就可以着手构建预测函数(称为预测器)。

在第1章中，我们还没有语言来精确描述模型。使用本书第一部分的概念，我们现在可以介绍“模型”的含义。我们在本书中介绍两种主要方法：作为函数的预测器，和作为概率模型的预测器。我们在这里描述前者，在下一小节描述后者。

预测器是一个函数，当给定特定输入示例(在我们的情况下，是特征向量)时，产生输出。现在，考虑输出为单个数字，即实值标量输出。这可以写为

$$f: \mathbb{R}^D \rightarrow \mathbb{R}, \quad (8.1)$$

其中输入向量 x 是 D 维的(有 D 个特征)，函数 f 然后应用于它(写作 $f(x)$)返回一个实数。图8.2说明了一个可用于计算输入值 x 的预测值的可能函数。

在本书中，我们不考虑所有函数的一般情况，这将涉及需要泛函分析。相反，我们考虑线性函数的特殊情况

$$f(x) = \theta^T x + \theta_0 \quad (8.2)$$

对于未知的 θ 和 θ_0 。这种限制意味着第2章和第3章的内容足以精确陈述非概率(与下面描述的相对)机器学习观点的预测器概念。

图8.3 示例函数(黑色实对角线)及其在 $x = 60$ 处的预测不确定性(绘制为高斯分布)。

x

线性函数在可以解决的问题的通用性和所需背景数学的数量之间取得了良好的平衡。

8.1.3 作为概率分布的模型

我们经常认为数据是某些真实潜在效应的噪声观测，并希望通过应用机器学习我们能够从噪声中识别信号。这要求我们有一种量化噪声影响的语言。我们也经常希望有表达某种不确定性的预测器，例如，量化我们对特定测试数据点预测值的信心。正如我们在第6章中看到的，概率论提供了量化不确定性的语言。图8.3说明了预测的不确定性

function作为Gaussian分布。

我们不是将预测器视为单一函数，而是可以将预测器视为概率模型，即描述可能函数分布的模型。在本书中，我们将自己限制在具有有限维参数的分布这一特殊情况，这使我们能够在不需要随机过程和随机测度的情况下描述概率模型。对于这种特殊情况，我们可以将概率模型看作多元概率分布，这已经允许丰富的模型类别。

我们将介绍如何使用概率概念（第6章）在第[8.4]节中定义机器学习模型，并在第[8.5]节中介绍一种图形语言，以紧凑的方式描述概率模型。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: [\[https://mml-book.com\]](https://mml-book.com)] [8.1 Data, Models, and Learning] [257]

8.1.4 学习就是寻找参数

学习的目标是找到一个模型及其相应参数，使得产生的预测器在未见过的数据上表现良好。在讨论机器学习算法时，概念上有三个不同的算法阶段：

1. 预测或推理
2. 训练或参数估计
3. 超参数调优或模型选择

预测阶段是当我们在之前未见过的测试数据上使用训练好的预测器时。换句话说，参数和模型选择已经固定，预测器被应用于表示新输入数据点的新向量。如第1章和前一小节所述，我们将在本书中考虑机器学习的两个学派，对应于预测器是函数还是概率模型。当我们有概率模型时（在第[8.4]节中进一步讨论），预测阶段称为推理。

注释。不幸的是，对于不同的算法阶段没有统一的命名约定。“推理”一词有时也用于指概率模型的参数估计，较少数情况下也可能用于指非概率模型的预测。◇

训练或参数估计阶段是我们基于训练数据调整预测模型的时候。我们希望根据训练数据找到好的预测器，有两种主要策略：基于某种质量度量找到最佳预测器（有时称为找到点估计），或使用Bayesian推理。找到点估计可以应用于两种类型的预测器，但Bayesian推理需要概率模型。

对于非概率模型，我们遵循经验风险最小化原理，我们在第[8.2]节中描述。经验风险最小化直接提供了一个优化问题来找到好的参数。对于统计模型，使用maximum likelihood原理来找到一组好的参数（第[8.3]节）。我们还可以使用概率模型对参数的不确定性建模，我们将在第[8.4]节中更详细地研究这一点。

我们使用数值方法来找到“拟合”数据的好参数，大多数训练方法可以被认为是爬山方法，用于找到目标的最大值，例如likelihood的最大值。为了应用爬山方法，我们使用第[5]章中描述的梯度，并实现第7章中的数值优化方法。

如第1章所述，我们有兴趣基于数据学习一个模型，使其在未来数据上表现良好。仅仅让模型在训练数据上拟合良好是不够的，预测器需要在未见过的数据上表现良好。我们使用交叉验证（第[8.2.4]节）来模拟我们的预测器在未来未见数据上的行为。正如我们将在本章中看到的，为了实现在未见数据上表现良好的目标，我们需要在训练数据上拟合良好和找到现象的“简单”解释之间取得平衡。这种权衡通过正则化（第[8.2.3]节）或添加先验（第[8.3.2]节）来实现。在哲学中，这被认为既不是归纳也不是演绎，而是称为溯因。根据斯坦福哲学百科全书，溯因是推理到最佳解释的过程(Douven, 2017)。

我们经常需要对预测器的结构做出高层次的建模决策，比如要使用的组件数量或要考虑的概率分布类别。组件数量的选择是超参数的一个例子，这个选择可以显著影响模型的性能。在不同模型中进行选择的问题称为模型选择，我们在第[8.6]节中描述。对于非概率模型，模型选择通常使用嵌套交叉验证来完成，这在第[8.6.1]节中描述。我们也使用模型选择来选择我们模型的超参数。

注释。参数和超参数之间的区别在某种程度上是任意的，主要是由可以数值优化的东西与需要使用搜索技术的东西之间的区别驱动的。考虑这种区别的另一种方式是将参数视为

[258] [*When Models Meet Data*]

参数概率模型的显式参数，并将超参数(hyperparameters)（更高级别的参数）视为控制这些显式参数分布的参数。[◇]

在接下来的章节中，我们将研究机器学习的三种类型：经验风险最小化（第8.2节），最大似然原理（第8.3节），以及概率建模（第8.4节）。

8.2 经验风险最小化

在掌握了所有数学知识之后，我们现在可以介绍学习的含义了。机器学习的”学习”部分归结为基于训练数据估计参数。

在本节中，我们考虑预测器是一个函数的情况，并在第8.3节中考虑概率模型的情况。我们描述经验风险最小化的思想，这最初因支持向量机(support vector machine)的提出而广受欢迎（在第12章中描述）。然而，其一般原理广泛适用，并允许我们在不显式构建概率模型的情况下询问什么是学习的问题。有四个主要的设计选择，我们将在以下小节中详细介绍：

8.2.1 我们允许预测器采用的函数集合是什么？

8.2.2 我们如何衡量预测器在训练数据上的表现？

8.2.3 我们如何仅从训练数据构建在未见测试数据上表现良好的预测器？

8.2.4 在模型空间上搜索的程序是什么？

8.2.1 函数的假设类

假设我们得到D维的N个样本 $x_n \in \mathbb{R}^D$ 和相应的标量标签 $y_n \in \mathbb{R}$ 。我们考虑监督学习设置，其中我们获得对 $(x_1, y_1), \dots, (x_N, y_N)$ 。给定这些数据，我们希望估计一个由 θ 参数化的预测器 $f(\cdot, \theta): \mathbb{R}^D \rightarrow \mathbb{R}$ 。我们希望能够找到一个好的参数 θ^* ，使得我们很好地拟合数据，即

$$f(x_n, \theta^*) \approx y_n \text{ 对所有 } n = 1, \dots, N. \quad (8.3)$$

在本节中，我们使用记号 $\hat{y}_n = f(x_n, \theta^*)$ 来表示预测器的输出。

备注。为了便于表述，我们将用监督学习（其中我们有标签）来描述经验风险最小化。这简化了假设类和损失函数(loss function)的定义。在机器学习中，选择参数化的函数类也很常见，例如仿射函数。◇

示例 8.1

我们介绍普通最小二乘回归问题来说明经验风险最小化。回归的更全面描述见第9章。当标签 y_n 是实值时，预测器函数类的流行选择是仿射函数集合。我们通过将附加的单位特征连接到 $x_n = [1, x_n^{(1)}, x_n^{(2)}, \dots, x_n^{(D)}]^T$ 来为仿射函数选择更紧凑的记号。参数向量相应地为 $\theta = [\theta_0, \theta_1, \theta_2, \dots, \theta_D]^T$ ，允许我们将预测器写作线性函数

$$f(x_n, \theta) = \theta^T x_n. \quad (8.4)$$

这个线性预测器等价于仿射模型

$$f(\mathbf{x}_n, \theta) = \theta_0 + \sum_{d=1}^D \theta_d x_n^{(d)}. \quad (8.5)$$

预测器将表示单个样本 $\mathbf{x}_n$ 的特征向量作为输入并产生实值输出，即 $f: \mathbb{R}^{D+1} \rightarrow \mathbb{R}$ 。本章前面的图形有一条直线作为预测器，这意味着我们假设了一个仿射函数。

除了线性函数，我们可能希望考虑非线性函数作为预测器。神经网络的最新进展允许高效计算更复杂的非线性函数类。

给定函数类，我们想要搜索一个好的预测器。我们现在转向经验风险最小化的第二个要素：如何衡量预测器对训练数据的拟合程度。

8.2.2 训练的损失函数

考虑特定样本的标签 y_n ；以及我们基于 $\mathbf{x}_n$ 做出的相应预测 $\hat{y}_n$ 。为了定义很好地拟合数据意味着什么，我们需要指定一个损失函数 $\ell(y_n, \hat{y}_n)$ ，它将真实标签和预测作为输入，并产生一个非负数（称为损失），表示我们在这个特定预测上犯了多少错误。我们寻找好参数向量 θ^* 的目标是最小化 N 个训练样本集合上的平均损失。

在机器学习中通常做的一个假设是样本集合 $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ 是独立同分布的。独立(independent)一词（第6.4.5节）意味着两个数据点 $(\mathbf{x}_i, y_i)$ 和 $(\mathbf{x}_j, y_j)$ 在统计上不相互依赖，这意味着经验均值是总体均值的良好估计（第6.4.1节）。这意味着我们可以使用训练数据上损失的经验均值。对于给定的训练集 $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ ，我们引入样本矩阵 $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times D}$ 和标签向量 $\mathbf{y} := [y_1, \dots, y_N]^T \in \mathbb{R}^N$ 的记号。使用这个矩阵

notation表示平均损失为

$$\frac{1}{N} \sum_{n=1}^N \ell(y_n, \hat{y}_n) = \ell(\mathbf{y}, \hat{\mathbf{y}}), \quad (8.6) \quad N \rightarrow \infty$$

其中 $\hat{y}_n = f(\mathbf{x}_n, \theta)$ 。方程(8.6)被称为*经验风险*，它依赖于三个参数：预测器 f 和数据 $\mathbf{X}, \mathbf{y}$ 。这种通用的学习策略被称为*经验风险最小化*。

例8.2（最小二乘损失）

继续最小二乘回归的例子，我们指定在训练过程中使用平方损失来衡量出错的代价 $\ell(y, \hat{y}) = (y - \hat{y})^2$ 。我们希望最小化经验风险(8.6)。

这是数据上损失的平均值

$$\frac{1}{N} \sum_{n=1}^N (y_n - f(\mathbf{x}_n, \theta))^2, \quad (8.7) \quad \theta \in \mathbb{R}^{D+1}$$

其中我们替换了预测器 $\hat{y}_n = f(\mathbf{x}_n, \theta)$ 。通过使用我们选择的线性预测器 $f(\mathbf{x}, \theta) = \theta^T \mathbf{x}$ ，我们得到优化问题

$$\min_{\theta} \frac{1}{N} \sum_{n=1}^N (y_n - \theta^T \mathbf{x}_n)^2. \quad (8.8) \quad \theta \in \mathbb{R}^{D+1}$$

这个方程可以等价地表示为矩阵形式

$$\min_{\theta} \|\mathbf{y} - \mathbf{X}\theta\|^2. \quad (8.9) \quad \theta \in \mathbb{R}^{D+1}$$

这被称为*最小二乘问题*。通过求解正规方程，存在一个闭式解析解，我们将在第9.2节讨论。

我们对仅在训练数据上表现良好的预测器不感兴趣。相反，我们寻求在未见测试数据上表现良好（风险较低）的预测器。更正式地说，我们感兴趣找到一个预测器 f （参数固定），使期望风险最小化

$$R[\text{true}](f) = E[x][y][\ell(y, f(x))], \quad (8.10)$$

其中 y 是标签， $f(x)$ 是基于样本 x 的预测。符号 R_{true} 表示如果我们有无量的数据时的真实风险。期望是在所有可能数据和标签的（无限）集合上。从我们最小化期望风险的愿望中产生了两个实际问题，我们在以下两个小节中解决：

我们应该如何改变训练程序以更好地泛化？我们如何从（有限）数据中估计期望风险？

备注。许多机器学习任务都指定了相关的性能度量，例如预测准确率或均方根误差。性能度量可能更复杂、对成本敏感，并捕获特定应用的细节。原则上，经验风险最小化的损失函数设计应该直接对应于机器学习任务指定的性能度量。在实践中，损失函数的设计和性能度量之间经常存在不匹配。这可能是由于实现的便利性或优化的效率等问题。◇

8.2.3 正则化以减少过拟合

本节描述了对经验风险最小化的补充，使其能够很好地泛化（近似最小化期望风险）。回想一下，训练机器学习预测器的目标是在未见数据上表现良好，即预测器泛化良好。我们通过保留整个数据集的一部分来模拟这些未见数据。这个保留集被称为 **测试集**。给定预测器 f 的足够丰富的函数类别，我们基本上可以记住训练数据以获得零经验风险。虽然这对于最小化训练数据上的损失（因此风险）很好，但我们不会期望预测器能很好地泛化到未见数据。在实践中，我们只有有限的数据集，因此我们将数据分为训练集和测试集。训练集用于拟合模型，测试集（在训练期间机器学习算法看不到）用于评估泛化性能。重要的是，用户在观察测试集后不要循环回到新一轮的训练。我们使用下标 `[train]` 和 `[test]` 分别表示训练集和测试集。我们将在第 8.2.4 节重新讨论使用有限数据集评估期望风险的想法。

事实证明，经验风险最小化可能导致 **过拟合**，即预测器过于紧密地拟合训练数据，无法很好地泛化到新数据 [(Mitchell, 1997)]。在训练集上平均损失很小但在测试集上平均损失很大的这种一般现象，往往在我们数据很少且假设类复杂时发生。对于特定的预测器 f （参数固定），当来自训练数据 R_{emp} 的风险估计低估了期望风险 R_{true} 时，就会发生过拟合现象。由于我们通过使用测试集上的经验风险 R_{emp} 来估计期望风险 $R_{\text{true}}(f)$ ，如果测试风险远大于训练风险，这就是过拟合的一个指示。我们在第 8.3.3 节重新讨论过拟合的概念。

因此，我们需要通过引入惩罚项来以某种方式偏向经验风险最小化器的搜索，这使得

优化器返回过度灵活的预测器。在机器学习中，惩罚项被称为 *regularization*。Regularization 是一种在经验风险最小化的准确解和解的大小或复杂性之间进行折衷的方法。

例 8.3 (正则化最小二乘法)

Regularization 是一种阻止优化问题产生复杂或极端解的方法。最简单的正则化策略是

[数学机器学习草稿 (2023-10-18)。反馈:] [https://mml-book.com] [8.2 经验风险最小化] [263]

将最小二乘问题

$$\min \|y - X\theta\|_2^2 \quad (8.11)$$

替换为“正则化”问题，通过添加一个仅涉及 θ 的惩罚项：

$$\min \|y - X\theta\|_2^2 + \lambda \|\theta\|_2^2 \quad (8.12)$$

附加项 $\|\theta\|_2^2$ 称为 *regularizer*，参数 λ 是 *regularization parameter*。Regularization parameter 在最小化训练集损失和参数 θ 的量级之间进行权衡。如果我们遇到过拟合，参数值的量级通常会变得相对较大 [(Bishop, 2006).]

Regularization 项有时被称为 *penalty term*，它使向量 θ 偏向更接近原点。Regularization 的思想也出现在概率模型中作为参数的先验概率。回顾第 6.6 节，为了使后验分布与先验分布具有相同的形式，先验和似然需要是共轭的。我们将在第 8.3.2 节重新讨论这个想法。我们将在第 12 章看到，regularizer 的思想等价于大间隔 (large margin) 的思想。

8.2.4 交叉验证评估泛化性能

我们在前一节中提到，我们通过将预测器应用于测试数据来估计泛化误差。这些数据有时也被称为 *validation set*。Validation set 是我们保留的可用训练数据的子集。这种方法的一个实际问题是数据量有限，理想情况下我们希望使用尽可能多的可用数据来训练模型。这需要我们保持 validation set V 较小，这会导致预测性能的估计产生噪声(高方差)。解决这些矛盾目标(大训练集、大 validation set)的一个解决方案是使用 *cross-validation*。K-fold cross-validation 有效地将数据分成 K 块，其中 $K-1$ 块形成训练集 R ，最后一块作为 validation set V (类似于之前概述的想法)。Cross-validation 遍历(理想情况下)所有将块分配给 R 和 V 的组合；见图 8.4。这个过程对所有 K 个 validation set 选择重复进行，并对 K 次运行的模型性能进行平均。

我们将数据集分成两个集合 $D = R \cup V$ ，使它们不重叠($R \cap V = \emptyset$)，其中 V 是 validation set，我们在 R 上训练模型。训练后，我们评估预测器 f 在

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由剑桥大学出版社出版 (2020).]

[264] [当模型遇到数据]



图 8.4 K-fold cross-validation。数据集被分成 $K = 5$ 块，其中 $K-1$ 块作为训练集(蓝色)，一块作为 validation set(橙色阴影)

图 8.4 K-fold cross-validation。数据集被分成 $K = 5$ 块，其中 $K-1$ 块作为训练集(蓝色)，一块作为 validation set(橙色阴影)

validation set V 上的性能(例如，通过计算训练模型在 validation set 上的均方根误差(RMSE))。更精确地，对于每个分区 k ，训练数据 $R^{(k)}$ 产生一个预测器 $f^{(k)}$ ，然后将其应用于 validation set $V^{(k)}$ 来计算经验风险 $R(f^{(k)}, V^{(k)})$ 。我们循环遍历所有可能的 validation 和 training sets 分区，并计算预测器的平均泛化误差。Cross-validation 近似期望泛化误差

$$\frac{1}{K} \mathbb{E} [R(f^{(k)}, V^{(k)})] \approx \frac{1}{K} \sum_{k=1}^K R(f^{(k)}, V^{(k)}), \quad (8.13)$$

其中 $R(f^{(k)}, V^{(k)})$ 是预测器 $f^{(k)}$ 在 validation set $V^{(k)}$ 上的风险(例如，RMSE)。近似有两个来源：首先，由于有限的训练集，导致不是最好的可能 $f^{(k)}$ ；其次，由于有限的 validation set，导致对风险 $R(f^{(k)}, V^{(k)})$ 的不准确估计。K-fold cross-validation 的一个潜在缺点是训练模型 K 次的计算成本，如果训练成本在计算上昂贵，这可能是负担。在实践中，仅查看直接参数通常是不够的。例如，我们需要探索多个复杂性参数(例如，多个 regularization parameters)，这些可能不是模型的直接参数。根据这些超参数评估模型质量，可能会导致训练运行次数与模型参数数量呈指数关系。可以使用嵌套交叉验证(第 8.6.1 节)来搜索良好的超参数。

然而，cross-validation 是一个 *embarrassingly parallel* 问题，即只需要很少的努力就可以将问题分解为许多并行任务。给定足够的计算资源(例如，云计算、服务器集群)，cross-validation 不需要比单次性能评估更长的时间。

在本节中，我们看到经验风险最小化基于以下概念：函数的假设类、损失函数和

正则化。在第 8.3 节中，我们将看到使用概率分布来替代损失函数和正则化概念的效果。

8.2.5 进一步阅读

由于经验风险最小化的原始发展 (Vapnik, 1998) 使用了大量理论语言，许多后续发展都是理论性的。这个研究领域被称为 *统计学习理论* (Vapnik, 1999; Evgeniou et al., 2000; Hastie et al., 2001; von Luxburg and Schölkopf, 2011)。最近一本基于理论基础并开发高效学习算法的机器学习教科书是 Shalev-Shwartz and Ben-David (2014)。

正则化概念起源于病态逆问题的解法 (Neumaier, 1998)。这里介绍的方法被称为 *Tikhonov 正则化*，还有一个密切相关的约束版本称为 *Ivanov 正则化*。Tikhonov 正则化与偏差-方差权衡和特征选择有着深刻的关系 (Bühlmann and Van De Geer, 2011)。交叉验证的替代方法是 bootstrap 和 jackknife (Efron and Tibshirani, 1993; Davidson and Hinkley, 1997; Hall, 1992)。

将经验风险最小化(第8.2节)视为“无概率”是不正确的。存在一个潜在的未知概率分布 $p(x, y)$ 控制着数据生成。然而，经验风险最小化的方法对分布的选择是不可知的。这与明确要求 $p(x, y)$ 知识的标准统计方法形成对比。此外，由于该分布是样本 x 和标签 y 的联合分布，标签可以是非确定性的。与标准统计学相比，我们不需要为标签 y 指定噪声分布。

8.3 参数估计

在第8.2节中，我们没有明确使用概率分布建模我们的问题。在本节中，我们将看到如何使用概率分布来建模由于观测过程产生的不确定性以及预测器参数的不确定性。在第8.3.1节中，我们介绍似然函数(likelihood)，它类似于经验风险最小化中的损失函数概念(第8.2.2节)。先验概念(第8.3.2节)类似于正则化概念(第8.2.3节)。

8.3.1 最大似然估计

最大似然估计(MLE)背后的思想是定义一个参数函数，使我们能够找到适合数据的模型。估计问题专注于似然函数，或更精确地说是其负对数。对于由随机变量 x 表示的数据和由 θ 参数化的概率密度族 $p(x | \theta)$ ，负对数似然由下式给出：

$$L_x(\theta) = -\log p(x | \theta). \quad (8.14)$$

符号 $L_x(\theta)$ 强调了参数 θ 是变化的而数据 x 是固定的这一事实。在写负对数似然时，我们经常省略对 x 的引用，因为它实际上是 θ 的函数，当表示数据不确定性的随机变量从上下文中清楚时，我们写作 $L(\theta)$ 。

让我们解释一下对于固定的 θ 值，概率密度 $p(x | \theta)$ 建模的是什么。它是一个分布，为给定参数设置建模数据的不确定性。对于给定数据集 x ，似然函数允许我们表达对参数 θ 不同设置的偏好，我们可以选择更“可能”生成数据的设置。

从互补的角度来看，如果我们认为数据是固定的(因为已经被观测到)，而我们改变参数 θ ， $L(\theta)$ 告诉我们什么？它告诉我们对于观测 x ， θ 的特定设置有多可能。基于这第二种观点，最大似然估计器给出了对于数据集最可能的参数 θ 。

我们考虑监督学习设置，其中我们获得对 $(x_1, y_1), \dots, (x_N, y_N)$ ，其中 $x_n \in \mathbb{R}^D$ ，标签 $y_n \in \mathbb{R}$ 。我们感兴趣构建一个预测器，它以特征向量 x_n 作为输入并产生预测 y_n (或接近它的东西)，即给定向量 x_n ，我们想要标签 y_n 的概率分布。换句话说，我们为特定参数设置 θ 指定给定样本的标签的条件概率分布。

示例8.4

经常使用的第一个例子是指定给定样本的标签的条件概率是高斯分布。换句话说，我们假设可以通过独立高斯噪声(参见第6.5节)来解释观测不确定性，该噪声具有零均值， $\epsilon_n \sim N(0, \sigma^2)$ 。我们进一步假设线性模型 $x_n^T \theta$ 用于预测。这意味着我们为每个样本标签对 (x_n, y_n) 指定高斯似然函数，

$$p(y_n | x_n, \theta) = N(y_n | x_n^T \theta, \sigma^2). \quad (8.15)$$

图8.3显示了给定参数 θ 的高斯似然函数的图解。我们将在第9.2节中看到如何根据高斯分布明确展开前面的表达式。

我们假设样本集 $(x_1, y_1), \dots, (x_N, y_N)$ 是独立的

[independently] 并且同分布(i.i.d.)。“独立”一词(第6.4.5节)[distributed]意味着涉及整个数据集的似然($Y = \{y_1, \dots, y_N\}$ 和 $X = \{x^{(1)}, \dots, x^{(N)}\}$)分解为每个单独样本似然的乘积

[机器学习数学草稿(2023-10-18)。反馈:] [https://mml-book.com][.] [8.3 参数估计] [267]

每个单独例子

$$p_Y(Y | X, \theta) = \prod_{n=1}^N p(y_n | x_n, \theta), \quad (8.16)$$

其中 $p(y_n | x_n, \theta)$ 是一个特定的分布（在例8.4中是高斯分布）。“同分布”这个表达意味着乘积(8.16)中的每一项都是同一种分布，并且它们都共享相同的参数。从优化角度来看，计算可以分解为更简单函数之和的函数通常更容易。因此，在机器学习中我们经常考虑负对数似然[回忆 $\log(ab) = \log(a) + \log(b)$]

$$L_X(\theta) = -\log p(Y | X, \theta) = -\sum_{n=1}^N \log p(y_n | x_n, \theta). \quad (8.17)$$

虽然容易解释 θ 在 $p(y_n | x_n, \theta)$ (8.15)条件符号右边的事实，因此应该被解释为观测到的和固定的，但这种解释是不正确的。负对数似然 $L(\theta)$ 是 θ 的函数。因此，为了找到一个能很好解释数据 $(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})$ 的好参数向量 θ ，要最小化关于 θ 的负对数似然 $L(\theta)$ 。

注记。(8.17)中的负号是一个历史遗迹，这是由于我们想要最大化似然的约定，但数值优化文献倾向于研究函数的最小化。◇

例8.5

继续我们的高斯似然例子(8.15)，负对数似然可以重写为

$$L_X(\theta) = -\sum_{n=1}^N \log p(y_n | x_n, \theta) = -\sum_{n=1}^N \log N(y_n | x_n^T \theta, \sigma^2) \quad (8.18a)$$

$$= \sum_{n=1}^N [-(y_n - x_n^T \theta)^2 / (2\sigma^2) - \log(1/\sqrt{2\pi\sigma^2}) \exp(-(y_n - x_n^T \theta)^2 / (2\sigma^2))] \quad (8.18b)$$

$$= \sum_{n=1}^N [-(y_n - x_n^T \theta)^2 / (2\sigma^2)] - \sum_{n=1}^N \log \exp(-(y_n - x_n^T \theta)^2 / (2\sigma^2)) - \sum_{n=1}^N \log(1/\sqrt{2\pi\sigma^2}) \quad (8.18c)$$

$$= (1/(2\sigma^2)) \sum_{n=1}^N (y_n - x_n^T \theta)^2 - \sum_{n=1}^N \log(1/\sqrt{2\pi\sigma^2}). \quad (8.18d)$$

由于 σ 是给定的，(8.18d)中的第二项是常数，最小化 $L(\theta)$ 对应于求解第一项中表达的最小二乘问题（与(8.8)比较）。

事实证明，对于高斯似然，得到的优化

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由Cambridge University Press出版 (2020).

[268] [[当模型遇到数据]]

图8.5 对于给定数据，参数的最大似然估计产生黑色对角线。橙色方块显示 $x = 60$ 处最大似然预测的值。



图8.5：对于给定数据，参数的最大似然估计产生黑色对角线

图8.5：对于给定数据，参数的最大似然估计产生黑色对角线

图8.6 比较 $x = 60$ 处最大似然估计和MAP估计的MLE预测。先验使斜率偏向不那么陡峭，截距偏向更接近零。在这个例子中，使截距更接近零的偏差实际上增加了斜率。



图8.6：比较最大似然估计和MAP估计的预测

图8.6：比较最大似然估计和MAP估计的预测

对应于最大似然估计的问题有闭式解。我们将在第9章看到更多细节。图8.5显示了一个回归数据集和由最大似然参数产生的函数。最大似然估计可能遭受过拟合（第8.3.3节），类似于未正则化的经验风险最小化（第8.2.3节）。对于其他似然函数，即如果我们用非高斯分布建模噪声，最大似然估计可能没有闭式解析解。在这种情况下，我们采用第7章讨论的数值优化方法。

8.3.2 最大后验估计

如果我们对参数 θ 的分布有先验知识，我们可以在似然上乘以一个额外项。这个额外项是参数上的先验概率分布 $p(\theta)$ 。对于给定的先验，在

[机器学习数学草稿（2023-10-18）。反馈：] [https://mml-book.com][.] [8.3 参数估计] [269]

观测到一些数据 x 后，我们应该如何更新 θ 的分布？换句话说，在观测到数据 x 后，我们应该如何表示我们对 θ 有了更具体的知识这一事实？贝叶斯定理，如第6.3节所讨论的，给了我们一个有原则的工具来更新随机变量的概率分布。它允许我们从一般的先验陈述（先验分布） $p(\theta)$ 和链接参数 θ 和观测数据 x 的函数 $p(x|\theta)$ （称为似然）计算参数 θ 上的后验分布 $p(\theta|x)$ （更具体的知识）：

$$p(\theta|x) = p(x|\theta)p(\theta)/p(x). \quad (8.19)$$

回忆我们感兴趣找到最大化后验的参数 θ 。由于分布 $p(x)$ 不依赖于 θ ，我们可以在优化中忽略分母的值并获得

$$p(\theta|x) \propto p(x|\theta)p(\theta). \quad (8.20)$$

前面的比例关系隐藏了数据的密度 $p(x)$ ，这可能难以估计。我们现在不是估计负对数似然的最小值，而是估计负对数后验的最小值，这被称为最大后验估计（MAP estimation）。添加零均值的效果的图示

在图8.6中显示了高斯先验。[[estimation]]

[MAP estimation]

例8.6

除了在前面例子中假设高斯似然之外，我们假设参数向量分布为零均值的多元高斯分布，即 $p(\theta) = N(0, \Sigma)$ ，其中 Σ 是协方差矩阵（第6.5节）。注意高斯分布的共轭先验也是高斯分布（第6.6.1节），因此我们期望后验分布也是高斯分布。我们将在第9章中看到最大后验估计的详细内容。

在机器学习中，包含关于良好参数位置的先验知识这一想法是广泛存在的。我们在第8.2.3节中看到的另一种观点是正则化的思想，它引入了一个额外项，使得结果参数偏向于接近原点。最大后验估计可以被认为是连接非概率和概率世界的桥梁，因为它明确承认了先验分布的必要性，但仍然只产生参数的点估计。

备注。最大似然估计 θ_{ML} 具有以下性质 [(Lehmann and Casella, 1998; Efron and Hastie, 2016)]：

渐近一致性：MLE 在无限多观测的极限下收敛到真实值，加上一个近似正态的随机误差。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[270] [[When Models Meet Data]]

图8.7 模型拟合。在参数化模型类 M_θ 中，我们优化模型参数 θ^* 来最小化与真实（未知）模型 M^* 的距离。

达到这些性质所需的样本大小可能相当大。

误差的方差以 $1/N$ 的速度衰减，其中 N 是数据点的数量。

特别是在“小”数据情况下，最大似然估计可能导致过拟合。

◇

最大似然估计（和最大后验估计）的原理使用概率建模来推理数据和模型参数中的不确定性。然而，我们还没有将概率建模发挥到极致。在本节中，结果训练过程仍然产生预测器的点估计，即训练返回代表最佳预测器的一组参数值。在第8.4节中，我们将采用参数值也应被视为随机变量的观点，而不是估计该分布的“最佳”值，我们将在进行预测时使用完整的参数分布。

8.3.3 模型拟合

考虑我们给定一个数据集，并且我们有兴趣将参数化模型拟合到数据的情况。当我们谈论“拟合”时，我们通常指优化/学习模型参数，使它们最小化某个损失函数，例如负对数似然。通过最大似然（第8.3.1节）和最大后验估计（第8.3.2节），我们已经讨论了两种常用的模型拟合算法。

模型的参数化定义了一个我们可以操作的模型类 M_θ 。例如，在线性回归设置中，我们可以定义输入 x 和（无噪声）观测 y 之间的关系为 $y = ax + b$ ，其中 $\theta := \{a, b\}$ 是模型参数。在这种情况下，模型参数 θ 描述了仿射函数族，即斜率为 a 的直线，从 0 偏移 b 。假设数据来自

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] [<https://mml-book.com>][.] [8.3 Parameter Estimation] [271]



图8.8 通过最大似然拟合不同模型类到回归数据集

(a) 过拟合 (b) 欠拟合 (c) 拟合良好

一个未知的模型 M 。对于给定的训练数据集，我们优化 θ 使得 M_{θ^*} 尽可能接近 M ，其中“接近程度”由我们优化的目标函数定义（例如，训练数据上的平方损失）。图8.7说明了一个设置，其中我们有一个小的模型类（由圆圈 M_θ 表示），而数据生成模型 M 位于所考虑的模型集合之外。我们从 M_{θ_0} 开始参数搜索。在优化之后，即当我们获得最佳可能参数 θ^* 时，我们区分三种不同情况：(i) 过拟合，(ii) 欠拟合，和 (iii) 拟合良好。我们将给出这三个概念含义的高层直觉。

粗略地说，过拟合指的是参数化模型类过于丰富而无法对由 M^* 生成的数据集进行建模的情况，即 M_θ 可以建模更复杂的数据集。例如，如果数据集由线性函数生成，而我们将 M_θ 定义为七阶多项式类，我们不仅可以建模线性函数，还可以建模二次、三次等多项式。过拟合的模型通常具有大量参数。我们经常观察到的是，过度灵活的模型类 M_θ 使用其所有建模能力来减少训练误差。如果训练数据有噪声，它因此会在噪声本身中找到一些有用的信号。这将在我们远离训练数据进行预测时造成巨大问题。图8.8(a)给出了

[[在实践中检测过拟合的一种方法是观察模型具有低训练风险但高]]

过度拟合(overfitting)在回归上下文中的例子，其中模型参数通过最大似然估计学习(参见第8.3.1节)。

我们将在第9.2.2节更详细地讨论回归中的过度拟合。

当我们遇到**欠拟合(underfitting)**时，我们遇到了相反的问题，即模型类 $M[0]$ 不够丰富。例如，如果我们的数据集是由正弦函数生成的，但 0 只参数化直线，那么最佳优化程序不会让我们接近真实模型。然而，我们仍然优化参数并找到最好的直线来建模数据集。图8.8(b)显示了一个模型欠拟合的例子，因为它的灵活性不够。欠拟合的模型通常参数较少。

第三种情况是参数化模型类大致正确。然后，我们的模型拟合良好，即它既不过拟合也不欠拟合。这意味着我们的模型类刚好足够丰富来描述给定的数据集。

图8.8(c)显示了一个相当好地拟合给定数据集的模型。理想情况下，

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

当模型遇见数据

这是我们想要使用的模型类，因为它具有良好的泛化特性。

在实践中，我们经常定义非常丰富的模型类 $M[0]$ ，具有许多参数，如深度神经网络。为了缓解过拟合问题，我们可以使用正则化(第8.2.3节)或先验(第8.3.2节)。

我们将在第8.6节讨论如何选择模型类。

8.3.4 延伸阅读

当考虑概率模型时，最大似然估计原理推广了线性模型的最小二乘回归思想，我们将在第9章详细讨论。当将预测器限制为具有线性形式，并在输出上应用额外的非线性函数 φ 时，即，

$$p(y[n] \mid \mathbf{x}[n], \theta) = \varphi(\theta^T \mathbf{x}[n]), \quad (8.21)$$

我们可以考虑其他预测任务的其他模型，如二元分类或建模计数数据(McCullagh and Nelder, 1989)。另一种观点是考虑来自指数族的似然(第6.6节)。参数与数据之间具有线性依赖关系，并且具有潜在非线性变换 φ (称为**链接函数(link function)**)的这类模型，被称为**广义线性模型(generalized linear models)**(Agresti, 2002, 第4章)。

最大似然估计有着丰富的历史，最初由Ronald Fisher爵士在1930年代提出。我们将在第8.4节扩展概率模型的思想。使用概率模型的研究人员之间的一个争论是Bayesian和频率统计学之间的讨论。如第6.1.1节所述，这归结为概率的定义。回想第6.1节，人们可以认为概率是逻辑推理的推广(通过允许不确定性)(Cheeseman, 1985; Jaynes, 2003)。最大似然估计方法本质上是频率主义的，感兴趣的读者可以参考Efron and Hastie (2016)以获得Bayesian和频率统计学的平衡观点。

有一些概率模型可能无法进行最大似然估计。读者可以参考更高级的统计教科书，例如Casella and Berger (2002)，了解诸如矩方法、M估计和估计方程等方法。

8.4 概率建模与推断

在机器学习中，我们经常关注数据的解释和分析，例如，用于预测未来事件和决策制定。为了使这项任务更易处理，我们经常构建描述生成观察数据的**生成过程(generative process)**的模型。

Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>. 8.4 概率建模与推断 273

例如，我们可以用两个步骤来描述抛硬币实验的结果(“正面”或“反面”)。首先，我们定义一个参数 μ ，它描述“正面”的概率作为Bernoulli分布的参数(第6章)；其次，我们可以从Bernoulli分布 $p(x | \mu) = \text{Ber}(\mu)$ 中采样一个结果 $x \in \{\text{正面}, \text{反面}\}$ 。参数 μ 产生特定的数据集 X ，并依赖于所使用的硬币。由于 μ 事先未知且永远无法直接观察到，我们需要机制来根据抛硬币实验的观察结果学习关于 μ 的一些信息。接下来，我们将讨论如何使用概率建模来实现这个目的。

8.4.1 概率模型

概率模型将实验的不确定方面表示为概率分布。使用概率模型的好处是它们为建模、推断、预测和模型选择提供了来自概率论(第6章)的统一且一致的工具集。

在概率建模中，观察变量 x 和隐藏参数 θ 的联合分布 $p(x, \theta)$ 至关重要：它封装了以下信息：

先验和似然(乘积规则，第6.3节)。

边际似然 $p(x)$ ，在模型选择中起重要作用(第8.6节)，可以通过取联合分布并积分出参数来计算(求和规则，第6.3节)。

后验，可以通过将联合分布除以边际似然来获得。

只有联合分布具有这个性质。因此，概率模型由其所有随机变量的联合分布来指定。

8.4.2 Bayesian推理

机器学习的一个关键任务是根据模型和数据来揭示模型隐藏变量 θ 在给定观测变量 x 的值。在第8.3.1节中，我们已经讨论了使用最大似然或最大后验估计来估计模型参数 θ 的两种方法。在这两种情况下，我们获得 θ 的单一最佳值，因此参数估计的关键算法问题是求解优化问题。一旦这些点估计 θ 已知，我们就使用它们进行预测。更具体地说，预测分布将是 $p(x|\theta)$ ，其中我们在似然函数中使用 θ^* 。

如第6.3节所讨论的，仅关注后验分布的某些统计量（如最大化后验的参数 θ ）会导致信息丢失，这在使用预测 $p(x|\theta)$ 进行决策的系统中可能是关键的。这些决策系统通常具有与似然不同的目标函数，如平方误差损失或错误分类误差。因此，拥有完整的后验分布会极其有用，并能做出更稳健的决策。**Bayesian推理**就是要找到这个后验分布(Gelman et al., 2004)。对于数据集 X 、参数先验 $p(\theta)$ 和似然函数，后验为

$$p(\theta|X) = p(X|\theta)p(\theta)/p(X), p(X) = \int p(X|\theta)p(\theta)d\theta, (8.22)$$

通过应用Bayes定理获得。关键思想是利用Bayes定理来逆转参数 θ 和数据 X 之间的关系（由似然给出），以获得后验分布 $p(\theta|X)$ 。

拥有参数后验分布的含义是它可以用来将不确定性从参数传播到数据。更具体地说，对于参数上的分布 $p(\theta)$ ，我们的预测将是

$$p(x) = \int p(x|\theta)p(\theta)d\theta = E_{\theta}[p(x|\theta)], (8.23)$$

它们不再依赖于已被边际化/积分掉的模型参数 θ 。方程(8.23)揭示了预测是所有合理参数值 θ 上的平均值，其中合理性由参数分布 $p(\theta)$ 封装。

在第8.3节讨论了参数估计，这里讨论了Bayesian推理之后，让我们比较这两种学习方法。通过最大似然或MAP估计的参数估计产生参数的一致点估计 θ^* ，需要解决的关键计算问题是优化。相比之下，Bayesian推理产生一个（后验）分布，需要解决的关键计算问题是积分。使用点估计进行预测很直接，而Bayesian框架中的预测需要解决另一个积分问题；见(8.23)。然而，Bayesian推理为我们提供了一种原则性的方式来融入先验知识、考虑辅助信息和融入结构知识，这些在参数估计的上下文中都不容易做到。此外，将参数不确定性传播到预测在决策系统对风险评估和数据高效学习上下文中的探索可能很有价值(Deisenroth et al., 2015; Kamthe and Deisenroth, 2018)。

虽然Bayesian推理是学习参数和进行预测的数学原则性框架，但由于我们需要解决的积分问题，它也带来了一些实际挑战；见(8.22)和(8.23)。更具体地说，如果我们不选择参数的共轭先验（第6.6.1节），(8.22)和(8.23)中的积分在解析上不可处理，我们无法以封闭形式计算后验、预测或边际似然。在这些情况下，我们需要求助于近似方法。这里，我们可以使用随机近似，如Markov链Monte Carlo (MCMC) (Gilks et al., 1996)，或确定性近似，如Laplace近似(Bishop, 2006; Barber, 2012; Murphy, 2012)、变分推理(Jordan et al., 1999; Blei et al., 2017)或期望传播(Minka, 2001a)。

尽管存在这些挑战，Bayesian推理已成功应用于各种问题，包括大规模主题建模(Hoffman et al., 2013)、点击率预测(Graepel et al., 2010)、控制系统中的数据高效强化学习(Deisenroth et al., 2015)、在线排名系统(Herbrich et al., 2007)和大规模推荐系统。有一些通用工具，如Bayesian优化(Brochu et al., 2009; Snoek et al., 2012; Shahriari et al., 2016)，是高效搜索模型或算法元参数的非常有用的组件。

注释: 在机器学习文献中, (随机) “变量”和“参数”之间可能存在某种程度上任意的分离。虽然参数是被估计的 (例如, 通过maximum likelihood), 变量通常是被边缘化处理的。在本书中, 我们对这种分离并不那么严格, 因为原则上, 我们可以对任何参数放置一个prior并将其积分出去, 这样根据前述分离, 参数就会变成一个随机变量。◇

8.4.3 Latent-Variable模型

在实践中, 有时在模型中引入额外的 *latent variables* z (除了模型参数 θ 之外) 是有用的 [(Moustaki et al., 2015)]。这些 *latent variables* 与模型参数 θ 不同, 因为它们不会显式地参数化模型。*Latent variables* 可能描述数据生成过程, 从而有助于模型的可解释性。它们还经常简化模型的结构, 并允许我们定义更简单和更丰富的模型结构。模型结构的简化通常与较少的模型参数数量密切相关 [(Paquet, 2008; Murphy, 2012)]。在latent-variable模型中的学习 (至少通过maximum likelihood) 可以使用expectation maximization (EM)算法以原则性的方式完成 [(Dempster et al., 1977; Bishop, 2006)]。这种 *latent variables* 有用的例子包括: 用于降维的principal component analysis (第10章), 用于密度估计的Gaussian mixture models (第11章), hidden Markov models [(Maybeck, 1979)]或用于时间序列建模的dynamical systems [(Ghahramani and Roweis, 1999; Ljung, 1999)], 以及meta learning和任务泛化 [(Hausman et al., 2018; Sæmundsson et al., 2018)]。尽管引入这些 *latent variables* 可能会使模型结构和生成过程更容易理解, 但在latent-variable模型中的学习通常是困难的, 正如我们将在第11章中看到的那样。

由于latent-variable模型也允许我们定义从参数生成数据的过程, 让我们来看看这个生成过程。用 x 表示数据, 用 θ 表示模型参数, 用 z 表示latent variables, 我们得到条件分布

$$p(x | z, \theta) \quad (8.24)$$

它允许我们为任何模型参数和latent variables生成数据。鉴于 z 是latent variables, 我们在它们上放置一个prior $p(z)$ 。

如同我们之前讨论的模型一样, 具有latent variables的模型可以在我们在第8.3节和第8.4.2节中讨论的框架内用于参数学习和推理。为了促进学习 (例如, 通过maximum likelihood estimation或Bayesian inference), 我们遵循一个两步过程。首先, 我们计算模型的likelihood $p(x | \theta)$, 它不依赖于latent variables。其次, 我们使用这个likelihood进行参数估计或Bayesian inference, 其中我们使用与第8.3节和第8.4.2节中完全相同的表达式。

由于likelihood函数 $p(x | \theta)$ 是给定模型参数的数据的预测分布, 我们需要将latent variables边缘化出去, 使得

$$\int p(x | \theta) = p(x | z, \theta)p(z)dz, \quad (8.25)$$

其中 $p(x | z, \theta)$ 在(8.24)中给出, $p(z)$ 是latent variables上的prior。注意likelihood不能依赖于latent variables z , 它只是数据 x 和模型参数 θ 的函数。

(8.25)中的likelihood直接允许通过maximum likelihood进行参数估计。MAP estimation在第8.3.2节中讨论的模型参数 θ 上的额外prior下也是直接的。此外, 使用likelihood (8.25), 在latent-variable模型中的Bayesian inference (第8.4.2节) 以通常的方式工作: 我们在模型参数上放置一个prior $p(\theta)$, 并使用Bayes定理获得后验分布

$$p(\theta | X) = p(X | \theta)p(\theta) / p(X) \quad (8.26)$$

给定数据集 X 的模型参数上的后验分布。(8.26)中的后验可以在Bayesian inference框架内用于预测; 见(8.23)。

我们在这个latent-variable模型中面临的一个挑战是likelihood $p(X | \theta)$ 需要根据(8.25)对latent variables进行边缘化。除非我们为 $p(x | z, \theta)$ 选择一个共轭prior $p(z)$, 否则(8.25)中的边缘化在解析上是不可处理的, 我们要求助于近似方法 [(Bishop, 2006; Paquet, 2008; Murphy, 2012; Moustaki et al., 2015)]。

类似于参数后验(8.26)，我们可以根据以下方式计算latent variables上的后验

$$p(z | X) = p(X | z)p(z) / p(X), p(X | z) = \int p(X | z, \theta)p(\theta)d\theta, (8.27)$$

其中 $p(z)$ 是latent variables上的prior, $p(X | z)$ 需要我们积分出模型参数 θ 。

鉴于解析求解积分的困难，很明显，同时边际化latent variables和模型参数在一般情况下是不可能的[(Bishop, 2006; Murphy, 2012)]。一个更容易计算的量是latent variables上的后验分布，但以模型参数为条件，即

$$p(z | X, \theta) = p(X | z, \theta)p(z) / p(X | \theta), (8.28)$$

其中 $p(z)$ 是latent variables上的prior, $p(X | z, \theta)$ 在(8.24)中给出。

在第[10]章和第[11]章中，我们分别推导了PCA和高斯混合模型的似然函数。此外，我们还计算了PCA和高斯混合模型中潜在变量的后验分布[(8.28)]。

注释：在接下来的章节中，我们可能不会在潜在变量 z 和不确定的模型参数 θ 之间做出如此明确的区分，也会将模型参数称为“潜在”或“隐藏”变量，因为它们是不可观测的。在第[10]章和第[11]章中，当我们使用潜在变量 z 时，我们会注意这种差异，因为我们将有两种不同类型的隐藏变量：模型参数 θ 和潜在变量 z 。[◇]

我们可以利用概率模型的所有元素都是随机变量这一事实来定义一种统一的语言来表示它们。在第[8.5]节中，我们将看到一种简洁的图形语言来表示概率模型的结构。我们将使用这种图形语言来描述后续章节中的概率模型。

8.4.4 延伸阅读

机器学习中的概率模型[(Bishop, 2006; Barber, 2012; Murphy, 2012)]为用户提供了一种以原则性方式捕获数据和预测模型不确定性的方法。[Ghahramani (2015)]对机器学习中的概率模型进行了简短回顾。给定一个概率模型，我们可能足够幸运能够分析地计算感兴趣的参数。然而，一般来说，分析解很少见，需要使用计算方法，如采样[(Gilks et al., 1996; Brooks et al., 2011)]和变分推断[(Jordan et al., 1999; Blei et al., 2017)]。[Moustaki et al. (2015)]和[Paquet (2008)]对潜在变量模型中的贝叶斯推断提供了很好的概述。

近年来，出现了几种编程语言，旨在将软件中定义的变量视为对应于概率分布的随机变量。目标是能够编写概率分布的复杂函数，同时编译器在底层自动处理贝叶斯推断的规则。这个快速变化的领域被称为 *概率编程*(probabilistic programming)。

8.5 有向图模型

在本节中，我们介绍一种用于指定概率模型的图形语言，称为**有向图模型**(directed graphical model)。它提供了一种紧凑简洁的方式来指定概率模型，并允许读者可视化地解析随机变量之间的依赖关系。图模型可视化地捕获了所有随机变量上的联合分布如何分解为仅依赖于这些变量子集的因子乘积的方式。在第[8.4]节中，我们确定了概率模型的联合分布是关键的兴趣量，因为它包含了关于先验、似然和后验的信息。然而，联合分布本身可能非常复杂，并且它没有告诉我们概率模型结构特性的任何信息。例如，联合分布 $p(a, b, c)$ 没有告诉我们任何关于独立性关系的信息。这就是图模型发挥作用的地方。本节依赖于第[6.4.5]节中描述的独立性和条件独立性概念。

在**图模型**(graphical model)中，节点是随机变量。在图[8.9(a)]中，节点表示随机变量 a 、 b 、 c 。边表示变量之间的概率关系，例如条件概率。

注释：并非每个分布都可以用特定选择的图模型来表示。相关讨论可以在[Bishop (2006)]中找到。[◇]

概率图模型具有一些便利的特性：

- 它们是可视化概率模型结构的简单方法
- 它们可以用于设计或激发新类型的统计模型
- 仅检查图就能让我们洞察特性，例如条件独立性
- 统计模型中推断和学习的复杂计算可以用图操作来表达

8.5.1 图语义

有向图模型(directed graphical model)/**贝叶斯网络**(Bayesian network)是一种表示概率模型中条件依赖关系的方法。它们提供了条件概率的可视化描述，因此为描述复杂的相互依赖关系提供了一种简单的语言。模块化描述也带来了计算简化。两个节点(随机变量)之间的有向链接(箭头)表示条件概率。例如，图[8.9(a)]中 a 和 b 之间的箭头给出了给定 a 的 b 的条件概率 $p(b | a)$ 。



图8.9 有向图模型示例

图8.9 有向图模型示例

(a) 完全连接 (b) 非完全连接

如果我们知道联合分布的某些分解信息，就可以从联合分布导出有向图模型。

例8.7

考虑三个随机变量 a 、 b 、 c 的联合分布

$$p(a, b, c) = p(c | a, b)p(b | a)p(a) \quad (8.29)$$

联合分布的分解

在[(8.29)]中告诉我们随机变量之间的关系：

c直接依赖于a和b。

b直接依赖于a。

a既不依赖于b也不依赖于c。

对于[(8.29)]中的因式分解，我们得到图[8.9(a)]中的有向图模型。

一般来说，我们可以从因式分解的联合分布构造相应的有向图模型，步骤如下：

1. 为所有随机变量创建节点。
2. 对于每个条件分布，我们从对应于分布所依赖变量的节点向图中添加有向链接（箭头）。

图的布局

图的布局取决于联合分布的因式分解选择。

我们讨论了如何从已知的联合分布因式分解得到相应的有向图模型。现在，我们将做完全相反的事情，描述如何从给定的图模型中提取一组随机变量的联合分布。

例8.8

观察图[8.9(b)]中的图模型，我们利用两个性质：

我们寻求的联合分布 $p(x_1, \dots, x_5)$ 是一组条件概率的乘积，图中每个节点对应一个条件概率。在这个特定例子中，我们需要五个条件概率。

每个条件概率只依赖于图中相应节点的父节点。例如， x_4 将以 x_2 为条件。

这两个性质产生了联合分布的期望因式分解

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_5)p(x_2 | x_5)p(x_3 | x_1, x_2)p(x_4 | x_2). \quad (8.30)$$

一般来说，联合分布 $p(x) = p(x_1, \dots, x_k)$ 给出为

$$p(\prod_{k=1}^k x_k) = \prod_{k=1}^k p(x_k | Pa_k), \quad (8.31)$$

其中 Pa_k 表示“ x_k 的父节点”。 x_k 的父节点是指向 x_k 的节点。

我们用抛硬币实验的具体例子来结束这个小节。考虑一个Bernoulli实验(例[6.8])，其中这个实验结果 x 为“正面”的概率是

$$p(x | \mu) = \text{Ber}(\mu). \quad (8.32)$$

现在我们重复这个实验 N 次，观察结果 $x_1, \dots, x_n$ ，从而得到联合分布

$$p(x_1, \dots, x_n | \mu) = \prod_{n=1}^N p(x_n | \mu). \quad (8.33)$$

右边的表达式是每个单独结果上Bernoulli分布的乘积，因为实验是独立的。回忆第[6.4.5]节，统计独立意味着分布可以因式分解。为了写出这种设置的图模型，我们区分未观测的/潜在变量和观测变量。在图形上，观测变量用阴影节点表示，因此我们得到图[8.10(a)]中的图模型。我们看到单个参数 μ 对所有 $x_n, n = 1, \dots, N$ 都是相同的，因为结果 x_n 是同分布的。这种设置的更紧凑但等价的图模型如图[8.10(b)]所示，其中我们使用

图8.10

重复Bernoulli实验的图模型

板记号(plate notation)。板(盒子)重复内部的所有内容(在这种情况下是观测 x_n) N 次。因此，两个图模型是等价的，但板记号更紧凑。图模型立即允许我们在 μ 上放置超先验。超先验(hyperprior)是先验分布第一层参数上的第二层先验分布。图[8.10(c)]在潜在变量 μ 上放置了Beta(α, β)先验。如果我们将 α 和 β 视为确定参数，即不是随机变量，我们省略它周围的圆圈。

8.5.2 条件独立和d-分离

有向图模型允许我们仅通过观察图就找到联合分布的条件独立(第[6.4.5]节)关系性质。称为*d-分离*(Pearl, 1988)的概念是关键。

考虑一个一般的有向图，其中A、B、C是任意非相交的节点集合(它们的并集可能小于图中节点的完整集合)。我们希望确定一个特定的条件独立语句”A在给定的C的条件下与B条件独立”，记为

$$A \perp\!\!\!\perp B \mid C, (8.34)$$

是否由给定的有向无环图暗示。为此，我们考虑从A中任何节点到B中任何节点的所有可能路径(忽略箭头方向的路径)。如果任何此类路径包含满足以下任一条件的节点，则称该路径被阻塞：

路径上的箭头在该节点处头尾相接或尾尾相接，且该节点在集合C中。

箭头在该节点处头头相接，且该节点及其任何后代都不在集合C中。

如果所有路径都被阻塞，则称A被C *d-分离*于B，图中所有变量的联合分布将满足 $A \perp\!\!\!\perp B \mid C$ 。



三种类型的图模型：(a)有向图模型(Bayesian网络)；(b)无向图

[图形模型] [(a) 有向图形模型] [(b)] [无向] [图形] [(c) 因子图] [(马尔可夫随机] [模型]

[域]；[(c) 因子]

[图.]

例子 8.9 (条件独立)

[图 8.11] [a] [b] [c] [D-分离]

[例子.]

[d]

[c]

考虑图 8.11 中的图形模型。通过视觉检查我们得到

$$b \perp\!\!\!\perp d \mid a, c \text{ [(8.35)]}$$

$$a \perp\!\!\!\perp c \mid b \text{ [(8.36)]}$$

$$b/\!\!\!\perp\!\!\!\perp d \mid c \text{ [(8.37)]}$$

$$a/\!\!\!\perp\!\!\!\perp c \mid b, c \text{ [(8.38)]}$$

有向图形模型允许对概率模型进行紧凑表示，我们将在第 9、10 和 11 章中看到有向图形模型的例子。这种表示以及条件独立的概念，使我们能够将相应的概率模型分解为更容易优化的表达式。

概率模型的图形表示使我们能够直观地看到我们对模型结构所做的设计选择的影响。我们经常需要对模型的结构做出高层次的假设。这些建模假设（超参数）会影响预测性能，但不能直接使用我们迄今为止所见的方法来选择。我们将在第 8.6 节中讨论选择结构的不同方法。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] [<https://mml-book.com>][.](<https://mml-book.com>)
[8.6 Model Selection] [283]

8.5.3 进一步阅读

概率图形模型的介绍可以在 Bishop (2006, 第8章) 中找到，不同应用的详细描述和相应的算法含义可以在 Koller and Friedman (2009) 的书中找到。概率图形模型主要有三种类型：

[有向图形]

有向图形模型 (贝叶斯网络); 见图 8.12(a) [模型]

无向图形模型 [贝叶斯网络] (马尔可夫随机域); 见图 8.12(b)

因子图 [无向图形]; 见图 8.12(c)

[模型]

图形模型允许基于图的推理和学习算法，例如通过局部消息传递。应用范围从在线游戏排名 (Herbrich et al., 2007) 和计算机视觉（例如图像分割、语义标记、图像去噪、图像恢复 (Kittler and Föglein, 1984; Sucar and Gillies, 1994; Shotton et al., 2006; Szeliski et al., 2008)）到编码理论 (McEliece et al., 1998)、求解线性方程组 (Shental et al., 2008)，以及信号处理中的迭代贝叶斯状态估计 (Bickson et al., 2007; Deisenroth and Mohamed, 2012)。

在实际应用中特别重要但我们在本书中不讨论的一个主题是结构化预测 (Bakir et al., 2007; Nowozin et al., 2014) 的思想，它允许机器学习模型处理结构化的预测，例如序列、树和图。神经网络模型的流行使得更灵活的概率模型得以使用，导致结构化模型的许多有用应用 (Goodfellow et al., 2016, 第16章)。近年来，由于图形模型在因果推理中的应用，人们对图形模型重新产生了兴趣 (Pearl, 2009; Imbens and Rubin, 2015; Peters et al., 2017; Rosenbaum, 2017)。

8.6 Model Selection

在机器学习中，我们经常需要做出高层次的建模决策，这些决策会对模型的性能产生关键影响。我们所做的选择（例如，似然的函数形式）会影响模型中自由参数的数量和类型，从而也影响模型的灵活性和表达能力。更复杂的模型在某种意义上更灵活，因为它们可以用来描述更多的数据集。例如，[多项式] 2次多项式 $y = a_0 + a_1x + a_2x^2$ [也可以描述] 1次多项式（直线 $y = a_0 + a_1x$ ）只能用来描述输入 x 和观测 y 之间的线性关系。2次多项式还可以额外描述输入和观测之间的二次关系。[通过设置] [线性函数] [$a_2 = 0$ ，即] [它严格比] [一阶] [多项式更具表达力。]

现在人们可能会认为非常灵活的模型通常比简单模型更可取，因为它们更具表达力。一个普遍的问题

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[284] [*When Models Meet Data*]

[图 8.13] [嵌套]

[交叉验证。我们] [所有标记数据] [执行两个层次]

[的] [K]-折]

[所有训练数据] [测试数据]

[交叉验证。]

[训练模型用] [验证]

是在训练时我们只能使用训练集来评估模型的性能并学习其参数。然而，训练集上的性能并不是我们真正感兴趣的。在第 8.3 节中，我们已看到最大似然估计可能导致过拟合，特别是当训练数据集很小时。理想情况下，我们的模型在测试集上（训练时不可用）也能表现良好。因此，我们需要一些机制来评估模型如何泛化到未见的测试数据。*Model selection* 正是关注这个问题。

8.6.1 嵌套交叉验证

我们已经看到了一种可用于 model selection 的方法（第 8.2.4 节中的交叉验证）。回想一下，交叉验证通过反复将数据集分割为训练和验证集来提供泛化误差的估计。我们可以再次应用这个思想，即对于每次分割，我们可以执行另一轮交叉验证。这就是

[嵌套] 有时被称为 *嵌套交叉验证*；见图 [8.13]。内层 [交叉验证] [层级用于估计在内部 validation set 上特定模型选择] 或超参数的性能。外层用于估计内层循环选择的最佳模型的泛化性能。我们可以在内层循环中测试不同的模型和超参数选择。为了区分这两个层级，用于估计 [test set] 泛化性能的集合通常被称为 *test set*，用于 [validation set] 选择最佳模型的集合被称为 *validation set*。内层循环通过使用 validation set 上的经验误差来近似估计给定模型泛化误差的期望值 [(8.39)]，[标准误差] [定义为] $\sqrt{\sigma^2/K}$

[实验数量] $E[V][R(V | M)] \approx R(V | M)$, (8.39) K [其中] K [是] X [的] k [的] 1

[和] $k=1$] σ

[是模型的标准] 其中 $R(V | M)$ 是 [validation set] M 上的经验风险（例如，均方根误差）。我们对所有模型 V 重复此过程，[每个实验的风险][偏差]

选择表现最佳的模型。注意 cross-validation 不仅给出了期望泛化误差，我们还可以获得高阶统计量，例如标准误差，即对

图8.14

[Evidence] [[贝叶斯推断]]

[体现了 Occam' s]

[razor。]

[横轴]

$p([描述了所有可能] [D])$

M]

[数据集的空间]

[datasets] $[D]$ 。[Evidence（纵

[轴）评估模型]

[预测可用]

$p([D] [M][2]D)$ [[数据的能力。由于]]

$p([([D] [M] i)D)$ [需要] [积分为] $[1]$ ，我们

D [[应该选择]]

C [[evidence最大的]]

[模型。]

[改编]

意味着不确定性的估计。一旦选择了模型，我们可以在 test set 上评估最终 [[自 [MacKay]]]

性能。[[(2003)。]]

8.6.2 贝叶斯模型选择

有许多模型选择的方法，本节涵盖了其中一些。一般来说，它们都试图在模型复杂性和数据拟合之间进行权衡。我们假设简单模型比复杂模型更不容易过拟合，因此模型选择的目标是找到能够合理解释数据的最简单模型。这个概念也被称为 *Occam's razor*。[[Occam's razor]]

注释。如果我们将模型选择视为假设检验问题，我们正在寻找与数据一致的最简单假设[[Mur-

[phy, 2012]]。◇

人们可能考虑在模型上放置一个偏向简单模型的先验。然而，这样做并不必要：一个“自动的Occam's Razor”在贝叶斯概率的应用中得到了定量体现[[Smith]

[and Spiegelhalter, 1980;] [Jefferys and Berger, 1992;] [MacKay, 1992]]。图[8.14]，改编自[MacKay (2003)]，给出了为什么复杂且表达能力很强的模型对于建模给定数据集D可能是不太可能选择的基本直觉。让我们将横轴[[这些预测]]

表示所有可能数据集的空间[[通过一个] D。如果我们对给定数据集D的模型M[i]的后验概率 $p(M[i] | D)$ 感兴趣[归一化]，我们可以[概率]使用贝叶斯定理。假设所有模型的均匀先验 $p(M)$ [分布在][D][,]

贝叶斯定理根据模型对发生数据的预测程度来奖励模型。给定模型[[即，它需要]]

M[i]对数据的这种预测， $p(D | M[i])$ ，被称为M[i]的 *evidence*。简单模型M1只能[[积分/求和为] 1]

预测少量数据集，这由 $p(D | M1)$ 所示；更强大的模型M[2]，例如，比M1有更多自由参数，能够[evidence]

当模型遇见数据

预测更大variety的数据集。然而，这意味着M[2]

在区域C中预测数据集的能力不如M1。假设两个模型被分配了相等的先验概率。那么，如果数据集落在区域C中，能力较弱的模型M1是更可能的模型。

在本章前面，我们论证了模型需要能够解释数据，即应该有一种从给定模型生成数据的方法。

此外，如果模型已经从数据中适当学习，那么我们期望生成的数据应该与经验数据相似。为此，将模型选择表述为分层推断问题是有帮助的，这允许我们计算模型上的后验分布。

让我们考虑有限数量的模型 $M = \{M1, \dots, M[K]\}$ ，其中[贝叶斯模型]每个模型M[k]具有参数 $\theta[k]$ 。在贝叶斯模型选择中，我们[选择]在模型集合上放置先验 $p(M)$ 。允许我们从此模型生成数据的相应 *generative* [generative process] *process*是 [图8.15]

[分层的图示] $M[k] \sim p(M)$ (8.40) [generative process] θ (8.41) $\sim p(\theta | M)$ [在贝叶斯模型] [k] [k]

[选择中。我们在] $p(D | \theta[k])$ (8.42) [模型集合上放置]

[先验] $p(\mathcal{M})$ 。[对于] $\mathcal{M}$ 并在图[8.15]中说明。给定训练集 $\mathcal{D}$ ，我们应用贝叶斯[每个模型，都有] 定理并计算模型上的后验分布为[相应的]

$$[\text{分布}] p(\mathcal{M}[k] \mid \mathcal{D}) \propto p(\mathcal{M}[k])p(\mathcal{D} \mid \mathcal{M}[k]). \quad (8.43)$$

[注意这个后验不再依赖于模型参数 $\theta[k]$]

[模型参数，]因为它们已经在贝叶斯设置中被积分出去了，[用于]

$$[\text{生成数据}] Z[\mathcal{D}] = p(\mathcal{D} \mid \mathcal{M}[k]) = \int p(\mathcal{D} \mid \theta[k])p(\theta[k] \mid \mathcal{M}[k])d\theta[k], \quad (8.44)$$

其中 $p(\theta[k] \mid \mathcal{M}[k])$ 是模型 $\mathcal{M}[k]$ 的模型参数 $\theta[k]$ 的先验分布。术语 [(8.44)] 被称为 *模型证据* 或 *边际似然*。从后验 [(8.43)] 中，我们确定 MAP 估计

$$\theta[*] = \arg \max_{\mathcal{M}[k]} p(\mathcal{M}[k] \mid \mathcal{D}). \quad (8.45)$$

使用均匀先验 $p(\mathcal{M}[k]) = 1/K$ ，它给每个模型相等的（先验）概率，在模型上确定 MAP 估计相当于选择使模型证据 [(8.44)] 最大化的模型。

备注（似然和边际似然）。似然和边际似然（证据）之间存在一些重要差异：虽然似然倾向于过拟合，但边际似然通常不会，因为模型参数已被边缘化（即，我们不再需要拟合参数）。此外，边际似然自动体现了模型复杂性和数据拟合之间的权衡（奥卡姆剃刀）。◇

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] [<https://mml-book.com>.] [8.6 Model Selection] [287]

8.6.3 用于模型比较的 Bayes 因子

考虑比较两个概率模型 M_1 、 M_2 在给定数据集 D 的问题。如果我们计算后验 $p(M_1 | D)$ 和 $p(M_2 | D)$ ，我们可以计算后验的比值

$$p(M_1 | D) / p(M_2 | D) = [p(D | M_1)p(M_1) / p(D)] / [p(D | M_2)p(M_2) / p(D)] = [p(D | M_1)p(M_1)] / [p(D | M_2)p(M_2)] \quad (8.46)$$

$p(M_1 | D) / p(M_2 | D)$ [后验几率] [先验几率] [Bayes 因子]

后验的比值也称为后验几率。[(8.46)] 右侧的第一个分数，先验几率，衡量我们的先验（初始）信念对 M_1 相对于 M_2 的偏好程度。边际似然的比值（右侧第二个分数）称为 Bayes 因子，衡量数据 D 被 M_1 预测得比 M_2 好的程度。

备注。Jeffreys-Lindley 悖论指出” Bayes 因子总是偏向更简单的模型，因为在具有扩散先验的复杂模型下数据的概率会非常小” [(Murphy, 2012)]。这里，扩散先验指的是不偏向特定模型的先验，即在此先验下许多模型先验上都是合理的。◇

如果我们对模型选择均匀先验，[(8.46)] 中的先验几率项为 1，即后验几率是边际似然的比值（Bayes 因子）

$$p(D | M_1) / p(D | M_2) \quad (8.47)$$

如果 Bayes 因子大于 1，我们选择模型 M_1 ，否则选择模型 M_2 。与频率统计类似，在考虑结果的”显著性”之前，对比值的大小为指导原则 [(Jeffreys, 1961)]。

备注（计算边际似然）。边际似然在模型选择中起重要作用：我们需要计算 Bayes 因子 [(8.46)] 和模型上的后验分布 [(8.43)]。

不幸的是，计算边际似然需要我们求解积分 [(8.44)]。这种积分通常在解析上是不可解的，我们必须求助于近似技术，例如数值积分 [(Stoer and Burlirsch, 2002)]、使用 Monte Carlo 的随机近似 [(Murphy, 2012)]，或 Bayesian Monte Carlo 技术 [(O’ Hagan, 1991; Rasmussen and Ghahramani, 2003)]。

然而，在特殊情况下我们可以求解它。在第 6.6.1 节中，我们讨论了共轭模型。如果我们选择共轭参数先验 $p(\theta)$ ，我们可以以封闭形式计算边际似然。在第 9 章中，我们将在线性回归的背景下准确地做到这一点。◇

我们在本章中简要介绍了机器学习的基本概念。在本书这一部分的其余部分，我们将看到第 8.2、8.3 和 8.4 节中三种不同风格的学习如何应用于机器学习的四大支柱（回归、降维、密度估计和分类）。

8.6.4 进一步阅读

我们在本节开始时提到，有高级建模选择会影响模型的性能。例子包括：

- 回归设置中多项式的次数
- 混合模型中成分的数量
- （深层）神经网络的网络架构
- 支持向量机中核的类型
- PCA 中潜在空间的维数
- 优化算法中的学习率（调度）

[在参数模型中，参数的数量通常与模型类的复杂性相关。] [Rasmussen and Ghahramani (2001)] 表明，自动奥卡姆剃刀不一定惩罚模型中参数的数量，但它在函数复杂性方面是活跃的。他们还表明，自动奥卡姆剃刀也适用于具有许多参数的 Bayesian 非参数模型，例如 Gaussian 过程。

如果我们专注于最大似然估计，存在许多阻止过拟合的模型选择启发式方法。它们被称为信息准则，我们选择具有最大值的模型。*Akaike 信息准则* (AIC) [(Akaike, 1974)]

$$\log p(\mathbf{x} | \theta) - M \quad (8.48)$$

通过添加惩罚项来补偿具有大量参数的更复杂模型的过拟合，从而纠正了最大似然估计器的偏差。这里， M 是模型参数的数量。AIC 估计给定模型丢失的相对信息。

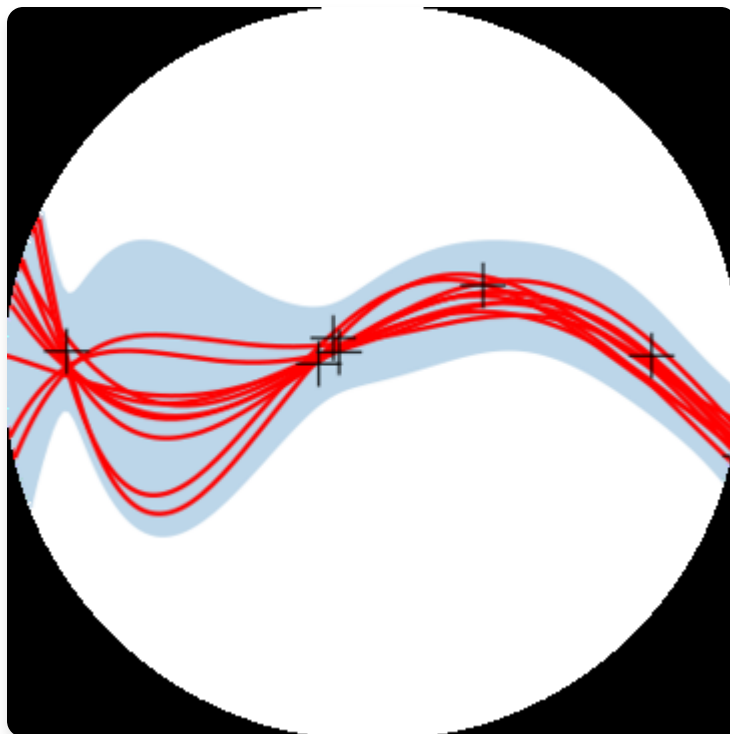
[贝叶斯] *贝叶斯信息准则* (BIC) [(Schwarz, 1978)]

$$[\text{准则}] \quad [(8.49) \log p(\mathbf{x}) = \log p(\mathbf{x} | \theta) p(\theta) d\theta \approx \log p(\mathbf{x} | \theta) - M \log N \quad [\text{信息}] \quad Z 1]$$

可用于指数族分布。这里， N 是数据点的数量， M 是参数的数量。BIC 比 AIC 更严厉地惩罚模型复杂性。

[《机器学习数学》草稿 (2023-10-18)。反馈：] <https://mml-book.com>

线性回归



在接下来的内容中，我们将应用第2、5、6和7章的数学概念来解决线性回归（曲线拟合）问题。在回归中，我们旨在找到一个函数 f ，将输入 $x \in \mathbb{R}$ 映射到相应的函数值 $f(x) \in \mathbb{R}$ 。我们假设给定了一组训练输入 $x[n]$ 和相应的噪声观测值 $y[n] = f(x[n]) + \epsilon$ ，其中 ϵ 是一个独立同分布的随机变量，描述了测量/观测噪声和可能未建模的过程（我们在本章中不会进一步考虑）。在整个章节中，我们假设零均值高斯噪声。我们的任务是找到一个不仅能建模训练数据，而且能很好地泛化到预测不属于训练数据的输入位置处的函数值的函数（见第8章）。这种回归问题的示例如图9.1所示。典型的回归设置如图9.1(a)所示：对于某些输入值 $x[n]$ ，我们观察到（噪声）函数值 $y[n] = f(x[n]) + \epsilon$ 。任务是推断生成数据的函数 f ，并能很好地泛化到新输入位置的函数值。可能的解决方案如图9.1(b)所示，我们还显示了以函数值 $f(x)$ 为中心的三个分布，它们代表数据中的噪声。

回归是机器学习中的一个基本问题，回归问题出现在各种研究领域和应用中

图9.1

0.4 0.4 [(a) 数据集;] [(b) 回归问题的可能解决方案] 0.2 0.2 y 0.0 y 0.0 -0.2 -0.2 -0.4 -0.4 -4 -2 0 2 4 -4 -2 0 2 4 x x

- a. 回归问题：从观察到的噪声函数值中，我们希望推断生成数据的潜在函数。
- b. 回归解决方案：可能生成数据的函数（蓝色），并指示了相应输入处函数值的测量噪声（橙色分布）。

[此材料由Cambridge University Press出版，作为Marc Peter Deisenroth、A. Aldo Faisal和Cheng Soon Ong (2020)的《机器学习数学》。此版本仅供个人使用免费查看和下载。不得重新分发、转售或用于衍生作品。] [©by M. P. Deisenroth, A. A. Faisal, and C. S. Ong, 2023.] <https://mml-book.com>

[线性回归]

中，包括时间序列分析（例如，系统辨识）、控制和机器人学（例如，强化学习、前向/逆向模型学习）、优化（例如，线搜索、全局优化）和深度学习应用（例如，计算机游戏、语音到文本翻译、图像识别、自动视频注释）。回归也是分类算法的关键组成部分。寻找回归函数需要解决各种问题，包括以下几个：

模型（类型）的选择和回归函数的参数化。 给定数据集，什么函数类（例如，多项式）是建模数据的好候选，我们应该选择什么特定的参数化（例如，多项式的次数）？如第8.6节所讨论的，模型选择允许我们比较各种模型，以找到能够合理解释训练数据的最简单模型。

寻找好的参数。 选择了回归函数的模型后，我们如何找到好的模型参数？在这里，我们需要查看不同的损失/目标函数（它们决定什么是“好的”拟合）和允许我们最小化这种损失的优化算法。

过拟合和模型选择。 当回归函数拟合训练数据“太好”但不能泛化到未见的测试数据时，过拟合就是一个问题。当潜在模型（或其参数化）过于灵活和表达力强时，通常会发生过拟合；见

8.6 节

我们将研究其深层原因，并讨论在线性回归背景下缓解过拟合(overfitting)效应的方法。

损失函数与参数先验的关系。 损失函数(优化目标)通常由概率模型激发和诱导。我们将研究损失函数与产生这些损失的潜在先验假设之间的联系。

不确定性建模。 在任何实践环境中，我们只能获得有限的、可能很大的(训练)数据量来选择模型类别和相应参数。由于这有限的训练数据并不能覆盖所有可能的场景，我们可能希望描述剩余的参数不确定性，以获得模型在测试时预测的置信度度量；训练集越小，不确定性建模就越重要。一致的不确定性建模为模型预测配备了置信区间。

接下来，我们将使用第3、5、6和7章的数学工具来解决线性回归问题。我们将讨论最大似然估计和最大后验(MAP)估计来寻找最优模型参数。使用这些参数估计，我们将简要了解泛化误差和过拟合。在本章末尾，我们将讨论贝叶斯线性回归，它允许我们在更高层次上推理模型参数，从而消除最大似然和MAP估计中遇到的一些问题。

9.1 问题表述

由于观测噪声的存在，我们将采用概率方法，并使用似然函数显式建模噪声。更具体地说，在本章中，我们考虑具有似然函数的回归问题

$$p(y|x) = N(y|f(x), \sigma^2) \quad (9.1)$$

这里， $x \in \mathbb{R}^D$ 是输入， $y \in \mathbb{R}$ 是噪声函数值(目标)。

根据(9.1)， x 和 y 之间的函数关系给出为

$$y = f(x) + \epsilon \quad (9.2)$$

其中 $\epsilon \sim N(0, \sigma^2)$ 是独立同分布(i.i.d.)高斯测量噪声，均值为0，方差为 σ^2 。我们的目标是找到一个接近(相似)于生成数据的未知函数 f 并且泛化良好的函数。

在本章中，我们专注于参数化模型，即我们选择一个参数化函数并找到”适用于”建模数据的参数 θ 。目前，我们假设噪声方差 σ^2 已知，并专注于学习模型参数 θ 。在线性回归中，我们考虑参数 θ 在模型中线性出现的特殊情况。线性回归的一个例子由以下给出

$$p(y|x, \theta) = N(y|x^\top \theta, \sigma^2) \quad (9.3) \Leftrightarrow y = x^\top \theta + \epsilon, \epsilon \sim N(0, \sigma^2) \quad (9.4)$$

其中 $\theta \in \mathbb{R}^D$ 是我们寻求的参数。(9.4)描述的函数类是通过原点的直线。在(9.4)中，我们选择了参数化 $f(x) = x^\top \theta$ 。

(9.3)中的似然是 y 在 $x^\top \theta$ 处评估的概率密度函数。注意，不确定性的唯一来源来自观测噪声(因为在(9.3)中假设 x 和 θ 已知)。没有观测噪声， x 和 y 之间的关系将是确定性的，(9.3)将是狄拉克 δ 函数。

例9.1

对于 $x, \theta \in \mathbb{R}$ ，(9.4)中的线性回归模型描述直线(线性函数)，参数 θ 是直线的斜率。图9.2(a)显示了不同 θ 值的一些示例函数。

(9.3)-(9.4)中的线性回归模型不仅在参数上是线性的，在输入 x 上也是线性的。图9.2(a)显示了此类函数的示例。我们稍后会看到，对于非线性变换 ϕ ， $y = \phi(x)^\top \theta$ 也是线性回归模型，因为”线性回归”



线性回归示例。(a)属于此类别的示例函数；(b)训练集；(c)最大似然估计。

(a)可以使用(9.4)中线性模型描述的示例函数(直线)。(b)训练集。(c)最大似然估计。

指的是”在参数上线性”的模型，即通过输入特征的线性组合描述函数的模型。这里，“特征”是输入 x 的表示 $\phi(x)$ 。

接下来，我们将更详细地讨论如何找到好的参数 θ 以及如何评估参数集是否“工作良好”。目前，我们假设噪声方差 σ^2 已知。

9.2 参数估计

考虑线性回归设置(9.4)，假设我们给定一个训练集 $D := \{(x_1, y_1), \dots, (x_N, y_N)\}$ ，由 N 个输入 $x_n \in \mathbb{R}^D$ 和相应的观测/目标 $y_n \in \mathbb{R}$ 组成， $n = 1, \dots, N$ 。



图9.3

对应的图形模型如图 [9.3] 所示。注意 $y[i]$ 和 $y[j]$ 在给定各自输入 $x[i], x[j]$ 的条件下是条件独立的，因此似然函数按如下方式分解：

$$p(Y | X, \theta) = p(y_1, \dots, y[N] | x_1, \dots, x[N], \theta) \quad (9.5a)$$

$$= \prod_{[n=1 \text{ to } N]} p(y[n] | x[n], \theta) = \prod_{[n=1 \text{ to } N]} N(y[n] | x[n] \top \theta, \sigma^2) \quad (9.5b)$$

其中我们定义 $X := \{x_1, \dots, x[N]\}$ 和 $Y := \{y_1, \dots, y[N]\}$ 分别为训练输入和对应目标值的集合。似然函数和因子 $p(y[n] | x[n], \theta)$ 由于噪声分布而呈高斯分布；参见 (9.3)。

在下文中，我们将讨论如何为线性回归模型 (9.4) 找到最优参数 $\theta^* \in \mathbb{R}^D$ 。一旦找到参数 θ ，我们可以通过在 (9.4) 中使用此参数估计来预测函数值，使得在任意测试输入 x 处，对应目标 y^* 的分布为

$$p(y^* | x, \theta) = N(y^* | x \top \theta, \sigma^2). \quad (9.6)$$

在下文中，我们将通过最大化似然函数来研究参数估计，这是我们在第 8.3 节已经在某种程度上涵盖的主题。

9.2 参数估计

9.2.1 最大似然估计

找到所需参数 θ_{ML} 的一种广泛使用的方法是最大似然估计，其中我们找到最大化似然函数 (9.5b) 的参数 θ_{ML} 。直观地说，最大化似然函数意味着最大化给定模型参数下训练数据的预测分布。我们获得最大似然参数为

$$\theta_{ML} \in \arg \max_{\theta} p(Y | X, \theta). \quad (9.7)$$

备注。似然函数 $p(y | x, \theta)$ 在 θ 中不是概率分布：它只是参数 θ 的函数，但不积分为 1（即，它是未归一化的），甚至可能在 θ 上不可积分。然而，(9.7) 中的似然函数在 y 中是归一化的概率分布。◇

为了找到最大化似然函数的所需参数 θ_{ML} ，我们通常执行梯度上升（或在负似然函数上进行梯度下降）。然而，在我们这里考虑的线性回归情况下，存在闭式解，这使得迭代梯度下降变得不必要。在实践中，我们不是直接最大化似然函数，而是对似然函数应用对数变换并最小化负对数似然函数。

备注（对数变换）。由于似然函数 (9.5b) 是 N 个高斯分布的乘积，对数变换是有用的，因为 (a) 它不会遭受数值下溢，(b) 微分规则将变得更简单。更具体地说，当我们乘以 N 个概率时，数值下溢将是一个问题，其中 N 是数据点的数量，因为我们无法表示非常小的数字，如 10^{-256} 。此外，对数变换将乘积转化为对数概率的和，使得相应的梯度是各个梯度的和，而不是重复应用乘积规则 (5.46) 来计算 N 项乘积的梯度。◇

为了找到线性回归问题的最优参数 θ_{ML} ，我们最小化负对数似然函数

$$-\log p(Y | X, \theta) = -\sum_{n=1}^N \log p(y[n] | x[n], \theta) = -\sum_{n=1}^N \log p(y[n] | x[n], \theta), \quad (9.8)$$

其中我们利用了似然函数 (9.5b) 由于我们对训练集的独立性假设而在数据点数量上分解。

在线性回归模型 (9.4) 中，似然函数是高斯的（由于高斯加性噪声项），使得我们得到

$$\log p(y[n] | x[n], \theta) = -1/(2\sigma^2)(y[n] - x[n]^T \theta)^2 + \text{const}, \quad (9.9)$$

其中常数包括所有与 θ 无关的项。在负对数似然函数 (9.8) 中使用 (9.9)，我们得到（忽略常数项）

$$L(\theta) := 1/(2\sigma^2) \sum_{n=1}^N (y[n] - x[n]^T \theta)^2 \quad (9.10a)$$

$$= 1/(2\sigma^2) (y - X\theta)^T (y - X\theta) = 1/(2\sigma^2) \|y - X\theta\|^2, \quad (9.10b)$$

其中我们定义设计矩阵 $X := [x_1^T, \dots, x_N^T]^T \in \mathbb{R}^{(N \times D)}$ 为训练输入的集合， $y := [y_1, \dots, y_N]^T \in \mathbb{R}^N$ 为收集所有训练目标的向量。注意设计矩阵 X 中的第 n 行对应于训练输入 $x[n]$ 。在 (9.10b) 中，我们使用了观测值 $y[n]$ 和相应模型预测 $x[n]^T \theta$ 之间的平方误差和等于 y 和 $X\theta$ 之间的平方距离这一事实。

通过 (9.10b)，我们现在有了需要优化的负对数似然函数的具体形式。我们立即看到 (9.10b) 是关于 θ 的二次函数

在 mini- θ 的情况下。这意味着如果我们选择点积作为内积，我们可以找到一个唯一的全局解 θ_{ML} ，通过最小化负对数似然 L 。我们可以通过计算 L 的梯度、将其设置为 0 并求解 θ 来找到全局最优解。

使用第5章的结果，我们计算L关于参数的梯度为

$$dL/d\theta = (1/2\sigma^2)d/d\theta(y - X\theta)^T(y - X\theta) \quad (9.11a)$$

$$= (1/2\sigma^2)d/d\theta(y^T y - 2y^T X\theta + \theta^T X^T X\theta) \quad (9.11b)$$

$$= (1/\sigma^2)(-y^T X + \theta^T X^T X) \in \mathbb{R}^{(1 \times D)} \quad (9.11c)$$

最大似然估计量 θ_{ML} 解决 $dL/d\theta = 0$ （必要最优性条件），忽略重复数据点的可能性，如果 $\text{rk}(X) = D$ ，即 $N \geq D$ 时，我们没有比数据点更多的参数，我们得到

$$dL/d\theta = 0 \iff \theta_{ML}^T X^T X = y^T X \quad (9.12a) \iff \theta_{ML} = y^T X (X^T X)^{-1} \quad (9.12b) \iff \theta_{ML} = (X^T X)^{-1} X^T y \quad (9.12c)$$

我们可以在第一个方程右乘 $(X^T X)^{-1}$ ，因为如果 $\text{rk}(X) = D$ ，其中 $\text{rk}(X)$ 表示X的秩，则 $X^T X$ 是正定的。

备注。 设置梯度为0是必要充分条件，我们得到全局最小值，因为Hessian $\nabla^2_{\theta} L(\theta) = X^T X \in \mathbb{R}^{(D \times D)}$ 是正定的。
◇

备注。 (9.12c)中的最大似然解要求我们求解形式为 $A\theta = b$ 的线性方程组，其中 $A = (X^T X)$ ， $b = X^T y$ 。◇

例9.2（直线拟合）

让我们看看图9.2，我们的目标是将直线 $f(x) = \theta x$ （其中 θ 是未知斜率）使用最大似然估计拟合到数据集上。该模型类（直线）中的函数示例如图9.2(a)所示。对于图9.2(b)所示的数据集，我们使用(9.12c)找到斜率参数 θ 的最大似然估计，并在图9.2(c)中得到最大似然线性函数。

带特征的最大似然估计

到目前为止，我们考虑了(9.4)中描述的线性回归设置，它允许我们使用最大似然估计将直线拟合到数据。然而，当涉及拟合更有趣的数据时，直线的表达能力不够。幸运的是，线性回归为我们提供了在线性回归框架内拟合非线性函数的方法：由于“线性回归”仅指“参数线性”，而输入可以进行任何非线性变换，我们可以对输入 x 执行任意非线性变换 $\varphi(x)$ ，然后线性组合该变换的组件。相应的线性回归模型是

$$p(y|x, \theta) = N(y|\varphi(x)^\top \theta, \sigma^2) \quad (9.13) \iff y = \varphi(x)^\top \theta + \varepsilon = \sum_{k=0}^{K-1} \theta_k \varphi_k(x) + \varepsilon,$$

其中 $\varphi: \mathcal{R}^D \rightarrow \mathcal{R}^K$ 是输入 x 的（非线性）变换， $\varphi_k: \mathcal{R}^D \rightarrow \mathcal{R}$ 是特征向量 φ 的第 k 个分量。注意模型参数 θ 仍然只是线性出现。

例9.3（多项式回归）

我们关注回归问题 $y = \varphi(x)^\top \theta + \varepsilon$ ，其中 $x \in \mathcal{R}$ ， $\theta \in \mathcal{R}^K$ 。在此情况下经常使用的变换是

$$\varphi(x) = [1, x, x^2, \dots, x^{(K-1)}]^\top \in \mathcal{R}^K \quad (9.14)$$

这意味着我们将原始一维输入空间“提升”到由所有单项式 x^k （ $k = 0, \dots, K-1$ ）组成的 K 维特征空间。通过这些特征，我们可以在线性回归框架内建模次数 $\leq K-1$ 的多项式：次数 $K-1$ 的多项式是

$$f(x) = \sum_{k=0}^{K-1} \theta_k x^k = \varphi(x)^\top \theta \quad (9.15)$$

其中 φ 在(9.14)中定义， $\theta = [\theta_0, \dots, \theta_{(K-1)}]^\top \in \mathcal{R}^K$ 包含（线性）参数 θ_k 。

现在让我们看看线性回归模型(9.13)中参数 θ 的最大似然估计。我们考虑训练输入 $x_n \in \mathcal{R}^D$ 和目标 $y_n \in \mathcal{R}$ ， $n = 1, \dots, N$ ，并定义特征矩阵（设计矩阵）为

$$\Phi := [\varphi(x_1)^\top; \varphi(x_2)^\top; \dots; \varphi(x_N)^\top] = [\varphi_0(x_1) \ \varphi_1(x_1) \ \dots \ \varphi_{(K-1)}(x_1); \varphi_0(x_2) \ \varphi_1(x_2) \ \dots \ \varphi_{(K-1)}(x_2); \dots; \varphi_0(x_N) \ \varphi_1(x_N) \ \dots \ \varphi_{(K-1)}(x_N)] \in \mathcal{R}^{(N \times K)} \quad (9.16)$$

其中 $\Phi_{ij} = \varphi_j(x_i)$ ， $\varphi_j: \mathcal{R}^D \rightarrow \mathcal{R}$ 。

例9.4（二阶多项式的特征矩阵）

对于二阶多项式和 N 个训练点 $x_n \in \mathcal{R}$ ， $n = 1, \dots, N$ ，特征矩阵是

$$\Phi = [1 \ x_1 \ x_1^2; 1 \ x_2 \ x_2^2; \dots; 1 \ x_N \ x_N^2] \quad (9.17)$$

使用(9.16)中定义的特征矩阵 Φ ，线性回归模型(9.13)的负对数似然可以写成

$$-\log p(Y|X, \theta) = (1/2\sigma^2)(y - \Phi\theta)^\top (y - \Phi\theta) + \text{const} \quad (9.18)$$

将(9.18)与“无特征”模型的负对数似然(9.10b)比较，我们立即看到只需要用 Φ 替换 X 。由于 X 和 Φ 都独立于我们希望优化的参数 θ ，我们立即得到最大似然估计

$$\theta_{ML} = (\Phi^T \Phi)^{-1} \Phi^T y \quad (9.19)$$

针对在 (9.13) 中定义的具有非线性特征的线性回归问题。

备注。 当我们不使用特征时，我们要求 $X^T X$ 是可逆的，这在 $\text{rk}(X) = D$ 时成立，即 X 的列线性无关。在 (9.19) 中，我们因此要求 $\Phi^T \Phi \in \mathbb{R}^{(K \times K)}$ 是可逆的。当且仅当 $\text{rk}(\Phi) = K$ 时成立。◇

示例 9.5 (最大似然多项式拟合)

图 9.4



图 9.4: 多项式回归

图 9.4: 多项式回归

考虑图 9.4(a) 中的数据集。该数据集由 $N = 10$ 对 (x_n, y_n) 组成，其中 $x_n \sim U[-5, 5]$ 且 $y_n = -\sin(x_n/5) + \cos(x_n) + \epsilon$ ，其中 $\epsilon \sim N(0, 0.2^2)$ 。

我们使用最大似然估计拟合 4 次多项式，即参数 θ_{ML} 在 (9.19) 中给出。最大似然估计在任何测试位置 x^* 处产生函数值 $\phi(x^*)^T \theta_{ML}$ 。结果如图 9.4(b) 所示。

估计噪声方差

到目前为止，我们假设噪声方差 σ^2 是已知的。然而，我们也可以使用最大似然估计原理来获得噪声方差的最大似然估计器 σ^2_{ML} 。为此，我们遵循标准程序：写出对数似然，计算其对 $\sigma^2 > 0$ 的导数，设为 0，然后求解。对数似然为

$$\log p(Y|X, \theta, \sigma^2) = \log \prod_{n=1}^N N(y_n | \phi(x_n)^T \theta, \sigma^2) \quad (9.20a)$$

$$= \sum_{n=1}^N [-1/2 \log(2\pi) - 1/2 \log \sigma^2 - 1/(2\sigma^2)(y_n - \phi(x_n)^T \theta)^2] \quad (9.20b)$$

$$= -N/2 \log \sigma^2 - 1/(2\sigma^2) \sum_{n=1}^N (y_n - \phi(x_n)^T \theta)^2 + \text{const.} \quad (9.20c) \quad \text{=: } s$$

对数似然对 σ^2 的偏导数为

$$\partial/\partial \sigma^2 \log p(Y|X, \theta, \sigma^2) = -N/(2\sigma^2) + s/(2\sigma^4) = 0 \quad (9.21a)$$

$$\Leftrightarrow N/(2\sigma^2) = s/(2\sigma^4) \quad (9.21b)$$

因此我们识别出

$$\sigma^2_{ML} = s/N = 1/N \sum_{n=1}^N (y_n - \phi(x_n)^T \theta)^2 \quad (9.22)$$

因此，噪声方差的最大似然估计是无噪声函数值 $\phi(x_n)^T \theta$ 与输入位置 x_n 处对应的有噪声观测 y_n 之间平方距离的经验均值。

9.2.2 线性回归中的过拟合

我们刚刚讨论了如何使用最大似然估计将线性模型（例如多项式）拟合到数据。我们可以通过计算产生的误差/损失来评估模型的质量。一种方法是计算负对数似然 (9.10b)，我们最小化它来确定最大似然估计器。或者，鉴于噪声参数 σ^2 不是自由模型参数，我们可以忽略 $1/\sigma^2$ 的缩放，这样我们最终得到平方误差损失函数 $\|y - \Phi\theta\|^2$ 。我们通常使用**均方根误差 (RMSE)** 而不是使用这种平方损失

$$\text{RMSE} = 1/\sqrt{N} \|y - \Phi\theta\| = \sqrt{(1/N \sum_{n=1}^N (y_n - \phi(x_n)\top\theta)^2)} \quad (9.23)$$

它 (a) 允许我们比较不同大小数据集的误差，(b) 与观测函数值 y_n 具有相同的尺度和相同的单位。例如，如果我们拟合一个将邮政编码 (x 以纬度和经度给出) 映射到房价 (y 值为欧元) 的模型，那么 RMSE 也以欧元为单位，而平方误差以欧元² 为单位。如果我们选择包含原始负对数似然 (9.10b) 中的因子 σ^2 ，那么我们最终得到一个无单位的目标，即在前面的例子中，我们的目标不再是欧元或欧元²。

对于模型选择（见第 8.6 节），我们可以使用 RMSE（或负对数似然）通过找到最小化目标的多项式次数 M 来确定多项式的最佳次数。鉴于多项式次数是自然数，我们可以执行暴力搜索并枚举 M 的所有（合理）值。对于大小为 N 的训练集，测试 $0 \leq M \leq N-1$ 就足够了。对于 $M < N$ ，最大似然估计器是唯一的。对于 $M \geq N$ ，我们有更多参数

图 9.5 不同多项式次数 M 的最大似然拟合。



图 9.5：不同多项式次数的最大似然拟合

图 9.5：不同多项式次数的最大似然拟合
比数据点多，需要解决一个欠定系统

线性方程组 ($\top\Phi\Phi$ 在 [(9.19)] 中也不再可逆)，因此存在无限多个可能的最大似然估计器。

图 [9.5] 显示了使用最大似然法为来自图 [9.4(a)] 的数据集 ($N = 10$ 个观测值) 确定的多项式拟合。我们注意到低次多项式（例如，常数项 ($M = 0$) 或线性项 ($M = 1$)）对数据拟合较差，因此是真实潜在函数的较差表示。对于 $M = 3, \dots, 6$ 次，拟合看起来合理并平滑地插值数据。当我们转向更高次多项式时，我们注意到它们越来越较好地拟合数据。在 $M = N - 1 = 9$ 的极端情况下，函数将通过每个数据点。然而，这些高次多项式剧烈振荡，是生成数据的潜在函数的较差表示，因此我们遭受**过拟合**。

记住目标是通过对新（未见过的）数据进行准确预测来达到良好的泛化。我们通过考虑包含 200 个数据点的独立测试集，获得对 M 次多项式泛化性能依赖性的一些定量洞察，该测试集使用与生成训练集完全相同的程序生成。作为测试输入，我们选择了区间 $[-5, 5]$ 内 200 个点的线性网格。对于 M 的每个选择，我们评估训练数据和测试数据的 RMSE [(9.23)]。注意噪声方差 $\sigma^2 > 0$ 。

现在看测试误差，这是相应多项式泛化特性的定性衡量，我们注意到最初测试误差降低；见图 [9.6]（橙色）。对于四阶多项式，测试误差相对较低，并在 5 次以下保持相对恒定。然而，从 6 次开始，测试误差显著增加，高阶多项式具有非常差的泛化特性。在这个特定例子中，这也从图 [9.5] 中相应的最大似然拟合中明显看出。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[300] 线性回归

图 9.6 训练和测试误差。训练误差（蓝色曲线）在多项式次数增加时从不增加。测试误差（橙色曲线）在度数 6 之后显著增加。



图9.6：训练和测试误差图表

图9.6：训练和测试误差图表

注意**训练误差**（图 [9.6] 中的蓝色曲线）在多项式次数增加时从不增加。在我们的例子中，最佳泛化（最小**测试误差**的点）是通过 $M = 4$ 次多项式获得的。

9.2.3 最大后验估计

我们刚刚看到最大似然估计容易产生过拟合。我们经常观察到，如果我们遇到过拟合，参数值的大小会变得相对较大 [(Bishop, 2006)]。

为了减轻巨大参数值的影响，我们可以在参数上放置先验分布 $p(\theta)$ 。先验分布明确编码哪些参数值是合理的（在看到任何数据之前）。例如，单个参数 θ 上的高斯先验 $p(\theta) = N(0, 1)$ 编码参数值预期位于区间 $[-2, 2]$ （围绕均值的两个标准差）。一旦数据集 X, Y 可用，我们不是最大化似然性，而是寻求最大化后验分布 $p(\theta | X, Y)$ 的参数。这个程序称为**最大后验 (MAP)** 估计。

给定训练数据 X, Y 的参数 θ 上的后验通过应用贝叶斯定理（第 [6.3] 节）获得：

$$p(\theta | X, Y) = p(Y | X, \theta)p(\theta) / p(Y | X) \quad (9.24)$$

由于后验明确依赖于参数先验 $p(\theta)$ ，先验将对我们找到的作为后验最大化器的参数向量产生影响。我们将在下面更明确地看到这一点。最大化后验 [(9.24)] 的参数向量 θ^{MAP} 是 MAP 估计。

为了找到 MAP 估计，我们遵循与最大似然估计类似的步骤。我们从对数变换开始，计算对数后验：

$$\log p(\theta | X, Y) = \log p(Y | X, \theta) + \log p(\theta) + \text{const} \quad (9.25)$$

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>] **9.2 参数估计** [301]

其中常数包含与 θ 无关的项。我们看到 [(9.25)] 中的对数后验是对数似然 $p(Y | X, \theta)$ 和对数先验 $\log p(\theta)$ 的和，因此 MAP 估计将是先验（我们在观察数据之前对合理参数值的建议）和数据依赖似然之间的“折中”。

为了找到 MAP 估计 θ^{MAP} ，我们最小化关于 θ 的负对数后验分布，即我们求解：

$$\theta^{\text{MAP}} \in \arg \min \{-\log p(Y | X, \theta) - \log p(\theta)\} \quad (9.26)$$

关于 θ 的负对数后验的梯度是：

$$-d \log p(\theta | X, Y) / d\theta = -d \log p(Y | X, \theta) / d\theta - d \log p(\theta) / d\theta \quad (9.27)$$

其中我们将右侧的第一项识别为来自 [(9.11c)] 的负对数似然的梯度。

对于参数 θ 上的（共轭）高斯先验 $p(\theta) = N(0, b^2 I)$ ，线性回归设置 [(9.13)] 的负对数后验，我们获得负对数后验：

$$-\log p(\theta | X, Y) = 1/(2\sigma^2)(y - \Phi\theta)^T(y - \Phi\theta) + 1/(2b^2)\theta^T\theta + \text{const} \quad (9.28)$$

这里，第一项对应来自对数似然的贡献，第二项源于对数先验。对数后验关于参数 θ 的梯度为

$$-d \log p(\theta | X, Y) / d\theta = (1/\sigma^2)(\Phi^T\Phi - y^T\Phi) + \theta / b^2 \quad (9.29)$$

我们将通过将该梯度设为 0 并求解 θ^{MAP} 来找到 MAP 估计 θ^{MAP} 。我们得到

$$\sigma^{-1} [\tau]^{-1} [\tau]^{-1} (\theta^T \Phi - y^T \Phi) + \theta = 0 \quad (9.30a) \quad [2] \quad b [2]$$

$$1 \quad 1 \quad 1$$

$$\Leftrightarrow [\tau] [\tau] [\tau] \theta^T \Phi - \Phi + I y^T \Phi = 0 \quad (9.30b) \quad -$$

$$\sigma [2] \quad b [2] \quad \sigma [2]$$

$$[2] \quad \sigma$$

$$\Leftrightarrow [\tau] [\tau] [\tau] \theta^T \Phi - \Phi + I = y^T \Phi \quad (9.30c)$$

$$b [2]$$

$$1 \quad [2] \quad [-] \quad \sigma$$

$$\Leftrightarrow [\tau] [\tau] [\tau] \theta = y^T \Phi - \Phi + I \quad (9.30d)$$

$$b [2]$$

因此 MAP 估计为（通过转置最后等式的两边） $[\tau] [\Phi] [\Phi]$ 是对称的，]

$$[2] \quad 1 \quad [\tau] [\tau] \quad [\text{正定的。}] \quad \sigma \quad [-] \quad [[\text{正半}]]$$

$$\theta [\text{MAP}] = \Phi^T \Phi + I \quad \Phi^T y. \quad (9.31) \quad [2] \quad [\text{额外项}] \quad b$$

[在 (9.31)] 中是严格]

将 (9.31) 中的 MAP 估计与 (9.19) 中的最大似然估计进行比较，[[正定的，因此]]

我们看到两个解之间的唯一区别是逆矩阵中的额外项 $[\sigma] [2] \quad [[\text{逆矩阵}]]$

$[b] [2] \quad I$ 。这一项确保了 [[存在。]]

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[302] [[*Linear Regression*]]

$\Phi^T [\tau] [2] \quad [\sigma] \quad \Phi + I$ 是对称且严格正定的（即，其逆

$[b] [2]$

存在且 MAP 估计是线性方程组的唯一解）。此外，它反映了正则化器的影响。

示例 9.6（多项式回归的 MAP 估计）

在第 [9.2.1] 节的多项式回归示例中，我们对参数 θ 设置高斯先验 $p(\theta) = \mathcal{N}(0, I)$ 并根据 (9.31) 确定 MAP 估计。在图 [9.7] 中，我们展示了 6 次（左）和 8 次（右）多项式的最大似然和 MAP 估计。对于低次多项式，先验（正则化器）并不起重要作用，但对于高次多项式，它保持函数相对平滑。尽管 MAP 估计可以推进过拟合的边界，但它并不是解决这个问题的通用方案，所以我们需要更有原则的方法来处理过拟合。

图 9.7

[多项式] [4] [4] [训练数据] [回归:] [MLE] [最大似然] [2] [2] [MAP] [和 MAP 估计。]

[(a) 6 次多项式] [y] [0] [y] [0] [;]

[(b) 8 次多项式] [-] [训练数据] [2] [-][2]

[MLE]

[。] [MAP] [-] [4] [-][4]

[-][4] [-][2] [0] [2] [4] [-][4] [-][2] [0] [2] [4]

x x

[(a) 6 次多项式。] [(b) 8 次多项式。]

9.2.4 MAP 估计作为正则化

除了对参数 θ 设置先验分布外，还可以通过正则化手段惩罚参数的幅度来减轻过拟合的影响。在正则化最小二乘中，我们考虑损失函数

[regularization] [regularized least squares]

$$\|y - \Phi\theta\|^2 + \lambda \|\theta\|^2, \quad (9.32) \quad [2] \quad [2]$$

我们关于 θ 最小化该函数（见第 [8.2.3] 节）。这里，第一项是数据拟合项（也称为误差项），它与负对数似然成正比；见 [(9.10b)]。第二项称为正则化器，正则化参数 $\lambda \geq 0$ 控制正则化的“严格程度”。

[data-fit term] [misfit term] [regularizer] [regularization parameter]

备注。除了欧几里得范数 $\|\cdot\|_2$ ，我们可以在 [(9.32)] 中选择任何 p -范数 $\|\cdot\|_p$ 。在实践中， p 的较小值会导致更稀疏的解。这里，“稀疏”意味着许多参数值 $\theta[d] = 0$ ，这对于变量选择也很有用。对于 $p = 1$ ，正则化器称为 LASSO（最小绝对收缩和选择算子），由 [Tibshirani (1996)] 提出。◇

[(9.32)] 中的正则化器 $\lambda \|\theta\|_2^2$ 可以解释为负对数高斯先验，我们在 MAP 估计中使用；见 [(9.26)]。更具体地，使用高斯先验 $p(\theta) = \mathcal{N}(\theta | 0, bI)$ ，我们得到负对数高斯先验

$$-\log p(\theta) = \frac{1}{2b} \|\theta\|_2^2 + \text{const} \quad (9.33)$$

b

因此对于 $\lambda = 1/b$ ，正则化项和负对数高斯先验是相同的。

[2][b][2]

给定 [(9.32)] 中的正则化最小二乘损失函数由与负对数似然加负对数先验密切相关的项组成，当我们最小化该损失时，我们得到与 [(9.31)] 中 MAP 估计非常相似的解也就不奇怪了。更具体地，最小化正则化最小二乘损失函数产生

$$\theta_{\text{MAP}} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T y, \quad (9.34)$$

[RLS]

这与 [(9.31)] 中的 MAP 估计在 $[\sigma^2] = \lambda$ 时是相同的，其中 σ^2 是

[b][2]

噪声方差， $[2]b$ 是（各向同性）高斯先验 $p(\theta) = \mathcal{N}(\theta | 0, bI)$ 的方差。

[点估计是一个] [单一特定的]

到目前为止，我们已经涵盖了使用最大似然和 MAP 估计的参数估计，其中我们找到了优化目标函数（似然或后验）的点估计 θ 。我们看到最大似然和 MAP 估计都可能导致过拟合。在下一节中，我们将讨论贝叶斯线

性回归，其中我们使用贝叶斯推断（第 [8.4] 节）来找到未知参数的后验分布，随后我们用它来进行预测。更具体地，对于预测，我们将对所有合理的参数集合进行平均，而不是专注于点估计。

[[不像分布]] [[在合理的]] [[参数设置上。]]

9.3 贝叶斯线性回归

之前，我们研究了线性回归模型，通过最大似然估计或MAP估计等方法来估计模型参数 θ 。我们发现MLE可能导致严重的过拟合，特别是在小数据情况下。MAP通过在参数上放置先验来解决这个问题，先验起到正则化器的作用。

贝叶斯线性回归将参数先验的思想推进了一步，甚至不尝试计算参数的点估计，而是在进行预测时考虑参数的完整后验分布。这意味着我们不拟合任何参数，而是计算所有合理参数设置的均值（根据后验）。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[304] 线性回归

9.3.1 模型

在贝叶斯线性回归中，我们考虑模型

先验 $p(\theta) = N(\mathbf{m}_0, S_0)$, (9.35)

似然 $p(y|x, \theta) = N(y|\phi(x)^T\theta, \sigma^2)$,

图9.8 其中我们现在明确地在 θ 上放置高斯先验 $p(\theta) = N(\mathbf{m}_0, S_0)$,

贝叶斯线性回归的图模型。这将参数向量转换为随机变量。这使我们能够写出图9.8中相应的图模型，其中我们明确了 θ 上高斯先验的参数。完整的概率模型，即观测和未观测随机变量 y 和 θ 的联合分布，分别为

$p(y, \theta|x) = p(y|x, \theta)p(\theta)$. (9.36)

9.3.2 先验预测

在实践中，我们通常对参数值 θ 本身不那么感兴趣。相反，我们的重点往往在于用这些参数值进行预测。在贝叶斯设置中，我们采用参数分布，并在进行预测时对所有合理的参数设置求平均。更具体地说，要在输入 x^* 处进行预测，我们积分掉 θ 并得到

$$p(y/x) = \int p(y/x, \theta)p(\theta)d\theta = E_{\theta}[p(y/x, \theta)], (9.37)$$

我们可以将其解释为对于根据先验分布 $p(\theta)$ 的所有合理参数 θ ， y/x ， θ 的平均预测。请注意，使用先验分布的预测只需要我们指定输入 x^* ，而不需要训练数据。

在我们的模型(9.35)中，我们在 θ 上选择了共轭（高斯）先验，使得预测分布也是高斯的（并且可以以闭式形式计算）：使用先验分布 $p(\theta) = N(m_0, S_0)$ ，我们得到预测分布为

$$p(y/x) = N(\phi(x)^T m_0, \phi(x)^T S_0 \phi(x) + \sigma^2), (9.38)$$

其中我们利用了（i）由于共轭性（见第6.6节）和高斯的边际化性质（见第6.5节），预测是高斯的，（ii）高斯噪声是独立的，因此

$$V[y^*] = V_{\theta}[\phi(x^*)^T \theta] + V_{\epsilon}[\epsilon], (9.39)$$

以及（iii） y 是 θ 的线性变换，因此我们可以分别使用(6.50)和(6.51)分析地应用计算预测均值和协方差的规则。在(9.38)中，预测方差中的项 $\phi(x)^T S_0 \phi(x)$ 明确地考虑了与参数 θ 相关的不确定性，而 σ^2 是由于测量噪声造成的不确定性贡献。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>. 9.3 贝叶斯线性回归 305]

如果我们有兴趣预测无噪声函数值 $f(x) = \phi(x)^T \theta$ 而不是噪声损坏的目标 y^* ，我们得到

$$p(f(x)) = N(\phi(x)^T m_0, \phi(x)^T S_0 \phi(x)), (9.40)$$

这与(9.38)仅在预测方差中省略噪声方差 σ^2 方面不同。

注释（函数上的分布） 由于我们可以使用一组样本来表示参数分布 $p(\theta)$ ，并且每个样本 θ_i 产生一个函数 $f_i(\cdot) = \theta_i^T \phi(\cdot)$ ，因此参数分布 $p(\theta)$ 诱导出函数上的分布 $p(f(\cdot))$ 。这里我们使用符号 $(\cdot)$ 来明确表示函数关系。◇

例9.7（函数上的先验）

图9.9 函数上的先验。

(a) 函数上的分布由均值函数（黑线）和边际不确定性（阴影）表示，分别代表67%和95%的置信界限；(b) 来自先验分布的样本，这些样本由参数先验的样本诱导。

让我们考虑一个5次多项式的贝叶斯线性回归问题。我们选择参数先验 $p(\theta) = N(0, 4I)$ 。图9.9可视化了由这个参数先验诱导的函数上的诱导先验分布（阴影区域：深灰色：67%置信界限；浅灰色：95%置信界限），包括来自这个先验的一些函数样本。

函数样本通过首先从 $p(\theta)$ 采样参数向量 θ_i ，然后计算 $f_i(\cdot) = \theta_i^T \phi(\cdot)$ 得到。我们使用了200个输入位置 $x^* \in [-5, 5]$ ，对其应用特征函数 $\phi(\cdot)$ 。图9.9中的不确定性（由阴影区域表示）完全由于参数不确定性，因为我们考虑了无噪声预测分布(9.40)。

到目前为止，我们已经了解了如何使用参数先验 $p(\theta)$ 来计算预测。然而，当我们有参数后验分布(给定一些训练数据 X, Y)时，预测和推断的相同原理仍然适用，如[(9.37)]所示——我们只需要用后验分布 $p(\theta | X, Y)$ 替换先验分布 $p(\theta)$ 。在下面的内容中，我们将详细推导后验分布，然后使用它来进行预测。

9.3.3 Posterior Distribution

给定训练集输入 $x[n] \in \mathbb{R}^D$ 和对应的观测值 $y[n] \in \mathbb{R}$ ， $n = 1, \dots, N$ ，我们使用贝叶斯定理计算参数的后验分布：

$$p(\theta | X, Y) = p(Y | X, \theta)p(\theta) / p(Y | X), \quad (9.41)$$

其中 X 是训练输入的集合， Y 是对应训练目标的集合。此外， $p(Y | X, \theta)$ 是likelihood， $p(\theta)$ 是参数先验，而

$$p(Y | X) = \int p(Y | X, \theta)p(\theta)d\theta = E[\theta][p(Y | X, \theta)] \quad (9.42)$$

是marginal likelihood/evidence，它独立于参数 θ 并确保后验分布被归一化，即积分为1。我们可以将marginal likelihood视为在所有可能参数设置上(相对于先验分布 $p(\theta)$)平均的likelihood。

定理9.1 (Parameter Posterior)。在我们的模型(9.35)中，参数后验分布(9.41)可以以闭合形式计算为

$$p(\theta | X, Y) = N(\theta | m_N, S_N), \quad (9.43a)$$

$$S_N^{-1} = (S_0^{-1} + \sigma^{(-2)\Phi} \mathbf{T} \Phi), \quad (9.43b)$$

$$m_N = S_N(S_0^{-1}m_0 + \sigma^{(-2)\Phi} \mathbf{T} y), \quad (9.43c)$$

其中下标 N 表示训练集的大小。

证明 贝叶斯定理告诉我们后验分布 $p(\theta | X, Y)$ 与likelihood $p(Y | X, \theta)$ 和先验 $p(\theta)$ 的乘积成正比：

$$p(\theta | X, Y) = p(Y | X, \theta)p(\theta) / p(Y | X) \quad (9.44a)$$

$$\text{Likelihood } p(Y | X, \theta) = N(y | \Phi \theta, \sigma^2 I) \quad (9.44b)$$

$$\text{Prior } p(\theta) = N(\theta | m_0, S_0). \quad (9.44c)$$

我们可以将问题转换到对数空间，通过配方(completing the squares)来求解后验分布的均值和协方差，而不是直接处理先验和likelihood的乘积。

对数先验和对数likelihood的和为

$$\log N(y | \Phi \theta, \sigma^2 I) + \log N(\theta | m_0, S_0) \quad (9.45a)$$

$$= -1/2 \sigma^{(-2)}(y - \Phi \theta)^T (y - \Phi \theta) + (\theta - m_0)^T S_0^{-1} (\theta - m_0) + \text{const} \quad (9.45b)$$

其中常数项包含与 θ 无关的项。在接下来的推导中我们将忽略常数项。现在我们对(9.45b)进行因式分解，得到

$$-1/2 \sigma^{(-2)y} \top y - 2\sigma^{(-2)y} \top \Phi \theta + \theta^{\top \sigma(-2)} \Phi^{\top} \Phi \theta + \theta^{\top S_{-0}}(-1) \theta - 2m_{-0}^{\top S_{-0}}(-1) \theta + m_{-0}^{\top S_{-0}}(-1) m_{-0} \quad (9.46a)$$

$$= -1/2 \theta^{\top \sigma(-2)} \Phi^{\top} \Phi \theta + S_{-0}^{\wedge(-1)} \theta - 2(\sigma^{(-2)\Phi} \top y + S_{-0}^{(-1)m_{-0}} \top \theta) + \text{const}, \quad (9.46b)$$

其中常数项包含(9.46a)中与 θ 无关的黑色项。橙色项是 θ 的线性项，蓝色项是 θ 的二次项。检查(9.46b)，我们发现这个方程关于 θ 是二次的。未归一化对数后验分布是(负)二次形式这一事实意味着后验分布是Gaussian，即

$$p(\theta \mid X, Y) = \exp(\log p(\theta \mid X, Y)) \propto \exp(\log p(Y \mid X, \theta) + \log p(\theta)) \quad (9.47a)$$

$$\propto \exp\{-1/2 \theta^{\top \sigma(-2)} \Phi^{\top} \Phi \theta + S_{-0}^{\wedge(-1)} \theta - 2(\sigma^{(-2)\Phi} \top y + S_{-0}^{(-1)m_{-0}} \top \theta)\}, \quad (9.47b)$$

其中我们在最后的表达式中使用了(9.46b)。

剩下的任务是将这个(未归一化的)Gaussian带入与 $N(\theta \mid m_{-N}, S_{-N})$ 成正比的形式，即我们需要识别均值 m_{-N} 和协方差矩阵 S_{-N} 。为此，我们使用配方的概念。期望的对数后验为

$$\log N(\theta \mid m_{-N}, S_{-N}) = -1/2 (\theta - m_{-N})^{\top S_{-N}}(-1)(\theta - m_{-N}) + \text{const} \quad (9.48a)$$

$$= -1/2 \theta^{\top S_{-N}}(-1) \theta - 2m_{-N}^{\top S_{-N}}(-1) \theta + m_{-N}^{\top S_{-N}}(-1) m_{-N}. \quad (9.48b)$$

在这里，我们将二次形式 $(\theta - m_{-N})^{\top S_{-N}}(-1)(\theta - m_{-N})$ 因式分解为仅关于 θ 的二次项(蓝色)、关于 θ 的线性项(橙色)和常数项(黑色)。这现在允许我们通过匹配(9.46b)和(9.48b)中的彩色表达式来找到 S_{-N} 和 m_{-N} ，得到

$$S_{-N}^{\wedge(-1)} = \Phi^{\top \sigma(-2)} I \Phi + S_{-0}^{\wedge(-1)} \quad (9.49a)$$

$$\Leftrightarrow S_{-N} = (\sigma^{(-2)\Phi} \top \Phi + S_{-0}^{(-1)})(-1) \quad (9.49b)$$

和

$$m_{-N}^{\top S_{-N}}(-1) = (\sigma^{(-2)\Phi} \top y + S_{-0}^{(-1)m_{-0}} \top) \quad (9.50a)$$

$$\Leftrightarrow m_{-N} = S_{-N}(\sigma^{(-2)\Phi} \top y + S_{-0}^{\wedge(-1)} m_{-0}). \quad (9.50b)$$

备注 (配方的一般方法)。如果我们给定一个方程

$$x^{\top} A x - 2a^{\top} x + \text{const}_1, \quad (9.51)$$

其中 A 是对称正定的，我们希望将其转换为形式

$$(x - \mu)^{\top \Sigma}(-1)(x - \mu) + \text{const}_2, \quad (9.52)$$

我们可以通过设置

$$\Sigma^{\wedge(-1)} := A, \quad (9.53) \quad \mu := \Sigma a \quad (9.54)$$

和 $\text{const}_2 = \text{const}_1 - \mu^{\top \Sigma}(-1) \mu$ 来实现。◇

我们可以看到(9.47b)中指数项内的项具有(9.51)的形式，其中

$$\Lambda := \sigma^{(-2)\Phi} \top \Phi + S_0^{-1}, \quad (9.55) \quad \alpha := \sigma^{(-2)\Phi} \top y + S_0^{-1} m_0. \quad (9.56)$$

由于在像(9.46a)这样的方程中识别 Λ, α 可能很困难,

将这些方程转换为形式[(9.51)]是有帮助的, 该形式将

二次项、线性项和常数项解耦, 这简化了寻找

期望解的过程。

9.3.4 后验预测

在[(9.37)]中, 我们使用参数先验 $p(\theta)$ 计算了测试输入

$x[*]$ 处 $y[*]$ 的预测分布。原则上, 使用参

数后验 $p(\theta | X, Y)$ 进行预测在本质上并没有根本不同, 因为

在我们的共轭模型中, 先验和后验都是高斯分布 (具有

不同的参数)。因此, 通过遵循[9.3.2]节中的相同推理, 我们得到 (后验) 预测分布

Z

$$p(y[*] | X, Y, x[*]) = p(y[*] | x[*], \theta) p(\theta | X, Y) d\theta \quad (9.57a)$$

Z

$$= [\top] [2] N y | \phi(x) \theta, \sigma^2 N \theta | m, S d\theta \quad (9.57b) \quad [*] [*] [N] [N]$$

$$= [\top] [\top] [2] N y | \phi(x) m, \phi(x) S \phi(x) + \sigma^2. \quad (9.57c) \quad [*] [*] [N] [*] [N] [*]$$

$E[\phi(x) S \phi(x)]$ 反映了与

$[*] [N] [*]$

$[\phi] [\top] ([*] D) [m] [=]$ 参数 θ 相关的后验不确定性。注意 S 依赖于训练输入

$[N] [N]$

$[\phi] [\top] ([*] D) [\theta] [\top] [MAP] [.]$ 通过 Φ ; 见[(9.43b)]。预测均值 $\phi(x) m$ $[*] [N]$ 与

使用MAP估计 θ $[MAP]$ 做出的预测一致。

[《机器学习数学》草稿 (2023-10-18)。反馈:] [\[https://mml-book.com\]](https://mml-book.com) [.] [9.3 贝叶斯线性回归] [309]

注释 (边际似然和后验预测分布) .通过

替换[(9.57a)]中的积分, 预测分布可以等价地写为期望 $E[\theta | X] [.] [Y] [p(y[*] | x[*], \theta)]$, 其中期望

是关于参数后验 $p(\theta | X, Y)$ 计算的。

以这种方式写出后验预测分布突出了与

边际似然[(9.42)]的密切相似性。边际似然和后验预测分布之间的关键区别是 (i) 边际似然可以被认为是预测训练目标 y 而不是测试目标 y_* ，以及 (ii) 边际似然关于参数先验而不是参数后验进行平均。◇

注释（无噪声函数值的均值和方差）.在许多情况下，我们对（有噪声的）观测 y_* 的预测分布 $p(y_* | X, Y, x_*)$ 不感兴趣。相反，我们希望得到

（无噪声的）函数值 $f(x) = \phi(x)\theta$ 的分布。我们通过利用均值和

$[*][*]$

方差的性质来确定相应的矩，得到

$$E[\tau][\tau][f(x) | X, Y] = E[\phi(x)\theta | X, Y] = \phi(x)E[\theta | X, Y]$$

$$[*][\theta][*][*][\theta] \quad (9.58)$$

$$= [\tau][\tau]\phi(x)m = m\phi(x),$$

$$[*][N][N][*]$$

$$V[\tau][f(x) | X, Y] = V[\phi(x)\theta | X, Y]$$

$$[\theta][*][\theta][*]$$

$$= [\tau]\phi(x)V[\theta | X, Y]\phi(x) \quad (9.59) \quad [*][\theta][*]$$

$$= [\tau]\phi(x)S\phi(x).$$

$$[*][N][*]$$

我们看到预测均值与有噪声观测的预测均值相同，因为噪声的均值为0，预测方差仅相差 $[2]\sigma$ ，这是测量噪声的方差：当我们预测有噪声的函数值时，我们需要包含 $[2]\sigma$ 作为不确定性的来源，但对于无噪声预测则不需要这一项。这里，

唯一剩余的不确定性来自参数后验。◇

[积分掉] 注释（函数上的分布）.我们积分掉 [[参数导致]]

参数[[函数上的分布：] θ 导致函数上的分布：如果我们从参数后验 $p(\theta | X, Y)$ 中采样 $\theta[i] \sim$ [函数。] 我们得到单个函数实

现 $[\tau]\theta\phi(\cdot)$ 。这个函数分布的均值函数，即所有期望函数 $[i]$ [[均值函数]]

值 $[\tau]E[f(\cdot) | \theta, X, Y]$ 的集合，是 $m\phi(\cdot)$ 。 $[\theta][N]$

(边际) 方差, 即函数 $f(\cdot)$ 的方差, 由 $\phi^T \Sigma \phi(\cdot)$ 给出。◇ [N]

例9.8 (函数上的后验)

让我们重新审视具有

度数15的多项式的贝叶斯线性回归问题。我们选择参数先验 $p(\theta) = N(0, I)$ 。图[9.9]

[4]

可视化了由参数先验导致的函数先验以及从该先验中采样的函数。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由Cambridge University Press出版 (2020).]

[310] [[线性回归]]

图[9.10]显示了我们通过

贝叶斯线性回归获得的函数后验。训练数据集显示在面板[(a)]中;

面板[(b)]显示了函数上的后验分布, 包括

我们通过最大似然和MAP估计得到的函数。

我们使用MAP估计得到的函数也对应于

贝叶斯线性回归设置中的后验均值函数。面板[(c)]

显示了在该函数后验下的一些合理实现(样本)。

[图9.10]

[贝叶斯线性] [4] [4] [回归和] [2] [2] [2] [函数上的后验] [y] [0] [y] [0] [y] [0] [函数。] [训练数据]

[[(a) 训练数据;] [-][2] [-][2] [MLE] [-][2]

[MAP]

[[(b) 函数上的] [-][4] [-][4] [BLR] [-][4] [后验] [-][4] [-][2] [0] [2] [4] [-][4] [-][2] [0] [2] [4] [-][4] [-][2] [0] [2] [4] x x x [分布;]

[(a) 训练数据。] [(b) 由边际不确定性(阴影)表示的函数后验, 显示67%和95%的预测置信区间、最大似然估计(MLE)和MAP估计(MAP), 后者与后验均值函数相同。] [(c) 从函数后验中采样, 这些是由参数后验中的样本导致的。]

[[(c) 从函数后验]

[中采样。]

图[9.11]显示了一些函数上的后验分布

根据参数后验分布。对于不同的多项式阶数 M ，左侧面板显示了最大似然函数 $\theta_{ML}^T \phi(\cdot)$ 、MAP函数 $\theta_{MAP}^T \phi(\cdot)$ （与后验均值函数相同），以及通过Bayesian线性回归获得的67%和95%预测置信区间，用阴影区域表示。

右侧面板显示了函数后验分布的样本：在这里，我们从参数后验分布中采样参数 θ^{12} ，并计算函数 $\phi(x^*)^T \theta^{12}$ ，这是函数在后验分布下的单次实现。对于低阶多项式，参数后验分布不允许参数变化太大：采样的函数几乎相同。当我们通过添加更多参数（即得到更高阶的多项式）使模型更加灵活时，这些参数没有被后验分布充分约束，采样的函数在视觉上可以轻易区分。我们还可以在左侧相应的面板中看到不确定性如何增加，特别是在边界处。

尽管对于七阶多项式，MAP估计产生了合理的拟合，但Bayesian线性回归模型还告诉我们

图9.11 Bayesian线性回归。左侧面板：阴影区域表示67%（深灰色）和95%（浅灰色）预测置信区间。Bayesian线性回归模型的均值与MAP估计一致。预测不确定性是噪声项和后验参数不确定性的总和，这取决于测试输入的位置。右侧面板：从后验分布中采样的函数。

图9.11 Bayesian线性回归。左侧面板：阴影区域表示67%（深灰色）和95%（浅灰色）预测置信区间。Bayesian线性回归模型的均值与MAP估计一致。预测不确定性是噪声项和后验参数不确定性的总和，这取决于测试输入的位置。右侧面板：从后验分布中采样的函数。

- a. 阶数 $M = 3$ 的多项式的后验分布（左）和从函数后验分布中的采样（右）。
- b. 阶数 $M = 5$ 的多项式的后验分布（左）和从函数后验分布中的采样（右）。
- c. 阶数 $M = 7$ 的多项式的后验分布（左）和从函数后验分布中的采样（右）。

后验不确定性是巨大的。当我们在决策系统中使用这些预测时，这些信息可能至关重要，因为错误的决策可能会产生严重后果（例如，在强化学习或机器人技术中）。

9.3.5 计算边际似然

在第8.6.2节中，我们强调了边际似然对于Bayesian模型选择的重要性。在下面，我们计算具有参数conjugate Gaussian先验的Bayesian线性回归的边际似然，即正是我们在本章中一直讨论的设置。

回顾一下，我们考虑以下生成过程：

$$\theta \sim N(\mathbf{m}_0, \mathbf{S}_0) \quad (9.60a) \quad y_n \mid \mathbf{x}_n, \theta \sim N(\mathbf{x}_n^\top \theta, \sigma^2), \quad (9.60b)$$

$n = 1, \dots, N$ 。边际似然由以下给出：

$$p(\mathbf{Y} \mid \mathbf{X}) = \int p(\mathbf{Y} \mid \mathbf{X}, \theta) p(\theta) d\theta \quad (9.61a) = \int N(\mathbf{y} \mid \mathbf{X}\theta, \sigma^2 \mathbf{I}) N(\theta \mid \mathbf{m}_0, \mathbf{S}_0) d\theta, \quad (9.61b)$$

其中我们积分掉模型参数 θ 。我们分两步计算边际似然：首先，我们证明边际似然是Gaussian分布（作为 $\mathbf{y}$ 的分布）；其次，我们计算这个Gaussian分布的均值和协方差。

1. 边际似然是Gaussian分布：从第6.5.2节，我们知道(i)两个Gaussian随机变量的乘积是一个（非标准化的）Gaussian分布，(ii)Gaussian随机变量的线性变换是Gaussian分布。在(9.61b)中，我们需要一个线性变换将 $N(\mathbf{y} \mid \mathbf{X}\theta, \sigma^2 \mathbf{I})$ 转换为 $N(\theta \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ 的形式，对于某些 $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ 。一旦完成，积分可以以闭合形式求解。结果是两个Gaussian分布乘积的标准化常数。标准化常数本身具有Gaussian形状；见(6.76)。
2. 均值和协方差。我们通过利用随机变量仿射变换的均值和协方差的标准结果来计算边际似然的均值和协方差矩阵；见第6.4.4节。边际似然的均值计算为

$$E[\mathbf{Y} \mid \mathbf{X}] = E_{\{\theta, \boldsymbol{\varepsilon}\}}[\mathbf{X}\theta + \boldsymbol{\varepsilon}] = \mathbf{X} E_{\theta}[\theta] = \mathbf{X} \mathbf{m}_0. \quad (9.62)$$

注意 $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 \mathbf{I})$ 是i.i.d.随机变量的向量。协方差矩阵由以下给出：

$$\text{Cov}[\mathbf{Y} \mid \mathbf{X}] = \text{Cov}_{\{\theta, \boldsymbol{\varepsilon}\}}[\mathbf{X}\theta + \boldsymbol{\varepsilon}] = \text{Cov}_{\theta}[\mathbf{X}\theta] + \sigma^2 \mathbf{I} \quad (9.63a) = \mathbf{X} \text{Cov}_{\theta}[\theta] \mathbf{X}^\top + \sigma^2 \mathbf{I} = \mathbf{X} \mathbf{S}_0 \mathbf{X}^\top + \sigma^2 \mathbf{I}. \quad (9.63b)$$

因此，边际似然是

$$p(\mathbf{Y} \mid \mathbf{X}) = (2\pi)^{-N/2} \det(\mathbf{X} \mathbf{S}_0 \mathbf{X}^\top + \sigma^2 \mathbf{I})^{-1/2} \quad (9.64a) \cdot \exp[-1/2 (\mathbf{y} - \mathbf{X} \mathbf{m}_0)^\top (\mathbf{X} \mathbf{S}_0 \mathbf{X}^\top + \sigma^2 \mathbf{I})^{-1} (\mathbf{y} - \mathbf{X} \mathbf{m}_0)]$$



图9.12 最小二乘的几何解释。

图9.12 最小二乘的几何解释。

9.4 最大似然作为正交投影

在通过大量代数运算推导出最大似然和MAP估计之后，我们现在将提供最大似然估计的几何解释。让我们考虑一个简单的线性回归设置

$$y = x\theta + \epsilon, \epsilon \sim N(0, \sigma^2), (9.65)$$

其中我们考虑通过原点的线性函数 $f: \mathbb{R} \rightarrow \mathbb{R}$ （为了清晰起见，我们在这里省略特征）。参数 θ 决定直线的斜率。图9.12(a)显示了一个一维数据集。



图9.12: (a) 回归数据集，包含函数值 $f(x_n)$ 在输入位置 x_n 处的噪声观测值 y_n （蓝色）



图9.12: (b) 橙色点是噪声观测值（蓝点）在直线 $\theta_{ML} x$ 上的投影。线性回归问题的最大似然解找到一个子空间（直线），使观测值的总体投影误差（橙色线）最小化

给定训练数据集 $\{(x_1, y_1), \dots, (x_n, y_n)\}$ ，我们回顾第9.2.1节的结果，得到斜率参数的最大似然估计量为

$$\theta_{ML} = (X^T X)^{-1} X^T y = (X^T y) / (X^T X) \in \mathbb{R} \quad (9.66)$$

其中 $X = [x_1, \dots, x_n]^T \in \mathbb{R}^N$, $y = [y_1, \dots, y_n]^T \in \mathbb{R}^N$ 。

这意味着对于训练输入 X ，我们得到训练目标的最优（最大似然）重建为

$$X \theta_{ML} = X (X^T y) / (X^T X) = (X X^T y) / (X^T X) \quad (9.67)$$

即，我们得到 y 和 $X\theta$ 之间最小二乘误差最小的近似。

由于我们在寻找 $y = X\theta$ 的解，我们可以将线性回归视为求解线性方程组的问题。因此，我们可以联系到第2章和第3章讨论的线性代数和解析几何概念。特别是，仔细观察(9.67)，我们看到在我们的示例(9.65)中，最大似然估计量 θ_{ML} 实际上是将 y 正交投影到由 X 张成的一维子空间上。回顾第3.8节关于正交投影的结果，我们将 $XX^T / (X^T X)$ 识别为投影矩阵， θ_{ML} 作为投影到由 X 张成的一维子空间的坐标， $X\theta_{ML}$ 作为 y 到该子空间的正交投影。

因此，最大似然解还通过找到由 X 张成的子空间中“最接近”相应观测值 y 的向量，提供了几何上的最优解，其中“最接近”意味着函数值 y_n 到 $x_n \theta$ 的最小（平方）距离。这通过正交投影实现。图9.12(b)显示了噪声观测值投影到最小化原始数据集与其投影之间平方距离的子空间上（注意 x 坐标是 $fixed$ ），这对应于最大似然解。

在一般线性回归情况下，其中

$$y = \varphi(x)^T \theta + \epsilon, \epsilon \sim N(0, \sigma^2) \quad (9.68)$$

具有向量值特征 $\varphi(x) \in \mathbb{R}^K$ ，我们再次可以将最大似然结果解释为

$$y \approx \Phi \theta_{ML}, (9.69) \quad \theta_{ML} = (\Phi^T \Phi)^{-1} \Phi^T y \quad (9.70)$$

作为投影到 $\mathbb{R}^N$ 的 K 维子空间，该子空间由特征矩阵 Φ 的列张成；参见第3.8.2节。

如果我们用来构造特征矩阵 Φ 的特征函数 φ_k 是正交归一的（参见第3.7节），我们得到一个特殊情况，其中 Φ 的列形成正交归一基（参见第3.5节），使得 $\Phi^T \Phi = I_K$ 。这将导致投影

$$(\Phi^T \Phi)^{-1} \Phi^T y = \Phi^T y = \sum_{k=1}^K \varphi_k \varphi_k^T y \quad (9.71)$$

因此最大似然投影简单地是 y 到各个基向量 φ_k （即 Φ 的列）的投影之和。此外，由于基的正交性，不同特征之间的耦合消失了。信号处理中许多流行的基函数，如小波(wavelets)和傅里叶(Fourier)基，都是正交基函数。

当基不是正交的时候，可以使用Gram-Schmidt过程将一组线性无关的基函数转换为正交基；参见第3.8.3节和(Strang, 2003)。

9.5 进一步阅读

在本章中，我们讨论了Gaussian似然和模型参数的共轭Gaussian先验的线性回归。这允许闭式Bayesian推断。然而，在某些应用中，我们可能希望选择不同的似然函数。例如，在二元分类设置中，我们只观察到两种可能的（分类）结果，在这种设置下Gaussian似然是不合适的。相反，我们可以选择Bernoulli似然，它将返回

预测标签为1（或0）。我们参考[Barber (2012),] [Bishop (2006),] 和[Murphy (2012)]的书籍来深入了解分类问题。非高斯似然重要的另一个例子是计数数据。计数是非负整数，在这种情况下，二项分布或泊松似然比高斯分布是更好的选择。所有这些例子都属于 *generalized linear models*（广义线性模型）的范畴，这是线性回归的灵活推广，允许响应变量具有高斯分布以外的误差分布。GLM通过允许线性模型通过光滑可逆函数 $\sigma(\cdot)$ 与观测值相关联来推广线性回归，该函数可能是非线性的，使得 $y = \sigma(f(x))$ ，其中 $f(x) = \theta^T \phi(x)$ 是来自(9.13)的线性回归模型。因此，我们可以将generalized linear model视为函数复合 $y = \sigma \circ f$ ，其中 f 是线性回归模型， σ 是activation function。请注意，尽管我们谈论“generalized linear models”，但输出 y 不再是参数 θ 的线性函数。

在 *logistic regression* 中，我们选择 *logistic sigmoid* $\sigma(f) = 1/(1+\exp(-f)) \in [0,1]$ ，它可以解释为观察到伯努利随机变量 $y \in \{0,1\}$ 的 $y = 1$ 的概率。函数 $\sigma(\cdot)$ 称为 *transfer function* 或 *activation function*，其逆函数称为 *canonical link function*。从这个角度来看，很明显generalized linear models是（深度）前馈神经网络的构建块：如果我们考虑generalized linear model $y = \sigma(Ax + b)$ ，其中 A 是权重矩阵， b 是偏置向量，我们将此generalized linear model识别为具有activation function $\sigma(\cdot)$ 的单层神经网络。

对于普通线性回归，activation function只是恒等函数。我们现在可以通过以下方式递归地组合这些函数：

$$f(x^{(k+1)}) = f^{(k)}(x_k) = \sigma(A^{(k)} x^{(k)} + b^{(k)}) \quad (9.72)$$

对于 $k = 0, \dots, K-1$ ，其中 $x^{(0)}$ 是输入特征， $x^K = y$ 是观测输出，使得 $f = f^{(K-1)} \circ \dots \circ f^{(0)}$ 是 K 层深度神经网络。因此，这个深度神经网络的构建块是在(9.72)中定义的generalized linear models。

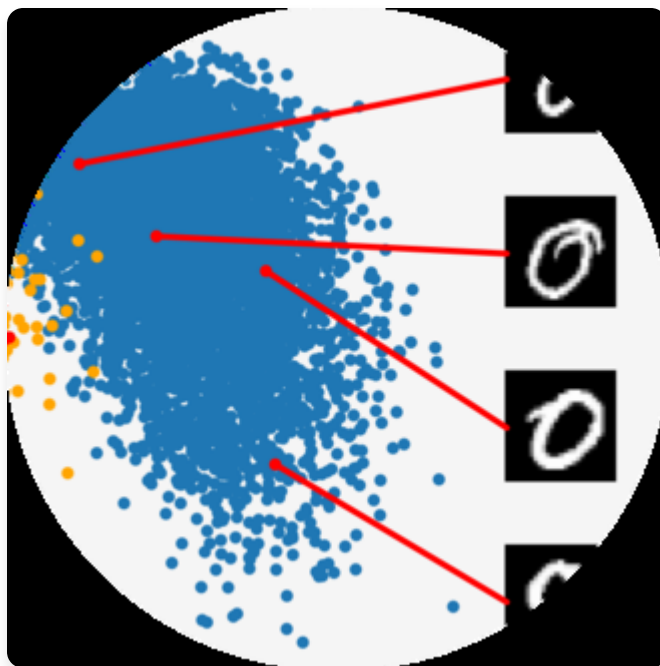
神经网络[Bishop, 1995; Goodfellow et al., 2016]比线性回归模型显著更具表达力和灵活性。然而，最大似然参数估计是非凸优化问题，在完全贝叶斯设置中参数的边际化在分析上是难以处理的。

我们简要提到了参数分布诱导回归函数分布的事实。*Gaussian processes* [Rasmussen and Williams, 2006]是回归模型，其中函数分布的概念是核心。Gaussian process不是在参数上放置分布，而是直接在函数空间上放置分布，无需通过参数的“绕行”。为此，Gaussian process利用 *kernel trick* [(Schölkopf and Smola, 2002)]，它允许我们仅通过查看相应的输入 x_i, x_j 来计算两个函数值 $f(x_i), f(x_j)$ 之间的内积。Gaussian process与贝叶斯线性回归和支持向量回归密切相关，但也可以解释为具有单个隐藏层的贝叶斯神经网络，其中单元数趋于无穷大[(Neal, 1996; Williams, 1997)]。Gaussian processes的优秀介绍可以在[MacKay (1998)]和[Rasmussen and Williams (2006)]中找到。

我们在本章的讨论中专注于高斯参数先验，因为它们允许在线性回归模型中进行闭式推断。然而，即使在具有高斯似然的回归设置中，我们也可以选择非高斯先验。考虑一个设置，其中输入是 $x \in \mathbb{R}^D$ ，我们的训练集很小，大小为 $N \ll D$ 。这意味着回归问题是欠定的。在这种情况下，我们可以选择强制稀疏性的参数先验，即试图将尽可能多的参数设置为0的先验（*variable selection*）。这种先验提供了比高斯先验更强的正则化器，这通常导致预测精度的提高和模型的可解释性。拉普拉斯先验是经常用于此目的的一个例子。参数上带有拉普拉斯先验的线性回归模型等价于带有L1正则化的线性回归（*LASSO*）[Tibshirani, 1996]。拉普拉斯分布在零处尖锐达到峰值（其一阶导数不连续），它

将概率质量集中在比高斯分布更接近零的地方，这鼓励参数为0。因此，非零参数与回归问题相关，这就是我们也谈论“variable selection”的原因。

使用主成分分析进行降维



成分分析

直接处理高维数据（如图像）会遇到一些困难：难以分析、解释困难、可视化几乎不可能，而且（从实用角度来看）数据向量的存储可能很昂贵。一个 640×480 像素的彩色图像是一个百万维空间中的数据点，其中每个像素对应三个维度，分别对应三个颜色通道（红、绿、蓝）。然而，高维数据通常具有我们可以利用的属性。例如，高维数据通常是过完备的，即许多维度是冗余的，可以通过其他维度的组合来解释。此外，高维数据中的维度通常是相关的，因此数据具有内在的低维结构。降维利用结构和相关性，使我们能够使用更紧凑的数据表示，理想情况下不会丢失信息。我们可以将降维视为一种压缩技术，类似于jpeg或mp3，它们是图像和音乐的压缩算法。

在本章中，我们将讨论**主成分分析（PCA）**，这是一种线性降维算法。PCA由Pearson (1901)和Hotelling (1933)提出，已经存在100多年，至今仍是数据压缩和数据可视化最常用的技术之一。它还用于识别简单模式、潜在因子和高维数据的结构。在信号处理社区中，PCA也被称为**Karhunen-Loève变换**。在本章中，我们从第一原理推导PCA，借鉴我们对基和基变换（第2.6.1节和第2.7.2节）、投影（第3.8节）、特征值（第4.2节）、高斯分布（第6.5节）和约束优化（第7.2节）的理解。



图10.1 降维示例：(a) 原始数据集在x2方向上变化不大。(b) 来自(a)的数据可以仅使用x1坐标表示，几乎没有损失。

图10.1 降维示例：(a) 原始数据集在x2方向上变化不大。(b) 来自(a)的数据可以仅使用x1坐标表示，几乎没有损失。降维通常利用高维数据（如图像）的一个属性，即它通常位于低维子空间上。图10.1给出了二维的说明性示例。虽然图10.1(a)中的数据并不完全位于一条线上，但数据在x2方向上变化不大，因此我们可以将其表示为好像在一条线上——几乎没有损失；见图10.1(b)。要描述图10.1(b)中的数据，只需要x1坐标，数据位于 $\mathbb{R}^2$ 的一维子空间中。

10.1 问题设定

在PCA中，我们感兴趣的是找到数据点 x_n 的投影 $\tilde{x}_n$ ，这些投影尽可能与原始数据点相似，但具有显著较低的内在维度。图10.1给出了这可能的样子的说明。

更具体地说，我们考虑一个独立同分布数据集 $X = \{x_1, \dots, x_N\}$ ， $x_n \in \mathbb{R}^D$ ，均值为0，具有**数据协方差矩阵**

$$S_X = (1/N) \sum_{n=1}^N x_n x_n^T \quad (10.1)$$

此外，我们假设存在一个低维压缩表示（编码）

$$z_n = B^T x_n \in \mathbb{R}^M \quad (10.2)$$

其中 x_n ，我们定义投影矩阵

$$B := [b_1, \dots, b_M] \in \mathbb{R}^D \times M \quad (10.3)$$

我们假设 B 的列是标准正交的（定义3.7），使得 $b_i^T b_j = 0$ 当且仅当 $i \neq j$ ，且 $b_i^T b_i = 1$ 。我们寻求一个 M 维子空间 $U \subseteq \mathbb{R}^D$ ， $\dim(U) = M < D$ ，将数据投影到其上。 B 的列 $b_1, \dots, b_M$ 形成投影数据所在 M 维子空间的基。我们用 $\tilde{x}_n \in U$ 表示投影数据，用 z_n 表示它们的坐标（相对于 U 的基向量 $b_1, \dots, b_M$ ）。我们的目标是找到投影 $\tilde{x}_n \in \mathbb{R}^D$ （或等价地找到编码 z_n 和基向量 $b_1, \dots, b_M$ ），使得它们与原始数据 x_n 尽可能相似，并最小化由于压缩造成的损失。

示例10.1（坐标表示/编码）

考虑 $\mathbb{R}^2$ 与标准基 $e_1 = [1, 0]^T$ ， $e_2 = [0, 1]^T$ 。



图10.2 PCA的图形说明。在PCA中，我们找到原始数据 x 的压缩版本 z 。压缩数据可以重构为 $\tilde{x}$ 。

图10.2 PCA的图形说明。在PCA中，我们找到原始数据 x 的压缩版本 z 。压缩数据可以重构为 $\tilde{x}$ 。

第二章，我们知道 $\mathbf{x} \in \mathbb{R}^D$ 的 [原始数据空间] 中具有内在的 [低维表示，比] 可以表示为这些基向量的线性组合，例如， $\mathbf{x}$ 。

$$= 5\mathbf{e}_1 + 3\mathbf{e}_2. \quad (10.4)$$

然而，当我们考虑形式为

$$\tilde{\mathbf{x}} = \mathbf{z} \in \mathbb{R}^2, \quad \mathbf{z} \in \mathbb{R}, \quad (10.5)$$

的向量时，它们总是可以写成 $0\mathbf{e}_1 + z\mathbf{e}_2$ 。为了表示这些向量，记住/存储 $\tilde{\mathbf{x}}$ 相对于 $\mathbf{e}_2$ 向量的 坐标/代码 z 就足够了。向量空间的维数

更精确地说， $\tilde{\mathbf{x}}$ 向量的集合（具有标准向量加法和标量乘法）形成一个向量空间 U （参见第 2.4 节），其中 $\dim(U) = 1$ ，因为 $U = \text{span}[\mathbf{e}_2]$ 。对应于其基向量的数量（参见第 2.6.1 节）。

在第 10.2 节中，我们将找到保留尽可能多信息并最小化压缩损失的低维表示。

PCA 的另一种推导在第 10.3 节中给出，我们将寻求最小化原始数据 $\mathbf{x}_n$ 和其投影 $\tilde{\mathbf{x}}_n$ 之间的平方重构误差 $\|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2$ 。

图 10.2 说明了我们在 PCA 中考虑的设置，其中 z 表示压缩数据 $\tilde{\mathbf{x}}$ 的低维表示，并起到瓶颈的作用，控制 $\mathbf{x}$ 和 $\tilde{\mathbf{x}}$ 之间可以流动多少信息。在 PCA 中，我们考虑原始数据 $\mathbf{x}$ 和其低维代码 z 之间的线性关系，使得 $z = \mathbf{B}^T \mathbf{x}$ 和 $\tilde{\mathbf{x}} = \mathbf{B} \mathbf{z}$ ，其中 $\mathbf{B}$ 是合适的矩阵。基于将 PCA 视为数据压缩技术的动机，我们可以将图 10.2 中的箭头解释为表示编码器和解码器的操作对。由 $\mathbf{B}$ 表示的线性映射可以被视为解码器，它将低维代码 $z \in \mathbb{R}^M$ 映射回原始数据空间 $\mathbb{R}^D$ 。类似地， $\mathbf{B}^T$ 可以被视为编码器，它将原始数据 $\mathbf{x}$ 编码为低维（压缩）代码 z 。

在本章中，我们将使用 MNIST 数字数据集作为经

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由剑桥大学出版社出版 (2020)。

320 主成分分析降维

图 10.3



MNIST 数据集中手写数字的示例。



[手写数字]



[来自 MNIST]



[数据集。] <http://yann.lecun.com/exdb/mnist/>



[//yann.lecun.](http://yann.lecun.com/exdb/mnist/)



[com/exdb/mnist/](http://yann.lecun.com/exdb/mnist/).



常出现的例子，包含 60,000 个手写数字 0 到 9 的示例。每个数字是大小为 28×28 的灰度图像，即包含



784 个像素，因此我们可以将此数据集中的每个图像解释为向量



$x \in \mathbb{R}^{784}$ 。这些数字的示例如图 10.3 所示。



10.2 最大方差视角

图 10.1 给出了如何使用单一坐标表示二维数据集的示例。在图 10.1(b) 中，我们选择忽略数据的 x_2 坐标，因为它没有增加太多信息，所以压缩数据与图 10.1(a) 中的原始数据相似。我们本可以选择忽略 x_1 坐标，但那样压缩数据就会与原始数据非常不同，数据中的大量信息会丢失。

如果我们将数据中的信息内容解释为数据集的“空间填充”程度，那么我们可以通过查看数据的分布来描述数据中包含的信息。从第 6.4.1 节中，我们知道方差是数据分布的指标，我们可以将 PCA 推导为一种降维算法，该算法最大化数据的低维表示中的方差，以保留尽可能多的信息。图 10.4 说明了这一点。

考虑第 10.1 节中讨论的设置，我们的目标是找到一个矩阵 B （参见 (10.3)），在通过将数据投影到由 B 的列 $b_1, \dots, b_M$ 张成的子空间上来压缩数据时，保留尽可能多的信息。数据压缩后保留最多信息等价于在低维代码中捕获最大方差量 (Hotelling, 1933)。

备注。（中心化数据）对于 (10.1) 中的数据协方差矩阵，我们假设数据已中心化。我们可以在不失一般性的情况下做这个假设：假设 μ 是数据的均值。使用我们在第 6.4.4 节中讨论的方差性质，我们得到

$$V[z] = V[B^T(x - \mu)] = V[B^T x - B^T \mu] = V[B^T x], \quad (10.6)$$

即，低维代码的方差不依赖于数据的均值。因此，我们在本节的剩余部分假设数据的均值为 0，不失一般性。在这个假设下，低维代码的均值也是 0，因为 $E[z] = E[B^T x] = B^T E x = 0$ 。◇

x

《机器学习数学》草稿 (2023-10-18)。反馈：<https://mml-book.com>。10.2 最大方差视角 321

图 10.4 PCA 找到一个低维子空间（直线），当数据（蓝色）投影到该子空间（橙色）时，保持尽可能多的方差（数据的分布）。

10.2.1 具有最大方差的方向

我们使用序列方法最大化低维编码的方差。我们首先寻找一个向量 $b_1 \in \mathbb{R}^D$ ，使投影数据的方差最大化，即我们的目标是最大化

$$V_1 := \frac{1}{N} \sum_{n=1}^N z_{1n}^2$$

的方差，其中我们利用了数据的独立同分布假设，并将 z_{1n} 定义为 $x_n \in \mathbb{R}^D$ 的低维表示 $z_n \in \mathbb{R}^M$ 的第一个坐标。注意 z 的第一个分量由下式给出：

$$z_{1n} = b_1^T x_n$$

即，它是 x_n 在由 b_1 张成的一维子空间上的正交投影的坐标（第 3.8 节）。我们将方程 (10.8) 代入方程 (10.7)，得到

$$V_1 = \frac{1}{N} \sum_{n=1}^N (b_1^T x_n)^2 = \frac{1}{N} b_1^T \left(\sum_{n=1}^N x_n x_n^T \right) b_1$$

$$b_1^T \left(\frac{1}{N} \sum_{n=1}^N x_n x_n^T \right) b_1 = b_1^T S b_1$$

其中 S 是在方程 (10.1) 中定义的数据协方差矩阵。在方程 (10.9a) 中，我们使用了两个向量的点积对其参数是对称的这一事实，即 $b_1^T x_n = x_n^T b_1$ 。

注意任意增加向量 b_1 的大小会增加 V_1 ，即一个两倍长的向量 b_1 可能导致 V_1 增大四倍。因此，我们将所有解限制为 $\|b_1\|_2 = 1$ ，这导致一个约束优化问题，其中我们寻找数据变化最大的方向。

在将解空间限制为单位向量的情况下，指向最大方差方向的向量 b_1 可以通过以下约束优化问题找到：

$$\max_{b_1} b_1^T S b_1 \text{ subject to } \|b_1\|_2^2 = 1$$

遵循第7.2节，我们得到拉格朗日函数

$$L(b_1, \lambda_1) = b_1^T S b_1 + \lambda_1 (1 - b_1^T b_1)$$

来解决这个约束优化问题。 L 对 b_1 和 λ_1 的偏导数分别为

$$\frac{\partial L}{\partial b_1} = 2Sb_1 - 2\lambda_1 b_1, \quad \frac{\partial L}{\partial \lambda_1} = 1 - b_1^T b_1$$

将这些偏导数设为0，我们得到关系式

$$Sb_1 = \lambda_1 b_1^T b_1 = \lambda_1$$

通过与特征值分解的定义（第4.4节）比较，我们看到 b_1 是数据协方差矩阵 S 的特征向量，拉格朗日乘子 λ_1 扮演相应特征值的角色。这个特征向量性质 (10.13) 允许我们将方差目标 (10.10) 重写为

$$V_1 = b_1^T S b_1 = \lambda_1 b_1^T b_1 = \lambda_1$$

即，投影到一维子空间上的数据方差等于与张成该子空间的基向量 b_1 相关的特征值。因此，为了最大化低维编码的方差，我们选择与数据协方差矩阵的最大特征值相关的基向量。这个特征向量称为第一个**主成分**(principal component)。我们可以通过将坐标 z_{1n} 映射回数据空间来确定主成分 b_1 在原始数据空间中的影响/贡献，这给出了投影的数据点

$$\tilde{x}_n = b_1 z_{1n} = b_1 b_1^T x_n \in \mathbb{R}^D$$

在原始数据空间中。

备注。 虽然 $\tilde{x}_n$ 是一个 D 维向量，但它只需要一个坐标 z_{1n} 来相对于基向量 $b_1 \in \mathbb{R}^D$ 表示它。◇

10.2.2 具有最大方差的M维子空间

假设我们已经找到了前 $m-1$ 个主成分，作为与前 $m-1$ 个最大特征值相关的 S 的 $m-1$ 个特征向量。由于 S 是对称的，谱定理（定理4.15）表明我们可以使用这些特征向量来构造 $\mathbb{R}^D$ 的一个 $(m-1)$ 维子空间的正交归一化特征基。通常，第 m 个主成分可以通过从数据中减去前 $m-1$ 个主成分 $b_1, \dots, b_{m-1}$ 的影响来找到，从而试图找到压缩剩余信息的主成分。然后我们得到新的数据矩阵

$$\hat{X} := X - \sum_{i=1}^{m-1} b_i b_i^{\top} X = X - B_{m-1} X$$

其中 $X = [x_1, \dots, x_N] \in \mathbb{R}^{D \times N}$ 包含作为列向量的数据点， $\hat{X} = [\hat{x}_1, \dots, \hat{x}_N] \in \mathbb{R}^{D \times N}$ 在 (10.17) 中包含尚未被压缩的数据中的信息。矩阵 $I - B_{m-1} = I - \sum_{i=1}^{m-1} b_i b_i^{\top}$ 是一个投影矩阵，投影到由 $b_1, \dots, b_{m-1}$ 张成的子空间上。

备注（记号）。 在本章中，我们不遵循将数据 $x_1, \dots, x_N$ 收集为数据矩阵的行的惯例，而是将它们定义为 X 的列。这意味着我们的数据矩阵

矩阵 X 是一个 $D \times N$ 矩阵，而不是传统的 $N \times D$ 矩阵。我们这样选择的原因是代数运算可以顺利进行，无需转置矩阵或将向量重新定义为左乘到矩阵上的行向量。◇

为了找到第 m 个主成分，我们最大化方差

$$\frac{1}{N} \sum_{i=1}^N \|\hat{x}_i\|^2 = \frac{1}{N} \text{tr}(\hat{X}^{\top} \hat{X}) = \frac{1}{N} \text{tr}(X^{\top} (I - B_{m-1}) X) = \frac{1}{N} \text{tr}(X^{\top} X - X^{\top} B_{m-1} X) \quad (10.18)$$

受约束 $\|b\| = 1$ ，其中我们遵循了与 (10.9b) 中相同的步骤 [m]

并定义 $\hat{S}$ 为变换数据集 $\hat{X} := \{\hat{x}_1, \dots, \hat{x}_N\}$ 的数据协方差矩阵。如前所述，当我们单独考虑第一个主成分时，我们解决一个约束优化问题并发现最优解 $\hat{b}^{[1]}$ 是与 $\hat{S}$ 的最大特征值相关联的 $\hat{S}$ 的特征向量。

事实证明 $b^{[m]}$ 也是 S 的特征向量。更一般地， $\hat{\hat{S}}$ 和 S 的特征向量集合是相同的。由于 S 和 $\hat{S}$ 都是对称的，我们可以找到特征向量的 ONB (谱定理 [4.15])，即，对于 S 和 $\hat{S}$ 都存在 D 个不同的特征向量。接下来，我们表明 $\hat{S}$ 的每个特征向量都是 S 的特征向量。假设我们已经找到了 S 的特征向量 $b_1, \dots, b_{m-1}$ 。考虑 S 的特征向量 $b^{[i]}$ ，即 $Sb^{[i]} = \lambda^{[i]}b^{[i]}$ 。一般来说，

$$Sb^{[1]} = X X^{\top} b^{[1]} = (X - B_{m-1} X)(X - B_{m-1} X)^{\top} b^{[1]} \quad (10.19a)$$

$$= (S - SB_{m-1} - B_{m-1}^{\top} S + B_{m-1}^{\top} S B_{m-1}) b^{[1]} \quad (10.19b)$$

我们区分两种情况。如果 $i \geq m$ ，即 $b^{[i]}$ 是不在前 $m-1$ 个主成分中的特征向量，那么 $b^{[i]}$ 与前 $m-1$ 个主成分正交，且 $B_{m-1} b^{[i]} = 0$ 。如果 $i < m$ ，即 $b^{[i]}$ 在前 $m-1$ 个主成分中，那么 $b^{[i]}$ 是主子空间的基向量， B_{m-1} 投影到该子空间上。由于 $b_1, \dots, b_{m-1}$ 是这个主子空间的 ONB，我们得到 $B_{m-1} b^{[i]} = b^{[i]}$ 。这两种情况可以总结如下：

$$B_{m-1} b^{[i]} = b^{[i]} \text{ 如果 } i < m, B_{m-1} b^{[i]} = 0 \text{ 如果 } i \geq m. \quad (10.20)$$

在情况 $i \geq m$ 中，通过在 (10.19b) 中使用 (10.20)，我们得到 $Sb^{[i]} = (S - B_{m-1}^{\top} S) b^{[i]} = Sb^{[i]} = \lambda^{[i]} b^{[i]}$ ，即 $b^{[i]}$ 也是 S 的特征向量，特征值为 $\lambda^{[i]}$ 。具体地，

$$Sb^{\wedge}[m] = Sb[m] = \lambda[m]b[m] \cdot [(10.21)]$$

方程 [(10.21)] 显示 $b[m]$ 不仅是 S 的特征向量，也是 $\hat{\hat{S}}$ 的特征向量。具体地， $\lambda[m]$ 是 S 的最大特征值， $\lambda[m]$ 是 S 的第 m 大特征值，两者都有相关联的特征向量 $b[m]$ 。

在情况 $i < m$ 中，通过在 [(10.19b)] 中使用 [(10.20)]，我们得到

$$Sb^{\wedge}[i] = (S - SB[m][-]1 - B[m][-]1S + B[m][-]1SB[m][-]1)b[i] = 0 = 0b[i] \quad (10.22)$$

这意味着 $b^{\wedge}1, \dots, b^{\wedge}[m][-]1$ 也是 S 的特征向量，但它们与特征值 0 相关联，所以 $b^{\wedge}1, \dots, b^{\wedge}[m][-]1$ 张成 S 的零空间。

总的来说， $\hat{\hat{S}}$ 的每个特征向量也是 S 的特征向量。然而，如果 S 的特征向量是 $(m-1)$ 维主子空间的一部分，那么 $\hat{\hat{S}}$ 的相关特征值为 0 。

[这个推导表明在] [具有最大方差的] 通过关系 [(10.21)] 和 $b[m]^T b[m] = 1$ ，投影到第 m 个主成分上的数据方差为 $[M$ 维子空间与]

[特征值分解之间] $V = b^{\wedge}[T] S b^{\wedge} = \lambda b^{\wedge}[T] b^{\wedge} = \lambda \cdot (10.23)$ [存在密切联系] [m] [m] [m] [m] [m] [m] [m] [我们将在]

[这意味着当数据投影到 M 维子空间时，数据的方差等于与数据协方差矩阵相应特征向量相关联的特征值之和。] [第 10.4 节中] [重新讨论这种] [联系]

示例 10.2 (MNIST “8” 的特征值)

图 10.5 [MNIST “8” 训练] [数据的性质。(a) [按降序排列的] [特征值；(b) 由] [与最大特征值] [相关联的主成分] [捕获的方差。]

50 500 40 400 [按降序排列的特征值] 30 值 [方差] 300 20 [特征] 200 [捕获的] 10 [主成分] 100 0 0 50 100 150 200 0 50 100 150 200 索引 [主成分数量]

a. MNIST 训练集中所有数字 “8” 的数据协方差矩阵的特征值（按降序排列）。[(b) 主成分捕获的方差。]

图 10.6 [投影方法的] [图解：找到一个] [子空间（直线）] [使投影后（橙色）] [和原始（蓝色）] [数据之间差向量] [的长度最小化。]

取 MNIST 训练数据中的所有数字 “8”，我们计算数据协方差矩阵的特征值。图 [10.5(a)] 显示了数据协方差矩阵的 200 个最大特征值。我们看到只有少数特征值与 0 显著不同。因此，当将数据投影到由相应特征向量张成的子空间时，大部分方差仅由少数主成分捕获，如图 [10.5(b)] 所示。

总的来说，为了找到保留尽可能多信息的 R^D 的 M 维子空间，PCA 告诉我们选择 [(10.3)] 中矩阵 B 的列为与 M 个最大特征值相关联的数据协方差矩阵 S 的 M 个特征向量。PCA 用前 M 个主成分能够捕获的最大方差量为

$$[M] \quad V[M] = \sum_{i=1}^M \lambda_i, \quad (10.24) \quad [m]$$

其中 $\lambda_{[m]}$ 是数据协方差矩阵 S 的 M 个最大特征值。因此，通过 PCA 数据压缩丢失的方差为

$$J := \sum_{i=M+1}^D \lambda_i = V[D] - V[M]. \quad (10.25) \quad [j] \quad [D] \quad [M] \quad j=M+1$$

与这些绝对量不同，我们可以定义相对方差捕获率为 $V[M]/V[D]$ ，压缩丢失的相对方差为 $1 - V[M]/V[D]$ 。

10.3 投影视角

在下文中，我们将推导 PCA 作为直接最小化平均重构误差的算法。这种视角允许我们将 PCA 解释为实现最优线性自编码器。我们将大量借鉴第 2 章和第 3 章的内容。

在上一节中，我们通过最大化投影空间中的方差来推导 PCA，以保留尽可能多的信息。在

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

326 主成分分析降维

图 10.7 简化投影设置。(a) 向量 $\mathbf{x} \in \mathbb{R}^2$ (红色十字) 将被投影到由 $\mathbf{b}$ 张成的一维子空间 $U \subseteq \mathbb{R}^2$ 。(b) 显示 $\mathbf{x}$ 与一些候选向量 $\tilde{\mathbf{x}}$ 之间的差异向量。

下文中，我们将研究原始数据 $\mathbf{x}[\mathbf{n}]$ 与其重构 $\tilde{\mathbf{x}}[\mathbf{n}]$ 之间的差异向量，并最小化这个距离，使得 $\mathbf{x}[\mathbf{n}]$ 和 $\tilde{\mathbf{x}}[\mathbf{n}]$ 尽可能接近。图 10.6 说明了这种设置。

10.3.1 设置和目标

假设一个 (有序) 正交标准基 (ONB) $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}[\mathbf{D}])$ of $\mathbb{R}[\mathbf{D}]$ ，即， $\mathbf{b}[\mathbf{i}]^T \mathbf{b}[\mathbf{j}] = 1$ 当且仅当 $\mathbf{i} = \mathbf{j}$ ，否则为 0。

从第 2.5 节我们知道，对于 $\mathbb{R}[\mathbf{D}]$ 的基 $(\mathbf{b}_1, \dots, \mathbf{b}[\mathbf{D}])$ ，任何 $\mathbf{x} \in \mathbb{R}[\mathbf{D}]$ 都可以写成基向量的线性组合，即，

$$\mathbf{x} = \sum_{(\mathbf{d}=1 \text{ to } \mathbf{D})} \zeta[\mathbf{d}] \mathbf{b}[\mathbf{d}] = \sum_{(\mathbf{m}=1 \text{ to } \mathbf{M})} \zeta[\mathbf{m}] \mathbf{b}[\mathbf{m}] + \sum_{(\mathbf{j}=\mathbf{M}+1 \text{ to } \mathbf{D})} \zeta[\mathbf{j}] \mathbf{b}[\mathbf{j}] \quad (10.26)$$

对于合适的坐标 $\zeta[\mathbf{d}] \in \mathbb{R}$ 。

我们感兴趣找到向量 $\tilde{\mathbf{x}} \in \mathbb{R}[\mathbf{D}]$ ，它们位于低维子空间 $U \subseteq \mathbb{R}[\mathbf{D}]$ 中， $\dim(U) = \mathbf{M}$ ，使得

$$\tilde{\mathbf{x}} = \sum_{(\mathbf{m}=1 \text{ to } \mathbf{M})} \mathbf{z}[\mathbf{m}] \mathbf{b}[\mathbf{m}] \in U \subseteq \mathbb{R}[\mathbf{D}] \quad (10.27)$$

尽可能与 $\mathbf{x}$ 相似。注意此时我们需要假设 $\tilde{\mathbf{x}}$ 的坐标 $\mathbf{z}[\mathbf{m}]$ 和 $\mathbf{x}$ 的坐标 $\zeta[\mathbf{m}]$ 不相同。

在下文中，我们确切地使用 $\tilde{\mathbf{x}}$ 的这种表示来找到最优坐标 $\mathbf{z}$ 和基向量 $\mathbf{b}_1, \dots, \mathbf{b}[\mathbf{M}]$ ，使得 $\tilde{\mathbf{x}}$ 与原始数据点 $\mathbf{x}$ 尽可能相似，即，我们目标是最小化 (欧几里得) 距离 $\|\mathbf{x} - \tilde{\mathbf{x}}\|$ 。图 10.7 说明了这种设置。

不失一般性，我们假设数据集 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}[\mathbf{N}]\}$ ， $\mathbf{x}[\mathbf{n}] \in \mathbb{R}[\mathbf{D}]$ ，以 0 为中心，即， $E\mathbf{X} = 0$ 。如果没有零均值假设，我们会得到完全相同的解，但符号会变得相当复杂。

我们感兴趣找到 $\mathbf{X}$ 到低维子空间 $U \subseteq \mathbb{R}[\mathbf{D}]$ 的最佳线性投影，其中 $\dim(U) = \mathbf{M}$ 且正交标准基向量为 $\mathbf{b}_1, \dots, \mathbf{b}[\mathbf{M}]$ 。我们将这个子空间 U 称为 *主子空间*。

数据点的投影表示为

$$\tilde{\mathbf{x}}[\mathbf{n}] := \sum_{(\mathbf{m}=1 \text{ to } \mathbf{M})} \mathbf{z}[\mathbf{mn}] \mathbf{b}[\mathbf{m}] = \mathbf{B} \mathbf{z}[\mathbf{n}] \in \mathbb{R}[\mathbf{D}], \quad (10.28)$$

其中 $\mathbf{z}[n] := [z_1[n], \dots, z[M][n]]^T \in \mathbb{R}[M]$ 是 $\tilde{\mathbf{x}}[n]$ 相对于基 $(\mathbf{b}_1, \dots, \mathbf{b}[M])$ 的坐标向量。更具体地说，我们有兴趣让 $\tilde{\mathbf{x}}[n]$ 与 $\mathbf{x}[n]$ 尽可能相似。

我们在下文中使用的相似性度量是 $\mathbf{x}$ 和 $\tilde{\mathbf{x}}$ 之间的平方距离(欧几里得范数) $\|\mathbf{x} - \tilde{\mathbf{x}}\|^2$ 。因此我们定义目标为最小化平均平方欧几里得距离(重构误差)(Pearson, 1901)

$$J[M] := (1/N) \sum_{n=1 \text{ to } N} \|\mathbf{x}[n] - \tilde{\mathbf{x}}[n]\|^2, \quad (10.29)$$

其中我们明确表示投影数据的子空间维度为 M 。为了找到这个最优线性投影，我们需要找到主子空间的正交标准基和投影相对于这个基的坐标 $\mathbf{z}[n] \in \mathbb{R}[M]$ 。

为了找到坐标 $\mathbf{z}[n]$ 和主子空间的 ONB，我们采用两步方法。首先，对于给定的 ONB $(\mathbf{b}_1, \dots, \mathbf{b}[M])$ ，我们优化坐标 $\mathbf{z}[n]$ ；其次，我们找到最优的 ONB。

10.3.2 寻找最优坐标

让我们首先找到投影 $\tilde{\mathbf{x}}[n]$ 的最优坐标 $z_1[n], \dots, z[M][n]$ ，其中 $n = 1, \dots, N$ 。考虑图 10.7(b)，其中主子空间由单个向量 $\mathbf{b}$ 张成。从几何角度讲，找到最优坐标 $\mathbf{z}$ 对应于找到线性投影 $\tilde{\mathbf{x}}$ 相对于 $\mathbf{b}$ 的表示，该表示最小化 $\tilde{\mathbf{x}} - \mathbf{x}$ 之间的距离。从图 10.7(b) 可以清楚地看出，这将是正交投影，在下文中我们将确切地展示这一点。

我们假设 $U \subseteq \mathbb{R}[D]$ 的 ONB $(\mathbf{b}_1, \dots, \mathbf{b}[M])$ 。为了找到最优坐标

关于这个基，我们需要坐标 $\mathbf{z}[m]$ 的偏导数

$$\partial J[M] / \partial \mathbf{z}[m] = 0, \quad (10.30a)$$

$$\partial J[M] / \partial \mathbf{z}[m] = -(\mathbf{x}[n] - \tilde{\mathbf{x}}[n]) \cdot \mathbf{b}[m] / N, \quad (10.30b)$$

图 10.8 向量 $\mathbf{x} \in \mathbb{R}^2$ 到一维子空间的最优投影（接图 10.7）。(a) 对于某个 $\tilde{\mathbf{x}} \in U$ ，距离 $\|\mathbf{x} - \tilde{\mathbf{x}}\|$ ；(b) 最小化面板 (a) 中距离的向量 $\tilde{\mathbf{x}}$ 是其在 U 上的正交投影。投影 $\tilde{\mathbf{x}}$ 相对于张成 U 的基向量 $\mathbf{b}$ 的坐标是我们需要缩放 $\mathbf{b}$ 以“到达” $\tilde{\mathbf{x}}$ 的因子。

$$\partial J[M] / \partial \mathbf{z}[m] = \mathbf{z}[m] \cdot \mathbf{b}[m] - \mathbf{x}[n] \cdot \mathbf{b}[m] / N, \quad (10.30c)$$

对于 $i = 1, \dots, M$ ，使得我们得到

$$\partial J[M] / \partial \mathbf{z}[m] = \mathbf{z}[m] \cdot \mathbf{b}[m] - \mathbf{x}[n] \cdot \mathbf{b}[m] / N = 0, \quad (10.31a)$$

$$[\text{ONB}] \cdot \mathbf{z}[m] = \mathbf{x}[n] \cdot \mathbf{b}[m] / N, \quad (10.31b)$$

由于 $\mathbf{b}[i] \cdot \mathbf{b}[i] = 1$ 。将此偏导数设为 0 立即得到最优坐标

$$\mathbf{z}[m] = \mathbf{x}[n] \cdot \mathbf{b}[m] / N, \quad (10.32)$$

对于 $i = 1, \dots, M$ 和 $n = 1, \dots, N$ 。这意味着投影 $\tilde{\mathbf{x}}[n]$ 的最优坐标 $\mathbf{z}[n]$ 是原始数据点 $\mathbf{x}[n]$ 在由 $\mathbf{b}[i]$ 张成的一维子空间上的正交投影（见第 3.8 节）的坐标。因此：

$x[n]$ 的最优线性投影 $\tilde{x}[n]$ 是正交投影。 $\tilde{x}[n]$ 相对于基 $(b_1, \dots, b[M])$ 的坐标是 $x[n]$ 在主子空间上的正交投影的坐标。

给定目标函数 (10.29)，正交投影是最佳线性映射。

(10.26) 中 x 的坐标 $\zeta[m]$ 和 (10.27) 中 $\tilde{x}$ 的坐标 $z[m]$ 对于 $m = 1, \dots, M$ 必须相同，因为 $U^\perp = \text{span}[b[M+1], \dots, b[D]]$ 是 $U = \text{span}[b_1, \dots, b[M]]$ 的正交补（见第 3.6 节）。

备注（具有标准正交基向量的正交投影）。 让我们简要回顾第 3.8 节中的正交投影。如果 $(b_1, \dots, b[D])$ 是 $\mathbb{R}^D$ 的标准正交基，那么

$$\tilde{x} = b[j] (b[j]^\top b[j])^{-1} b[j]^\top x = b[j] b[j]^\top x \in \mathbb{R}^D \quad (10.33)$$

是 x 在由第 j 个基向量张成的子空间上的正交投影，而 $z[j] = b[j]^\top x$ 是此投影相对于张成该子空间的基向量 $b[j]$ 的坐标，因为 $z[j] b[j] = \tilde{x}$ 。图 10.8(b) 说明了这种设置。

更一般地，如果我们想要投影到 $\mathbb{R}^D$ 的 M 维子空间上，我们得到 x 在具有标准正交基向量 $b_1, \dots, b[M]$ 的 M 维子空间上的正交投影为

$$\tilde{x} = B (B^\top B)^{-1} B^\top x = B B^\top x, \quad (10.34)$$

其中我们定义了 $B := [b_1, \dots, b[M]] \in \mathbb{R}^{D \times M}$ 。此投影相对于有序基 $(b_1, \dots, b[M])$ 的坐标是 $z := B^\top x$ ，如第 3.8 节所讨论的。

我们可以将坐标视为投影向量在由 $(b_1, \dots, b[M])$ 定义的新坐标系中的表示。注意，尽管 $\tilde{x} \in \mathbb{R}^D$ ，我们只需要 M 个坐标 $z_1, \dots, z[M]$ 来表示这个向量；相对于基向量 $(b[M+1], \dots, b[D])$ 的其他 $D - M$ 个坐标总是 0。◇

到目前为止，我们已经展示了对于给定的 ONB，我们可以通过正交投影到主子空间来找到 $\tilde{x}$ 的最优坐标。接下来，我们将确定什么是最佳基。

10.3.3 寻找主子空间的基

为了确定主子空间的基向量 $b_1, \dots, b[M]$ ，我们使用迄今为止的结果重新表述损失函数 (10.29)。这将使寻找基向量变得更容易。为了重新表述损失函数，我们利用之前的结果并得到

$$\tilde{x}[n] = \sum_{m=1}^M z[mn] b[m] = \sum_{m=1}^M (x[n]^T b[m]) b[m]. \quad (10.35)$$

我们现在利用点积的对称性，得到

$$\tilde{x}[n] = \sum_{m=1}^M b[m] b[m]^T x[n]. \quad (10.36)$$

图 10.9 正交投影和位移向量。当将数据投影

到子空间 U 上，我们得到 $\tilde{x}[n]$

。位移向量

$x[n] - \tilde{x}[n]$ 完全位于

—

补空间 $U^\perp$ (正交

的

$U^\perp$ 的

由于我们通常可以将原始数据点 $x[n]$ 写成所有基向量的线性组合，有

$x[n] = \sum_{d=1}^D d[n] b[d]$

$$x[n] = \sum_{d=1}^D d[n] b[d] = \sum_{d=1}^M d[n] b[d] + \sum_{d=M+1}^D d[n] b[d] \quad (10.37a)$$

其中 $d[n]$ 是标量。

将 $x[n]$ 代入 (10.36) 得到

$\tilde{x}[n] = \sum_{m=1}^M b[m] b[m]^T \sum_{d=1}^D d[n] b[d]$

$$= \sum_{m=1}^M \sum_{d=1}^D d[n] b[m] b[m]^T b[d] = \sum_{m=1}^M d[mn] b[m] \quad (10.37b)$$

其中 $d[mn] = b[m]^T b[m]$

将 $d[mn]$ 代入 (10.37b) 得到

其中我们将有 D 项的和分解为 M 项的和与 $D - M$ 项的和。根据这个结果，我们发现位移向量 $x[n] - \tilde{x}[n]$ ，即原始数据点与其投影之间的差向量，为

$[D]!$

$$\mathbf{x}^T \mathbf{X} [\tau] - \tilde{\mathbf{x}}^T = \mathbf{b}^T \mathbf{b} \mathbf{x} \quad (10.38a) \quad [n] \quad [n] \quad [j] \quad [j] \quad [n]$$

$$[j]=[M] [+1]$$

$[D]$

$$= \mathbf{X} [\tau] (\mathbf{x}^T \mathbf{b}) \mathbf{b}^T \quad (10.38b)$$

$$[n] \quad [j] \quad [j]$$

$$[j]=[M][+1]$$

这意味着该差向量恰好是数据点在主子空间正交补上的投影：我们将 $[(10.38a)]$ 中的矩阵 $\mathbf{P} [D] [\tau]$ 识别为执行此投影的投影矩阵 $\mathbf{b} \mathbf{b}^T$

$$[j]=[M] [+1] [j] [j]$$

。因此位移向量 $\mathbf{x}[n] - \tilde{\mathbf{x}}[n]$ 位于与主子空间正交的子空间中，如图 [10.9] 所示。

备注（低秩近似）。在 $[(10.38a)]$ 中，我们看到将 $\mathbf{x}$ 投影到 $\tilde{\mathbf{x}}$ 的投影矩阵为

$[M]$

$$\mathbf{X} [\tau] [\tau] \mathbf{b} \mathbf{b}^T \quad (10.39) = \mathbf{B} \mathbf{B}^T.$$

$$[m] \quad [m]$$

$$[m]=[1]$$

通过作为秩一矩阵 $[[\tau] [[\tau]] \mathbf{b} \mathbf{b}^T]$ 的和的构造，我们看到 $\mathbf{B} \mathbf{B}^T$ 是

$$[m] \quad [m]$$

[《机器学习数学》草稿 (2023-10-18)。反馈：] [<https://mml-book.com>][.] [10.3 投影视角] [331]

对称的且具有秩 M 。因此，平均平方重构误差也可以写成

$$\|\mathbf{x}[n] - \tilde{\mathbf{x}}[n]\|^2 = [\tau] \mathbf{x} [n] - \mathbf{B} \mathbf{B}^T \mathbf{x} [n] \quad (10.40a) \quad N \quad 1 \quad [N] \quad [N] \quad X \quad [2] \quad X \quad [2] \quad 1$$

$$[n]=[1] \quad [n]=[1]$$

$$1 \quad [N]$$

$$= [\tau] X \quad [2] \quad (I \quad (10.40b) - \mathbf{B} \mathbf{B}^T) \mathbf{x}[n].$$

$$N \quad [n]=[1]$$

寻找正交标准基向量 $b_1, \dots, b_M$, 使原始数据 $x[n]$ 与其投影 $\tilde{x}[n]$ 之间的差最小, 等价于寻找单位矩阵 I 的最佳秩- M 近似 $[[\tau]]$ $[[\text{近似}]] BB$

$[[\text{PCA 找到最佳}]]$ $[\text{秩-}] [M]$ $[[\text{单位矩阵的}]]$

(见第 4.6 节)。 $\diamond$

现在我们有重新表述损失函数 $[(10.29)]$ 的所有工具。

$[N]$ $[2]$

$$\|X X^T [M] - \tilde{x}[n] - \tilde{x}[n]\| = (b x[n]) b[j] \cdot (10.41) [j] N^{-1} [N] [D] X [2] [(10.38b)] 1$$

$[n] N^{-1} [n] [1] [j] [1] [M] [+1]$

现在我们显式计算平方范数并利用 $b[j]$ 构成 ONB 的事实, 得到

$$1 [N] [D] [N] [D] 1$$

$$X X^T [X X^T] [X X^T] [X X^T] = (b x) = b x b x (10.42a)$$

$$[M] [j] [n] [j] [n] [j] [n] N N$$

$$[n] [1] [j] [1] [M] [+1] [n] [1] [j] [1] [M] [+1]$$

$$1 [N] [D]$$

$$= X X^T [X X^T] b x x b, (10.42b)$$

$$N [j] [n] [n] [j]$$

$$[n] [1] [j] [1] [M] [+1]$$

其中在最后一步我们利用了点积的对称性来写成 $[X X^T] b x = x b$ 。现在我们交换求和顺序并得到

$$[j] [n] [n] [j]$$

$$[D] ! [N] [D] 1$$

$$X X X X^T [X X^T] [X X^T] = b x x b = b S b (10.43a)$$

$$[M] [j] [n] [n] [j] [j] [j] N$$

$$[j] [1] [M] [+1] [n] [1] [j] [1] [M] [+1]$$

$$| \{z\} |$$

$$[1:] [S]$$

$$[D] [D] [D]$$

$$= \mathbf{X}^T \mathbf{X} \mathbf{X}^T [\mathbf{T}] [\mathbf{T}]^T \text{tr}(\mathbf{b} \text{tr} \mathbf{S} \mathbf{b}) = \text{tr}(\mathbf{S} \mathbf{b} \mathbf{b}^T) = \mathbf{b}^T \mathbf{S} \mathbf{b},$$

$$[j] [j] [j] [j] [j]$$

$$[j] [=] [\mathbf{M}] [+1] \quad [j] [=] [\mathbf{M}] [+1] \quad [j] [=] [\mathbf{M}] [+1]$$

$$|\{z\}$$

$$[\text{投影矩阵}]$$

$$(10.43b)$$

其中我们利用了迹算子 $\text{tr}(\cdot)$ (见 [(4.18)]) 是线性的且对其参数的循环置换不变的性质。由于我们假设数据集是中心化的, 即 $E[\mathbf{X}] = 0$, 我们将 $\mathbf{S}$ 识别为数据协方差矩阵。由于 [(10.43b)] 中的投影矩阵被构造为秩一矩阵 $[\mathbf{T}] \mathbf{b} \mathbf{b}^T$ 的和, 它本身具有秩 $D - M$ 。

$$[j] [j]$$

方程 [(10.43a)] 表明我们可以等价地将平均平方重构误差表述为数据的协方差矩阵,

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由 Cambridge University Press 于 2020 年出版。]

[332] [[主成分分析的降维]]

投影到主子空间的正交补上。因此, 最小化平均平方重构误差等价于最小化数据投影到我们忽略的子空间 (即主子空间的正交补) 上时的方差。等价地, 我们最大化在主子空间中保留的投影的方差, 这将投影损失直接链接到第 [10.2] 节讨论的 PCA 的最大方差表述。但这也意味着我们将得到与最大方差视角所得到的相同解。因此, 我们略去与第 [10.2] 节中相同的推导, 并从投影视角的角度总结前面的结果。

当投影到 M 维主子空间时, 平均平方重构误差为

[最小化平均平方][重构误差等价于][最小化数据协方差][矩阵投影到主子空间][正交补上。][最小化平均平方][重构误差等价于][最大化投影数据的][方差。]

$$[\mathbf{D}]$$

$$\mathbf{J} \mathbf{X} = \lambda, (10.44)$$

$$[\mathbf{M}] [j]$$

$$[j] [=] [\mathbf{M}] [+1]$$

其中 $\lambda[j]$ 是数据协方差矩阵的特征值。因此, 为了最小化 [(10.44)], 我们需要选择最小的 $D - M$ 个特征值, 这意味着它们对应的特征向量是主子空间正交补的基。因此, 这意味着主子空间的基包含特征向量

$\mathbf{b}_1, \dots, \mathbf{b}_M$ 是与数据协方差矩阵最大 M 个特征值相关联的特征向量。

示例 10.3 (MNIST 数字嵌入)

图 10.10



[MNIST 数字嵌入] [0]



[(蓝色) 和] 1



[(橙色) 在一个]



[二维]

[主子空间中]

[使用PCA。四个]



[数字” 0” 和” 1” 的嵌入在]



[主子空间中以红色突出显示]



[以及它们]

[对应的]



[原始数字。]

图10.10可视化了MNIST数字”0”和”1”的训练数据，这些数据嵌入在由前两个主成分张成的向量空间中。我们观察到”0”（蓝点）和”1”（橙点）之间相对清晰的分离，并且可以看到每个单独簇内的变化。

图中以红色突出显示了数字”0”和”1”在主子空间中的四个嵌入，并显示了它们对应的原始数字。该图显示”0”集合内的变化明显大于”1”集合内的变化。

10.4 特征向量计算和低秩近似

在前面的章节中，我们获得了主子空间的基，它是与数据协方差矩阵最大特征值相关联的特征向量：

$$\frac{1}{N} S = X^T X = \sum_{n=1}^N x_n x_n^T, \quad (10.45)$$

$$X = [x_1, \dots, x_N] \in \mathbb{R}^{(D \times N)}. \quad (10.46)$$

注意 X 是一个 $D \times N$ 矩阵，即它是“典型”数据矩阵的转置(Bishop, 2006; Murphy, 2012)。为了获得 S 的特征值(和对应的特征向量)，我们可以采用两种方法：

使用特征分解或SVD来计算特征向量。

我们执行特征分解(参见第4.2节)并直接计算 S 的特征值和特征向量。

我们使用奇异值分解(参见第4.5节)。由于 S 是对称的且可以分解为 $\frac{1}{N} X X^T$ (忽略因子 $\frac{1}{N}$)， S 的特征值是 X 的奇异值的平方。

更具体地， X 的SVD为：

$$X = U \Sigma V^T, \quad (10.47) \quad \begin{matrix} \{z\} & \{z\} & \{z\} & \{z\} \\ D \times N & N \times D & D \times D & N \times N \end{matrix}$$

其中 $U \in \mathbb{R}^{(D \times D)}$ 和 $V \in \mathbb{R}^{(N \times N)}$ 是正交矩阵， $\Sigma \in \mathbb{R}^{(D \times N)}$ 是一个矩阵，其唯一非零项是奇异值 $\sigma_{ii} \geq 0$ 。因此有：

$$S = \frac{1}{N} X X^T = \frac{1}{N} U \Sigma V^T V \Sigma^T U^T = \frac{1}{N} U \Sigma \Sigma^T U^T. \quad (10.48) \quad \{z\} = IN$$

根据第4.5节的结果，我们得到 U 的列是 $X X^T$ (因此也是 S)的特征向量。此外， S 的特征值 λ_d 与 X 的奇异值通过以下关系相关联：

$$\lambda_d = \sigma_d^2 / N. \quad (10.49)$$

S 的特征值与 X 的奇异值之间的这种关系提供了最大方差观点(第10.2节)与奇异值分解之间的联系。

10.4.1 使用低秩矩阵近似的PCA

为了最大化投影数据的方差(或最小化平均平方重构误差)，PCA选择(10.48)中 U 的列作为与数据协方差矩阵 S 的 M 个最大特征值相关联的特征向量，这样我们将 U 识别为(10.3)中的投影矩阵 B ，它将原始数据投影到维度为 M 的低维子空间上。**Eckart-Young定理**(第4.6节中的定理4.25)提供了估计低维表示的直接方法。考虑最佳秩- M 近似：

$$\tilde{X}_M := \operatorname{argmin}_{\tilde{X} \in \mathbb{R}^{(D \times N)}} \|X - \tilde{X}\|_2 \quad (10.50) \quad \operatorname{rk}(\tilde{X}) \leq M$$

其中 $\|\cdot\|_2$ 是在(4.93)中定义的谱范数。Eckart-Young定理指出 $\tilde{X}_M$ 是通过在前 M 个奇异值处截断SVD得到的。换句话说，我们得到：

$$\tilde{X}M^T = UM\Sigma MVM^T \in \mathbb{R}^{(D \times N)} \quad (10.51) \quad \{z\} \mid \{z\} \mid \{z\} \quad D \times M \quad M \times M \quad M \times N$$

其中正交矩阵 $UM := [u_1, \dots, u_M] \in \mathbb{R}^{(D \times M)}$ 和 $VM := [v_1, \dots, v_M] \in \mathbb{R}^{(N \times M)}$ ，以及对角矩阵 $\Sigma_M \in \mathbb{R}^{(M \times M)}$ ，其对角项是 X 的 M 个最大奇异值。

10.4.2 实际考虑

寻找特征值和特征向量在其他需要矩阵分解的基础机器学习方法中也很重要。理论上，正如我们在第4.2节中讨论的，我们可以将特征值作为特征多项式的根来求解。然而，对于大于 4×4 的矩阵，这是不可能的，因为我们需要找到5次或更高次多项式的根。然而，**Abel-Ruffini定理**(Ruffini, 1799; Abel, 1826)指出，对于5次或更高次的多项式，不存在这个问题的代数解。因此，在实践中，我们使用迭代方法求解特征值或奇异值，这些方法在所有现代线性代数包中都有实现(如 `np.linalg.eigh` 或 `np.linalg.svd`)。

在许多应用中(如本章介绍的PCA)，我们只需要少数几个特征向量。计算完整分解然后丢弃前几个之外所有特征值对应的特征向量是浪费的。事实证明，如果我们只对前几个特征向量(具有最大特征值)感兴趣，那么迭代过程

直接优化这些特征向量，在计算上比完整的特征分解（或SVD）更高效。在只需要第一个特征向量的极端情况下，一种称为 **幂迭代** 的简单方法非常高效。幂迭代选择一个不在 S 的零空间中的随机向量 $x[0]$ ，然后按照以下迭代：

$$Sx[k] \quad x[k+1] = \frac{Sx[k]}{\|Sx[k]\|}, \quad k = 0, 1, \dots \quad (10.52) \quad \|Sx[k]\|$$

这意味着向量 $x[k]$ 在每次迭代中都被 S 相乘，然后被归一化，即我们总是有 $\|x[k]\| = 1$ 。如果 S 是可逆的，只需确保 $x[0] \neq 0$ 就足够了。这个向量序列收敛到与 S 的最大特征值相关联的特征向量。原始的 **Google PageRank** 算法 (Page et al., 1999) 使用这样的算法基于超链接对网页进行排名。

10.5 高维度中的 PCA

为了执行 PCA，我们需要计算数据协方差矩阵。在 D 维度中，数据协方差矩阵是一个 $D \times D$ 矩阵。计算这个矩阵的特征值和特征向量在计算上是昂贵的，因为它与 D 成三次方比例缩放。因此，如我们之前讨论的，PCA 在非常高的维度中将是不可行的。例如，如果我们的 $x[n]$ 是具有 10,000 个像素的图像（例如， 100×100 像素图像），我们需要计算一个 $10,000 \times 10,000$ 协方差矩阵的特征分解。在下文中，我们为数据点数量远少于维度的情况（即 $N \ll D$ ）提供了解决这个问题的方案。

假设我们有一个中心化的数据集 $x_1, \dots, x[N]$, $x[n] \in \mathbb{R}^D$ 。那么数据协方差矩阵为：

$$S = (1/N)XX^T \in \mathbb{R}^{(D \times D)}, \quad (10.53)$$

其中 $X = [x_1, \dots, x[N]]$ 是一个 $D \times N$ 矩阵，其列是数据点。

我们现在假设 $N \ll D$ ，即数据点的数量小于数据的维度。如果没有重复的数据点，协方差矩阵 S 的秩是 N ，所以它有 $D - N + 1$ 个值为 0 的特征值。直观上，这意味着存在一些冗余。在下文中，我们将利用这一点，将 $D \times D$ 协方差矩阵转换为一个所有特征值都为正的 $N \times N$ 协方差矩阵。

在 PCA 中，我们得到特征向量方程：

$$Sb[m] = \lambda[m]b[m], \quad m = 1, \dots, M, \quad (10.54)$$

其中 $b[m]$ 是主子空间的基向量。让我们稍微重写这个方程：使用在 (10.53) 中定义的 S ，我们得到：

$$Sb[m] = (1/N)XX^Tb[m] = \lambda[m]b[m]. \quad (10.55)$$

我们现在从左侧乘以 $X^T \in \mathbb{R}^{(N \times D)}$ ，得到：

$$(1/N)X^TXX^Tb[m] = \lambda[m]X^Tb[m] \Leftrightarrow (1/N)X^TXc[m] = \lambda[m]c[m], \quad (10.56) \quad \text{--- } N \times N =: c[m]$$

我们得到一个新的特征向量/特征值方程： $\lambda[m]$ 仍然是特征值，这证实了我们在第 4.5.3 节的结果，即 XX^T 的非零特征值等于 X^TX 的非零特征值。我们得到与 $\lambda[m]$ 相关联的矩阵 $(1/N)X^TX \in \mathbb{R}^{(N \times N)}$ 的特征向量为：

$c[m] := X^Tb[m]$ 。假设我们没有重复的数据点，这个矩阵的秩为 N 且是可逆的。这也意味着 $(1/N)X^TX$ 具有与数据协方差矩阵 S 相同的（非零）特征值。但这现在是一个 $N \times N$ 矩阵，所以我们可以比原始的 $D \times D$ 数据协方差矩阵更高效地计算特征值和特征向量。

现在我们有 $(1/N)X^TX$ 的特征向量，我们将恢复原始的特征向量，这些对于 PCA 仍然需要。目前，我们知道 $(1/N)X^TX$ 的特征向量。如果我们用 X 左乘我们的特征值/特征向量方程，我们得到：

$$(1/N)XX^TXc[m] = \lambda[m]Xc[m] \quad (10.57) \quad \text{--- } S$$

我们再次恢复数据协方差矩阵。这现在也意味着我们恢复了 $Xc[m]$ 作为 S 的特征向量。

备注。如果我们想应用在第 10.6 节中讨论的 PCA 算法，我们需要将 S 的特征向量 $Xc[m]$ 归一化，使它们的范数为 1。◇

10.6 PCA 实践中的关键步骤

在下文中，我们将通过一个运行示例来详细介绍 PCA 的各个步骤，这在图 10.11 中总结。我们给定一个二维数据集（图 10.11(a)），我们想使用 PCA 将其投影到一维子空间。

1. **均值减法** 我们首先通过计算数据集的均值 μ 并从每个数据点中减去它来中心化数据。这确保数据集的均值为 0（图 10.11(b)）。均值减法不是严格必要的，但减少了数值问题的风险。
2. **标准化** 将数据点除以数据集在每个维度 $d = 1, \dots, D$ 上的标准差 $\sigma[d]$ 。现在数据是无单位的，并且沿每个轴的方差为 1，这由图 10.11(c) 中的两个箭头表示。这一步完成了数据的**标准化**。
3. **协方差矩阵的特征分解** 计算数据协方差矩阵及其特征值和相应的特征向量。由于协方差矩阵是对称的，谱定理（定理 4.15）表明我们可以找到特征向量的 ONB。在图 10.11(d) 中，特征向量按相应特征值的大小进行缩放。

图 10.11 PCA 的步骤。(a) 原始数据集；

$x \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \end{bmatrix}$ (b) 居中; $\begin{bmatrix} 2 \\ 2 \end{bmatrix} x \begin{bmatrix} 5 \\ 5 \end{bmatrix}$ (c) 除以 $\begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

[标准差;]

$\begin{bmatrix} -1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \end{bmatrix}$ (d) 特征值分解-

$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$; (e) 投影; $\begin{bmatrix} 5 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \end{bmatrix}$

$x \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} x$ (f) 映射回]

[(a) 原始数据集。] [[原始数据空间。] [(b) 步骤1：通过从每个] [(c) 步骤2：除以标准差]]

[数据点减去均值进行居中。] [[使数据变为无量纲。数据在] [每个轴上的方差为1。]

$5 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 5 \\ 1 \end{bmatrix} \begin{bmatrix} 5 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix}$

$\begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \end{bmatrix}$

$x \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} x$

$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \end{bmatrix}$

$\begin{bmatrix} -1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \end{bmatrix}$

$\begin{bmatrix} 0 \\ 5 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \end{bmatrix}$

$x \begin{bmatrix} 2 \\ 2 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} x$

[(d) 步骤3：计算特征值] [(e) 步骤4：将数据投影到] [(f) 撤销标准化并将投影数据] [和特征向量（箭头）] [主子空间上。] [移回到原始数据空间] [数据协方差矩阵的] [从(a)中。]

对应的特征值。较长的向量跨越主子空间，我们用 U 表示。数据协方差矩阵由椭圆表示。

投影

我们可以将任何数据点 $\mathbf{x} \in \mathbb{R}^D$ 投影到主子空间：为了正确做到这一点，我们需要使用训练数据在第 d 维的均值 $\mu[d]$ 和标准差 $\sigma[d]$ 对 $\mathbf{x}$ 进行标准化，使得

$$x[d] \leftarrow \frac{x[d] - \mu[d]}{\sigma[d]}, d = 1, \dots, D, \quad (10.58)$$

其中 $x[d]$ 是 $\mathbf{x}$ 的第 d 个分量。我们获得投影为

$$\tilde{\mathbf{x}} = \mathbf{B}\mathbf{x} \quad (10.59)$$

坐标为

$$\mathbf{z} = \mathbf{B}\mathbf{x} \quad (10.60)$$

相对于主子空间的基。这里， $\mathbf{B}$ 是包含与数据协方差矩阵最大特征值相关的特征向量作为列的矩阵。PCA 返回坐标 $\mathbf{z}$ 而不是投影 $\tilde{\mathbf{x}}$ 。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

主成分分析的降维

标准化我们的数据集后，[(10.59)]只在标准化数据集的上下文中产生投影。为了在原始数据空间（即标准化之前）中获得我们的投影，我们需要撤销标准化[(10.58)]并在加上均值之前乘以标准差，使得我们获得

$$\mathbf{x}^{(d)}_{\sigma} \leftarrow \mathbf{x}^{(d)}_{\mu} \cdot \mathbf{D}^{(10.61)} + \mathbf{d}^{(10.61)}, d = 1, \dots, D$$

图[10.11(f)]显示了在原始数据空间中的投影。

示例10.4 (MNIST数字：重构)

接下来，我们将PCA应用于MNIST数字数据集，该数据集包含60,000个手写数字0到9的样本。每个数字是一个 28×28 大小的图像，即包含784个像素，因此我们可以将此数据集中的每个图像解释为向量 $\mathbf{x} \in \mathbb{R}^{784}$ 。这些数字的示例显示在图[10.3]中。

图10.12 [效果]



[增加主成分][原始][数量对重构的影响。]



[PC: 1]



[PC: 10]



[PC: 100]



[PC: 500]



为了说明目的，我们将PCA应用于MNIST数字的一个子集，



并且我们专注于数字”8”。我们使用了5,389个数字”8”的训练图像，



并按照本章详述的方法确定了主子空间。然后我们使用学习到的



投影矩阵来重构一组测试图像，如图[10.12]所示。图[10.12]的第一行



显示了测试集中四个原始数字的集合。下面的行显示了当使用



维度为1、10、100和500的主子空间时这些确切数字的重构。



我们看到即使使用一维主子空间，我们也能得到原始数字相当



不错的重构，但是模糊且通用。随着主成分(PC)数量的增加，



重构变得更清晰，更多细节得到体现。使用500个主成分，



我们有效地获得了近乎完美的重构。如果我们选择784个PC，



我们将恢复精确的数字而没有任何压缩损失。





图[10.13]显示了平均平方重构误差，即

$$\frac{1}{N} \sum_{n=1}^N \| \mathbf{x}^{(n)} - \tilde{\mathbf{x}}^{(n)} \|^2 = \lambda, \quad (10.62)$$

作为主成分数量M的函数。我们可以看到主成分的重要性迅速下降，通过添加更多PC只能获得边际收益。这完全符合我们在图[10.5]中的观察，我们发现投影数据的大部分方差仅由少数几个主成分捕获。使用大约550个PC，我们基本上可以完全重构包含数字”8”的训练数据（边界周围的一些像素没有变化，因为它们始终是黑色的）。

图10.13

[误差] [500] [平均平方]

[重构误差]

[400]

[作为主成分]

[数量的函数。]

[300]

[重构的]

[平均平方]

[200]

[重构误差]

[平方] [[是正交] [100]]

[特征值的和]

[平均] [0]

[Av] [[complement of the] [0] [200] [400] [600] [800]]

[PC数量] [[主子空间。]]

10.7 潜变量视角

在前面的章节中，我们没有使用任何概率模型的概念来推导PCA，而是使用了最大方差和投影视角。一方面，这种方法可能很有吸引力，因为它允许我们回避概率理论带来的所有数学困难，但另一方面，概率模型将为我们提供更多的灵活性和有用的见解。更具体地说，概率模型将

配备一个似然函数，我们可以明确处理噪声观测（我们之前甚至没有讨论过）

允许我们通过边际似然进行贝叶斯模型比较，如第[8.6]节所讨论的

将PCA视为生成模型，这允许我们模拟新数据

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[340] [[主成分分析降维]]

允许我们与相关算法建立直接联系通过应用贝叶斯定理处理随机缺失的数据维度

给我们一个新数据点新颖性的概念

给我们一个有原则的方式来扩展模型，例如，扩展到PCA模型的混合

将我们在前面章节中推导的PCA作为特殊情况允许通过边际化模型参数进行完全贝叶斯处理

通过引入连续值潜变量[M] $z \in \mathbb{R}$ ，可能将PCA表述为概率潜变量模型。[Tipping and Bishop] [概率PCA] [(1999)]提出了这种潜变量模型作为 *概率PCA(PPCA)*。[PPCA] [PPCA解决了上述大部分问题，我们通过最大化投影空间中的方差或通过最小化重构误差获得的PCA解]

是在无噪声设置中最大似然估计的特殊情况。

10.7.1 生成过程和概率模型

在PPCA中，我们明确写出线性降维的概率模型。为此我们假设一个连续潜变量

$z \in [M] \mathbb{R}$ 具有标准正态先验 $p(z) = N(0, I)$ 和潜变量与观测 x 数据之间的线性关系，其中

$$x[D] = Bz + \mu + \epsilon \in \mathbb{R}, \quad (10.63)$$

其中 $\epsilon \sim N([D] \times [M]) (0, \sigma^2 I)$ 是高斯观测噪声， $B \in \mathbb{R}$

和 $[D] \mu \in \mathbb{R}$ 描述从潜变量到观测变量的线性/仿射映射。因此，PPCA通过以下方式连接潜变量和观测变量：

$$p([D] \times [M] (x | z, B, \mu, \sigma)) = N(x | Bz + \mu, \sigma^2 I). \quad (10.64)$$

总体上，PPCA诱导以下生成过程：

$$z[n] \sim N(z | 0, I) \quad (10.65)$$

$$x[2] | z \sim N(x | Bz + \mu, \sigma^2 I) \quad [n] [n] [n] \quad (10.66)$$

为了生成在给定模型参数下典型的数据点，我们[祖先采样]遵循祖先采样方案：我们首先从 $p(z)$ 中采样一个潜变量 $z[n]$ 。然后我们在[(10.64)]中使用 $z[n]$ 来采样一个条件于

采样的 $z[n]$ 的数据点，即 $x[n] \sim [2] p(x | z[n], B, \mu, \sigma)$ 。

这个生成过程允许我们写出概率模型（即，所有随机变量的联合分布；见第[8.4]节）为

$$p(z|x,z|B,\mu,\sigma)=p(x|z,B,\mu,\sigma)p(z), (10.67)$$

这立即给出了图[10.14]中的图形模型，使用第[8.5]节的结果。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>][.][10.7 潜变量视角][341]

[图10.14]

[z][n] [[概率PCA的]]

[图形模型。]

[观测] [x][n]

[B] [μ] [明确依赖于]

[相应的]

x [[潜变量] [σ]]

[n] [[z] [n] [0] [~ N] [,]][.]

[模型参数]

[n] [= 1][, ... ,N]

[B][,][μ] [和] [似然]

[参数] [σ] [在]

备注。注意连接潜变量[[数据集中共享。]] z和观测数据x的箭头方向：箭头从z指向x，这意味着PPCA模型假设高维观测x有一个低维潜在原因z。最终，我们显然有兴趣在给定一些观测的情况下找出关于z的一些信息。为了达到这个目标，我们将应用贝叶斯推理来隐式地”反转”箭头，从观测到潜变量。◇

例10.5（使用潜变量生成新数据）

[图10.15]



生成新的MNIST数字。潜变量z可以用来生成新数据



$\tilde{x} = Bz$ 。我们越接近训练数据，生成的数据就越逼真。



图[10.15]显示了使用二维主子空间时PCA找到的MNIST数字”8”的潜坐标（蓝点）。我们可以查询这个潜空间中的任何向量 $z[*]$ 并生成一个类似数字”8”的图像 $\tilde{x}[*] = Bz[*]$ 。我们展示了八个这样的生成图像及其相应的潜空间表示。根据我们查询潜空间的位置，生成的图像看起来不同（形状、旋转、大小等）。如果我们查询远离训练数据的地方，我们会看到越来越多的伪影，例如左上角和右上角的数字。注意这些生成图像的内在维度只有二。



[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[342] [[主成分分析降维]]

10.7.2 似然和联合分布

[似然确实]

不依赖于潜在变量 z 。利用第6章的结果，我们通过积分去除潜在变量 z 来获得这个概率模型的likelihood（见8.4.3节）

因此

z

$$p(\mathbf{x} | \mathbf{B}, \mu, \sigma^2) = p(\mathbf{x} | \mathbf{z}, \mathbf{B}, \mu, \sigma^2) p(\mathbf{z}) d\mathbf{z} \quad (10.68a)$$

$\mathbf{Z}$

$$= N(\mathbf{x} | \mathbf{B}\mathbf{z} + \mu, \sigma^2 \mathbf{I}) N(\mathbf{z} | 0, \mathbf{I}) d\mathbf{z} . \quad (10.68b)$$

从6.5节我们知道，这个积分的解是一个Gaussian分布，其均值为

$$E_{\mathbf{x}} = E[\mathbf{B}\mathbf{z} + \mu] + E[\epsilon] = \mu \quad (10.69)$$

协方差矩阵为

$$\mathbf{V}_{\mathbf{x}} = \mathbf{V}[\mathbf{B}\mathbf{z} + \mu] + \mathbf{V}[\epsilon] = \mathbf{V}[\mathbf{B}\mathbf{z}] + \sigma^2 \mathbf{I} \quad (10.70a)$$

$$= \mathbf{B} \mathbf{V}[\mathbf{z}] \mathbf{B}^T + \sigma^2 \mathbf{I} = \mathbf{B} \mathbf{B}^T + \sigma^2 \mathbf{I} . \quad (10.70b)$$

(10.68b)中的likelihood可以用于模型参数的maximum likelihood或MAP估计。

注释. 我们不能使用(10.64)中的条件分布进行maximum likelihood估计，因为它仍然依赖于潜在变量。我们进行maximum likelihood（或MAP）估计所需的likelihood函数应该只是数据 $\mathbf{x}$ 和模型参数的函数，但不能依赖于潜在变量。◇

从6.5节我们知道，Gaussian随机变量 $\mathbf{z}$ 和它的线性/仿射变换 $\mathbf{x} = \mathbf{B}\mathbf{z}$ 是联合Gaussian分布的。我们已经知道边际分布 $p(\mathbf{z}) = N(\mathbf{z} | 0, \mathbf{I})$ 和 $p(\mathbf{x}) = N(\mathbf{x} | \mu, \mathbf{B} \mathbf{B}^T + \sigma^2 \mathbf{I})$ 。缺失的交叉协方差由下式给出

$$\text{Cov}[\mathbf{x}, \mathbf{z}] = \text{Cov}[\mathbf{B}\mathbf{z} + \mu, \mathbf{z}] = \mathbf{B} \text{Cov}[\mathbf{z}, \mathbf{z}] = \mathbf{B} . \quad (10.71)$$

因此，PPCA的概率模型，即潜在和观测随机变量的联合分布明确由下式给出

$$p(\mathbf{x}, \mathbf{z} | \mathbf{B}, \mu, \sigma^2) = N([\mathbf{x}/\mathbf{z}], [\mu/0], [\mathbf{B} \mathbf{B}^T + \sigma^2 \mathbf{I} \ \mathbf{B}/\mathbf{B}^T \ \mathbf{I}]) , \quad (10.72)$$

其中均值向量长度为 $D + M$ ，协方差矩阵大小为 $(D + M) \times (D + M)$ 。

10.7.3 后验分布

(10.72)中的联合Gaussian分布 $p(\mathbf{x}, \mathbf{z} | \mathbf{B}, \mu, \sigma^2)$ 允许我们通过应用6.5.1节的Gaussian条件化规则来立即确定后验分布 $p(\mathbf{z} | \mathbf{x})$ 。给定观测 $\mathbf{x}$ 的潜在变量的后验分布为

$$p(\mathbf{z} | \mathbf{x}) = N(\mathbf{z} | \mathbf{m}, \mathbf{C}) , \quad (10.73)$$

$$\mathbf{m} = \mathbf{B}^T (\mathbf{B} \mathbf{B}^T + \sigma^2 \mathbf{I})^{-1} (\mathbf{x} - \mu) , \quad (10.74)$$

$$\mathbf{C} = \mathbf{I} - \mathbf{B}^T (\mathbf{B} \mathbf{B}^T + \sigma^2 \mathbf{I})^{-1} \mathbf{B} . \quad (10.75)$$

注意后验协方差不依赖于观测数据 $\mathbf{x}$ 。对于数据空间中的新观测 $\mathbf{x}$ ，我们使用(10.73)来确定相应潜在变量 $\mathbf{z}$ 的后验分布。协方差矩阵 $\mathbf{C}$ 允许我们评估embedding的置信度。具有小行列式（测量体积）的协方差矩阵 $\mathbf{C}$ 告诉我们潜在embedding $\mathbf{z}$ 相当确定。如果我们获得一个具有很大方差的后验分布 $p(\mathbf{z} | \mathbf{x}^*)$ ，我们可能面临一个outlier。然而，我们可以探索这个后验分布来理解在这个后验下其他什么数据点 $\mathbf{x}$ 是合理的。为此，我们利用PPCA底层的生成过程，它允许我们通过生成在这个后验下合理的新数据来探索潜在变量上的后验分布：

1. 从潜在变量上的后验分布(10.73)中采样一个潜在变量 $z^* \sim p(z|x^*)$ 。

2. 从(10.64)中采样一个重构向量 $\tilde{x}^* \sim p(x|z^*, B, \mu, \sigma^2)$ 。

如果我们多次重复这个过程，我们可以探索潜在变量 z^* 上的后验分布(10.73)及其对观测数据的影响。采样过程有效地假设了在后验分布下合理的数据。

10.8 进一步阅读

我们从两个角度推导了PCA：(a) 最大化投影空间中的方差；(b) 最小化平均重构误差。然而，PCA也可以从不同的角度解释。让我们回顾一下我们所做的：我们取高维数据 $\mathbf{x} \in \mathbb{R}^D$ ，使用矩阵 $\mathbf{B}^T$ 找到低维表示 $\mathbf{z} \in \mathbb{R}^M$ 。 $\mathbf{B}$ 的列是与最大特征值相关联的数据协方差矩阵 $\mathbf{S}$ 的特征向量。一旦我们有了低维表示 $\mathbf{z}$ ，我们可以得到它的高维版本（在原始数据空间中）为 $\tilde{\mathbf{x}} = \mathbf{B}^T \mathbf{z} = \mathbf{B}^T \mathbf{B} \mathbf{x} \in \mathbb{R}^D$ ，其中 $\mathbf{B}^T \mathbf{B}$ 是一个投影矩阵。

我们也可以将PCA视为线性auto-encoder，如图10.16所示。auto-encoder将数据 $\mathbf{x}^{(n)} \in \mathbb{R}^D$ 编码为code $\mathbf{z}^{(n)} \in \mathbb{R}^M$ ，并将其解码为类似于 $\mathbf{x}^{(n)}$ 的 $\tilde{\mathbf{x}}^{(n)}$ 。从数据到code的映射称为encoder，从code回到原始数据空间的映射称为decoder。如果我们考虑线性映射，其中



图10.16 PCA可以被视为线性auto-encoder。它将高维数据 $\mathbf{x}$ 编码为低维表示（code） $\mathbf{z} \in \mathbb{R}^M$ ，并使用decoder解码 $\mathbf{z}$ 。解码向量 $\tilde{\mathbf{x}}$ 是原始数据 $\mathbf{x}$ 在 M 维主子空间上的正交投影。

图10.16 PCA可以被视为线性auto-encoder。它将高维数据 $\mathbf{x}$ 编码为低维表示（code） $\mathbf{z} \in \mathbb{R}^M$ ，并使用decoder解码 $\mathbf{z}$ 。解码向量 $\tilde{\mathbf{x}}$ 是原始数据 $\mathbf{x}$ 在 M 维主子空间上的正交投影。

code 由 $\mathbf{z}^{(n)} = \mathbf{B}^T \mathbf{x}^{(n)} \in \mathbb{R}^M$ 给出，我们感兴趣最小化数据 $\mathbf{x}^{(n)}$ 和其重构 $\tilde{\mathbf{x}}^{(n)} = \mathbf{B} \mathbf{z}^{(n)}$, $n = 1, \dots, N$ 之间的平均平方误差，我们得到

$$\frac{1}{N} \sum_{n=1}^N \|\mathbf{x}^{(n)} - \tilde{\mathbf{x}}^{(n)}\|^2 = \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}^{(n)} - \mathbf{B}^T \mathbf{B} \mathbf{x}^{(n)}\|^2. \quad (10.76)$$

这意味着我们最终得到与[(10.29)]中相同的目标函数，我们在第[10.3]节中讨论过，因此当我们最小化平方自编码损失时，我们得到PCA解。如果我们用非线性映射替换PCA的线性映射，我们就得到一个非线性自编码器(auto-encoder)。

这方面的一个突出例子是深度自编码器(deep auto-encoder)，其中线性函数被深度神经网络所替代。在这种情况下，编码器[recognition network]也被称为识别网络(recognition network)或推理网络(inference network)，而[inference network]解码器也被称为生成器(generator)。[generator] [PCA的另一种解释与信息论相关。我们可以]

将代码视为原始数据点的较小或压缩版本。当我们使用代码重构原始数据时，我们无法得到完全相同的数据点，而是得到一个略微失真或有噪声的版本[代码是][它的。这意味着我们的压缩是“有损的”。直观上，我们希望]

[原始数据的][最大化原始数据和低维]

[压缩版本。][代码之间的相关性。更正式地说，这与互信息(mutual information)有关。]

然后，我们将通过最大化互信息(mutual information)得到我们在第[10.3]节中讨论的PCA的相同解，互信息是信息论中的核心概念[(MacKay, 2003).]

在我们对PPCA的讨论中，我们假设模型的参数，即[2] $\mathbf{B}$ 、 μ 和似然参数 σ 是已知的。[Tipping]

[and Bishop (1999)]描述了如何在PPCA设置中推导这些参数的最大似然估计(注意我们在本章中使用不同的记号)。当将 D 维数据投影到 M 维子空间时，最大似然参数为

1 [N]

$$\mu^T X = x, \quad (10.77)$$

$$[ML] \in \mathbb{R}^{n \times N}$$

$$[n] = \{1, \dots, n\}$$

$$B \in \mathbb{R}^{2 \times 1} [ML] \in \mathbb{R}^{2 \times 2} = T (\Lambda - \sigma^2 I) R, \quad (10.78)$$

$$1 \in [D]$$

$$\sigma^2 [2] X = \lambda, \quad (10.79)$$

$$[ML] \in \mathbb{R}^{j \times D-M}$$

$$[j] = [M] + 1$$

其中 $T \in \mathbb{R}^{[D] \times [M]}$ 包含数据协方差矩阵的 M 个特征向量, $[\Lambda] \in \mathbb{R}^{[M] \times [M]}$

$\Lambda \in \mathbb{R}^{[M] \times [M]}$ 在 (10.78) 中 $= \text{diag}(\lambda_1, \dots, \lambda_M) \in \mathbb{R}^{[M] \times [M]}$ 是一个对角矩阵, 其对角线上有与主轴相关的特征值 (保证是 $[M] \times [M]$), $R \in \mathbb{R}^{[M] \times [M]}$ 是 [正半定的] 一个任意的正交矩阵。最大似然解 $B \in \mathbb{R}^{[M] \times [M]}$ 在任意正交变换下是 [因为最小的] 唯一的, 例如, 我们可以将 B [数据协方差的] $[ML]$ 右乘以任意旋转矩阵 R , 使得 (10.78) 本质上是一个 [特征值]

奇异值分解 (参见第 4.5 节)。证明的概要由 [Tipping and Bishop (1999)] 给出。[矩阵从下方]

[被噪声方差] $[\sigma^2]$ 限定。

在 (10.77) 中给出的 μ 的最大似然估计是数据的样本均值。在 (10.79) 中给出的观测噪声方差 σ^2 的最大似然估计量是主子空间正交补中的平均方差, 即, 我们无法用前 M 个主成分捕获的平均剩余方差被视为观测噪声。

在无噪声极限 $\sigma \rightarrow 0$ 的情况下, PPCA 和 PCA 提供相同的解: 由于数据协方差矩阵 S 是对称的, 它可以被对角化 (参见第 4.4 节), 即, 存在 S 的特征向量矩阵 T 使得

$$S = T \Lambda T^T. \quad (10.80)$$

在 PPCA 模型中, 数据协方差矩阵是 Gaussian 似然 $p(x | B, \mu, \sigma)$ 的协方差矩阵, 即 $BB^T + \sigma^2 I$, 参见 (10.70b). 对于 $\sigma \rightarrow 0$, 我们得到 BB^T , 因此这个数据协方差必须等于 PCA 数据协方差 (及其在 (10.80) 中给出的因式分解), 使得

$$\text{Cov}([x] \in \mathbb{R}^{[T]} X) = T \Lambda T^T = BB^T \iff B = T \Lambda^{1/2} R, \quad (10.81)$$

即, 我们得到 $\sigma = 0$ 时 (10.78) 中的最大似然估计。

从 (10.78) 和 (10.80) 可以清楚地看出, (P)PCA 执行数据协方差矩阵的分解。

在流式设置中, 数据按顺序到达, 建议使用迭代期望最大化 (EM) 算法进行最大似然估计 [Roweis, 1998].

为了确定潜在变量的维数 (代码的长度, 我们投影数据的低维子空间的维数), [Gavish and Donoho (2014)] 建议使用启发式方法, 如果我们能估计数据的噪声方差 σ^2 , 我们应该

$$[\sqrt{\cdot}]$$

丢弃所有小于 $4[\sigma][D]$ 的奇异值。或者，我们可以使用 $\sqrt{\cdot}$

3

(嵌套)交叉验证(第[8.6.1]节)或贝叶斯模型选择准则(在第[8.6.2]节中讨论)来确定数据固有维度的良好估计[(Minka, 2001b).]

类似于我们在第9章中对线性回归的讨论，我们可以在模型参数上放置先验分布并将它们积分出去。通过这样做，我们(a)避免了参数的点估计以及这些点估计带来的问题(参见第[8.6]节)，并且(b)允许自动选择潜在空间的适当维数 M 。在这种贝叶斯PCA(Bayesian PCA)中，由[Bishop (1999).]提出，

[Bayesian PCA]在模型参数上放置先验 $p(\mu, B, \sigma)$ 。生成过程允许我们积分出模型参数而不是基于它们进行条件化，这解决了过拟合问题。由于这种积分在解析上是难以处理的，[Bishop (1999)]提议使用近似推理方法，如MCMC或变分推理。我们参考

的工作以及[Blei et al. (2017)]的工作获取这些近似推理技术的更多细节。

在PPCA中，我们考虑了线性模型 $p(x[n] | z[n]) = N(x[n] | Bz[n] + \mu, \sigma^2 I)$ ，先验为 $p(z[n]) = N(0, I)$ ，其中所有观测维度都受到相同噪声量的影响。如果我们允许每个观测维度 d 具有不同的方差 $\sigma^2[d]$ ，我们就得到了**因子分析(FA)** [(Spearman, 1904; Bartholomew et al., 2011)]。这意味着FA比PPCA为似然函数提供了更多的灵活性，但仍然强制数据通过模型参数 B 、 μ 来解释。然而，FA不再允许闭式最大似然解，因此我们需要使用迭代方案，如期望最大化算法来估计模型参数。虽然在PPCA中所有驻点都是全局最优解，但这在FA中不再成立。与PPCA相比，FA在我们缩放数据时不会改变，但如果我们旋转数据，它会返回不同的解。

一个与PCA密切相关的算法是**独立成分分析(ICA)** [(Hyvarinen et al., 2001)]。再次从潜变量视角开始 $p(x[n] | z[n]) = N(x[n] | Bz[n] + \mu, \sigma^2 I)$ ，我们现在将 $z[n]$ 的先验改为非高斯分布。ICA可用于**盲源分离**。想象你在一个繁忙的火车站，有很多人在说话。你的耳朵扮演麦克风的角色，它们线性混合火车站中的不同语音信号。盲源分离的目标是识别混合信号的组成部分。

如前所述，在PPCA最大似然估计的背景下，原始PCA解对任何旋转都是不变的。因此，PCA可以识别信号所在的最佳低维空间，但不能识别信号本身[(Murphy, 2012)]。ICA通过修改潜在源上的先验分布 $p(z)$ 来解决这个问题，要求非高斯先验 $p(z)$ 。我们参考[Hyvarinen et al. (2001)]和[Murphy (2012)]的书籍获取ICA的更多细节。

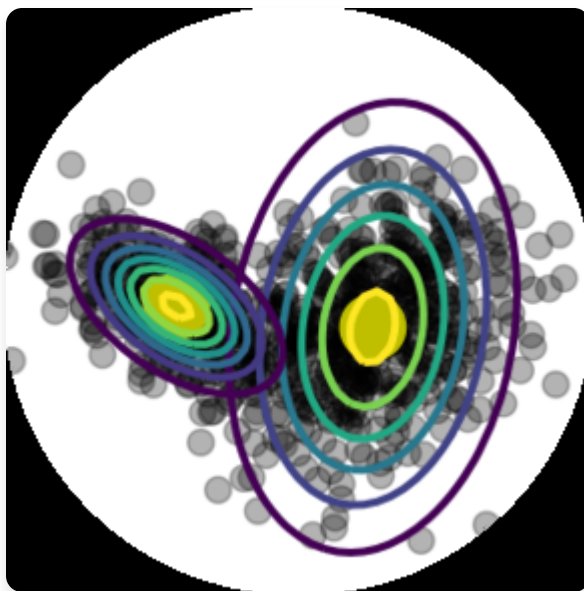
PCA、因子分析和ICA是使用线性模型进行维度约简的三个例子。[Cunningham and Ghahramani (2015)]提供了线性维度约简的更广泛调查。

我们在这里讨论的(P)PCA模型允许几个重要的扩展。在第10.5节中，我们解释了当输入维度 D 明显大于数据点数量 N 时如何进行PCA。通过利用PCA可以通过计算(多个)内积来执行的洞察，这个想法可以通过考虑无限维特征推向极端。**核技巧**是**核PCA**的基础，允许我们隐式计算无限维特征之间的内积[(Schölkopf et al., 1998; Schölkopf and Smola, 2002)]。

有从PCA派生的非线性维度约简技术[(Burgess (2010)提供了良好的概述)]。我们之前在本节中讨论的PCA的自编码器视角可以用来将PCA呈现为**深度自编码器**的特殊情况。在深度自编码器中，编码器和解码器都由多层前馈神经网络表示，它们本身是非线性映射。如果我们将这些神经网络中的激活函数设置为恒等函数，模型就等价于PCA。非线性维度约简的不同方法是[Lawrence (2005)]提出的**高斯过程潜变量模型(GP-LVM)**。GP-LVM从我们用来推导PPCA的潜变量视角开始，用高斯过程(GP)替换潜变量 z 和观测值 x 之间的线性关系。GP-LVM不是估计映射的参数(如我们在PPCA中

所做的), 而是边际化模型参数并对潜变量 z 进行点估计。类似于贝叶斯PCA, [Titsias and Lawrence (2010)]提出的**贝叶斯GP-LVM**在潜变量 z 上维持分布, 并使用近似推理将它们也积分出来。

11 使用高斯混合模型的密度估计



在前面的章节中，我们已经涵盖了机器学习中的两个基本问题：回归(第9章)和维度约简(第10章)。在本章中，我们将看看机器学习的第三个支柱：密度估计。在我们的旅程中，我们介绍重要概念，如期望最大化(EM)算法和混合模型密度估计的潜变量视角。

当我们将机器学习应用于数据时，我们经常旨在以某种方式表示数据。一种直接的方法是将数据点本身作为数据的表示；参见图11.1的例子。然而，如果数据集很大或者我们对表示数据的特征感兴趣，这种方法可能没有帮助。在密度估计中，我们使用参数族的密度(例如高斯分布或Beta分布)紧凑地表示数据。例如，我们可能

寻找数据集的均值和方差，以便使用 Gaussian 分布紧凑地表示数据。均值和方差可以使用我们在第 8.3 节中讨论的工具找到：maximum likelihood 或 maximum a posteriori 估计。然后我们可以使用这个 Gaussian 的均值和方差来表示数据背后的分布，即，如果从该分布中采样，我们认为数据集是该分布的典型实现。

图 11.1 无法用 Gaussian 有意义表示的二维数据集。



Two-dimensional dataset that cannot be meaningfully represented by a Gaussian

Two-dimensional dataset that cannot be meaningfully represented by a Gaussian

在实践中，Gaussian（或类似地，我们迄今为止遇到的所有其他分布）具有有限的建模能力。例如，对图 11.1 中生成数据的密度进行 Gaussian 近似将是一个糟糕的近似。在下面，我们将研究一个更具表达力的分布族，我们可以用它进行密度估计：mixture models。

Mixture models 可以用来通过 K 个简单（基础）分布的凸组合来描述分布 $p(x)$

$$p(x) = \sum_{k=1}^K \pi_k p_k(x) \quad (11.1)$$

$$0 \leq \pi_k \leq 1, \quad \sum_{k=1}^K \pi_k = 1 \quad (11.2)$$

其中组件 p_k 是基础分布族的成员，例如 Gaussians、Bernoullis 或 Gammas， π_k 是 *mixture weights*。

Mixture models 比相应的基础分布更具表达力，因为它们允许多模态数据表示，即它们可以描述具有多个“聚类”的数据集，如图 11.1 中的示例。

我们将专注于 Gaussian mixture models (GMMs)，其中基础分布是 Gaussians。对于给定的数据集，我们的目标是最大化模型参数的 likelihood 来训练 GMM。为此，我们将使用第 5 章、第 6 章和第 7.2 节的结果。然而，与我们之前讨论的其他应用（线性回归或 PCA）不同，我们不会找到封闭形式的 maximum likelihood 解。相反，我们将得到一组相互依赖的联立方程，只能迭代求解。

11.1 Gaussian Mixture Model

Gaussian mixture model 是一个密度模型，我们将有限数量的 K 个 Gaussian 分布 $N(x|\mu_k, \Sigma_k)$ 组合起来，使得

$$p(x|\theta) = \sum_{k=1}^K \pi_k N(x|\mu_k, \Sigma_k) \quad (11.3)$$

$$0 \leq \pi_k \leq 1, \quad \sum_{k=1}^K \pi_k = 1 \quad (11.4)$$

其中我们定义 $\theta := \{\mu_k, \Sigma_k, \pi_k : k = 1, \dots, K\}$ 作为模型所有参数的集合。这种 Gaussian 分布的凸组合为我们建模复杂密度提供了比简单 Gaussian 分布（我们从 (11.3) 中对 $K=1$ 恢复）显著更多的灵活性。图 11.2 给出了一个插图，显示了加权

图 11.2 Gaussian mixture model。Gaussian mixture 分布（黑色）由 Gaussian 分布的凸组合组成，比任何单个组件更具表达力。虚线表示加权的 Gaussian 组件。



Gaussian mixture model

Gaussian mixture model

组件和 mixture 密度，给出为

$$p(x|\theta) = 0.5N(x|-2, 2) + 0.2N(x|1, 3) + 0.3N(x|4, 1) \quad (11.5)$$

11.2 通过 Maximum Likelihood 进行参数学习

假设我们得到一个数据集 $X = \{x_1, \dots, x_N\}$, 其中 $x_n, n = 1, \dots, N$, 从未知分布 $p(x)$ 中独立同分布地抽取。我们的目标是通过具有 K 个 mixture 组件的 GMM 找到这个未知分布 $p(x)$ 的良好近似/表示。GMM 的参数是 K 个均值 μ_k 、协方差 Σ_k 和 mixture weights π_k 。我们将所有这些自由参数总结为 $\theta := \{\pi_k, \mu_k, \Sigma_k : k = 1, \dots, K\}$

Example 11.1 (初始设置)

图 11.3 初始设置：具有三个 mixture 组件（虚线）和七个数据点（圆盘）的 GMM（黑色）。



Initial setting: GMM with mixture components

Initial setting: GMM with mixture components

在本章中，我们将有一个简单的运行示例来帮助我们说明和可视化重要概念。

我们考虑一个一维数据集 $X = \{-3, -2.5, -1, 0, 2, 4, 5\}$ ，包含七个数据点，希望找到一个具有 $K = 3$ 个组件的 GMM 来建模数据的密度。我们将 mixture 组件初始化为

$$p_1(x) = \mathcal{N}(x|-4, 1) \quad (11.6) \quad p_2(x) = \mathcal{N}(x|0, 0.2) \quad (11.7)$$

$$p_3(x) = \mathcal{N}(x|8, 3) \quad (11.8)$$

并为它们分配相等的权重 $\pi_1 = \pi_2 = \pi_3 = 1/3$ 。

模型（和数据点）如图11.3所示。

在下文中，我们详细说明如何获得模型参数 θ 的最大似然估计 θ_{ML} 。我们首先写出似然函数，即给定参数的训练数据的预测分布。我们利用独立同分布假设，这导致因子化的似然函数

$$p(X|\theta) = \prod_{n=1}^N p(x_n|\theta), p(x_n|\theta) = \sum_{k=1}^K \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k), \quad (11.9)$$

其中每个单独的似然项 $p(x_n|\theta)$ 都是一个Gaussian mixture密度。然后我们得到对数似然函数为

$$\log p(X|\theta) = \sum_{n=1}^N \log p(x_n|\theta) = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k). \quad (11.10)$$

我们的目标是找到使(11.10)中定义的对数似然函数 L 最大化的参数 θ^*_{ML} 。我们的”常规”程序是计算对数似然函数相对于模型参数 θ 的梯度 $dL/d\theta$ ，将其设为0，并求解 θ 。然而，与我们之前的最大似然估计示例（例如，当我们在第9.2节讨论线性回归时）不同，我们无法获得封闭形式的解。但是，我们可以利用迭代方案来找到良好的模型参数 θ_{ML} ，这将证明是GMM的EM算法。关键思想是在保持其他参数固定的同时，一次更新一个模型参数。

备注。如果我们考虑单个Gaussian作为期望密度，(11.10)中对 k 的求和消失，对数可以直接应用于Gaussian分量，从而得到

$$\log \mathcal{N}(x|\mu, \Sigma) = -D/2 \log(2\pi) - 1/2 \log \det(\Sigma) - 1/2 (x-\mu)^T \Sigma^{-1} (x-\mu). \quad (11.11)$$

这种简单形式允许我们找到 μ 和 Σ 的封闭形式最大似然估计，如第8章所讨论的。在(11.10)中，我们不能将对数移入对 k 的求和中，因此无法获得简单的封闭形式最大似然解。◇

函数的任何局部最优值都具有相对于参数的梯度必须消失的性质（必要条件）；见第7章。在我们的情况下，当我们相对于GMM参数 μ_k, Σ_k, π_k 优化(11.10)中的对数似然函数时，我们得到以下必要条件：

$$\partial L / \partial \mu_k = \sum_{n=1}^N \partial \log p(x_n | \theta) / \partial \mu_k = 0, \quad (11.12)$$

$$\partial L / \partial \Sigma_k = \sum_{n=1}^N \partial \log p(x_n | \theta) / \partial \Sigma_k = 0, \quad (11.13)$$

$$\partial L / \partial \pi_k = \sum_{n=1}^N \partial \log p(x_n | \theta) / \partial \pi_k = 0. \quad (11.14)$$

对于所有三个必要条件，通过应用链式法则（见第5.2.2节），我们需要以下形式的偏导数

$$\partial \log p(x_n | \theta) / \partial \theta = 1/p(x_n | \theta) \times \partial p(x_n | \theta) / \partial \theta, \quad (11.15)$$

其中 $\theta = \{\mu_k, \Sigma_k, \pi_k, k = 1, \dots, K\}$ 是模型参数，

$$p(x_n | \theta) = \sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j). \quad (11.16)$$

在下文中，我们将计算偏导数(11.12)到(11.14)。但在此之前，我们引入一个将在本章剩余部分发挥核心作用的量：responsibilities。

11.2.1 Responsibilities

我们定义量

$$r_{nk} := \pi_k N(x_n | \mu_k, \Sigma_k) / \sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j) \quad (11.17)$$

作为第k个mixture分量对第n个数据点的 *responsibility*。

第k个mixture分量对数据点 x_n 的responsibility r_{nk} 与给定数据点的mixture分量的似然

$$p(x_n | \pi_k, \mu_k, \Sigma_k) = \pi_k N(x_n | \mu_k, \Sigma_k) \quad (11.18)$$

成正比。因此，当数据点可能是该mixture分量的合理样本时，mixture分量对数据点具有高responsibility。注意 $\mathbf{r}_n := [r_{n1}, \dots, r_{nK}]^T \in \mathbb{R}^K$ 是一个（归一化的）概率向量，即

$\sum_k r_{nk} = 1$ ，其中 $r_{nk} \geq 0$ 。这个概率向量在K个mixture分量之间分配概率质量，我们可以将 $\mathbf{r}_n$ 视为 x_n 到K个mixture分量的“软分配”。因此，来自(11.17)的responsibility r_{nk} 表示 x_n 由第k个mixture分量生成的概率。

例11.2 (Responsibilities)

对于图11.3中的例子，我们计算responsibilities r_{nk}

$$\begin{bmatrix} \text{THIS IS MATRIX:} & 1.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ & 0.057 & 0.943 & 0.0 & 0.001 & 0.999 & 0.0 \end{bmatrix} \in \mathbb{R}^{(N \times K)}. \quad (11.19)$$
$$\begin{bmatrix} 0.0 & 0.066 & 0.934 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

这里第n行告诉我们所有mixture分量对 x_n 的responsibilities。数据点的所有K个responsibilities的和（每行的和）为1。第k列给出了第k个mixture分量的responsibility概述。我们可以看到第三个mixture分量（第三列）对前四个数据点不负责任，但对其余数据点承担很大责任。列的所有条目的和给出了值 N_k ，即第k个mixture分量的总responsibility。在我们的例子中，我们得到 $N_1 = 2.058$ ， $N_2 = 2.008$ ， $N_3 = 2.934$ 。

在下文中，我们确定给定责任度(responsibilities)时模型参数 $\mu[k]$ ， $\Sigma[k]$ ， $\pi[k]$ 的更新。我们将看到更新方程都依赖于责任度，这使得最大似然估计问题无法得到闭式解。然而，对于给定的责任度，我们将一次更新一个模型参数，同时保持其他参数固定。之后，我们将重新计算责任度。迭代这两个步骤最终将收敛到局部最优解，这是EM算法的一个特定实例。我们将在第[11.3]节中更详细地讨论这一点。

11.2.2 更新均值

定理11.1 (GMM均值的更新)。GMM均值参数 $\mu[k], k = 1, \dots, K$ 的更新由下式给出：

$$\mu[k]_{\text{new}} = \frac{1}{N} \sum_{n=1}^N r[n] x[n], \quad (11.20)$$

其中责任度 $r[nk]$ 在(11.17)中定义。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[354] [[使用Gaussian混合模型进行密度估计]]

备注。单个混合分量均值 $\mu[k]$ 在(11.20)中的更新依赖于通过(11.17)中给出的 $r[nk]$ 的所有均值、协方差矩阵 $\Sigma[k]$ 和混合权重 $\pi[k]$ 。因此，我们无法一次获得所有 $\mu[k]$ 的闭式解。[◇]

证明 从(11.15)中，我们看到log-likelihood对于均值参数 $\mu[k], k = 1, \dots, K$ 的梯度需要我们计算偏导数

$$\frac{\partial}{\partial \mu[k]} \log p(x | \theta) = \frac{1}{N} \sum_{n=1}^N \frac{\partial}{\partial \mu[k]} p(x[n] | \theta) = \frac{1}{N} \sum_{n=1}^N r[n] \frac{\partial}{\partial \mu[k]} p(x[n] | \theta) \quad (11.21a)$$

$$= \frac{1}{N} \sum_{n=1}^N r[n] \frac{\partial}{\partial \mu[k]} \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x[n] - \mu[k])^2}{2\Sigma[k]}\right) \right) \quad (11.21b)$$

这里我们利用了只有第 k 个混合分量依赖于 $\mu[k]$ 。

我们在(11.15)中使用(11.21b)的结果并将所有内容整合在一起，以便 L 对 $\mu[k]$ 的所需偏导数给出为：

$$\frac{\partial}{\partial \mu[k]} L = \frac{1}{N} \sum_{n=1}^N r[n] \frac{\partial}{\partial \mu[k]} p(x[n] | \theta) = \frac{1}{N} \sum_{n=1}^N r[n] \frac{\partial}{\partial \mu[k]} \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x[n] - \mu[k])^2}{2\Sigma[k]}\right) \right) \quad (11.22a)$$

$$= \frac{1}{N} \sum_{n=1}^N r[n] \frac{\partial}{\partial \mu[k]} \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x[n] - \mu[k])^2}{2\Sigma[k]}\right) \right) = \frac{1}{N} \sum_{n=1}^N r[n] \frac{\partial}{\partial \mu[k]} \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x[n] - \mu[k])^2}{2\Sigma[k]}\right) \right) \quad (11.22b)$$

$$= \frac{1}{N} \sum_{n=1}^N r[n] \frac{\partial}{\partial \mu[k]} \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x[n] - \mu[k])^2}{2\Sigma[k]}\right) \right)$$

$$= \frac{1}{N} \sum_{n=1}^N r[n] \frac{\partial}{\partial \mu[k]} \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x[n] - \mu[k])^2}{2\Sigma[k]}\right) \right) \quad (11.22c)$$

这里我们使用了(11.16)的恒等式和(11.21b)中偏导数的结果来得到(11.22b)。值 $r[nk]$ 是我们在(11.17)中定义的责任度。

我们现在求解(11.22c)使得 $\frac{\partial}{\partial \mu[k]} L = 0$ 并得到：

$$\sum_{n=1}^N r[n] \frac{\partial}{\partial \mu[k]} p(x[n] | \theta) = 0 \Leftrightarrow \sum_{n=1}^N r[n] \frac{\partial}{\partial \mu[k]} \left(\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x[n] - \mu[k])^2}{2\Sigma[k]}\right) \right) = 0 \quad (11.23)$$

其中我们定义：

$$N[k] := \sum_{n=1}^N r[n] \quad (11.24)$$

作为第k个混合分量对整个数据集的总责任度。这完成了定理[11.1]的证明。

直观上，[(11.20)]可以解释为均值的重要性加权Monte Carlo估计，其中数据点 $x[n]$ 的重要性权重是第k个聚类对 $x[n]$ 的责任度 $r[nk]$, $k = 1, \dots, K$ 。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] <https://mml-book.com>[.] [11.2 通过最大似然进行参数学习] [355]

因此，均值 $\mu[k]$ 被拉向数据点 $x[n]$ ，拉力强度由 $r[nk]$ 给出。**图11.4** [GMM中混合分量均值参数的更新。]均值被更强地拉向对应混合分量具有高责任度（即高似然度）的数据点。图[11.4]说明了这一点。我们也可以将[(11.20)]中的均值更新解释为所有数据点在由以下分布给出的条件下的期望值：[均值 μ 被拉向具有相应责任度权重的各个数据点。]

$$r[\tau] := [r_1, \dots, r_K] / N, \quad (11.25) \quad [r]_{[K]} [1/k] [N/k] [k]$$

这是一个归一化的概率向量，即：

$$\mu \leftarrow \sum_k r[k] x[k], \quad (11.26) \quad [x]_{[2]} [x] [r]_{[2]}$$

示例11.3（均值更新） $[r]_{[3 \times 1]} [\mu]$

图11.5 [在GMM中更新均值的效果。]

[0] [30] $\pi[N] \left(\left(\frac{1}{N} \sum_{n=1}^N \mu_n \right), \sigma^2 \right) [1] [D] [\pi] [N] [0] [1] [30] [1] \left(\left(\frac{1}{N} \sum_{n=1}^N \mu_n \right), \sigma^2 \right) [1] [D] [0] [25] \pi[N] \left(\left(\frac{1}{N} \sum_{n=1}^N \mu_n \right), \sigma^2 \right) [3] [D] [\pi] [0] [1] [25] [N] \left(\left(\frac{1}{N} \sum_{n=1}^N \mu_n \right), \sigma^2 \right) [3] [D] [(\alpha)$ 更新均值前的GMM $[\pi]_{[2]} [N] \left(\left(\frac{1}{N} \sum_{n=1}^N \mu_n \right), \sigma^2 \right) [2] [D] [\pi]_{[2]} [N] \left(\left(\frac{1}{N} \sum_{n=1}^N \mu_n \right), \sigma^2 \right) [2] [D]$

[GMM density] [GMM density] [(b) 更新均值 μ_k 后的GMM，同时保持方差和混合权重不变。] [0] [20] [0] [20]

([0] [15] ([p] [D] [p] [0] [15]

[0] [10] [0] [10]

[0] [05] [0] [05]

[0] [1] [00] [0] [1] [00] [-] [5] [0] [5] [10] [15] [-] [5] [0] [5] [10] [15] x x

[(a) 更新均值前的GMM密度和各分量] [(b) 更新均值后的GMM密度和各分量]

在图[11.3]的示例中，均值按如下方式更新：

$$\mu_1: -4 \rightarrow -2.7 \quad (11.27) \quad \mu_2: 0 \rightarrow -0.4 \quad (11.28) \quad \mu_3: 8 \rightarrow 3.7 \quad (11.29)$$

这里我们看到第一和第三个混合分量的均值向数据范围移动，而第二个分量的均值变化不那么剧烈。图[11.5]说明了这种变化，其中图[11.5(a)]显示了更新均值前的GMM密度，图[11.5(b)]显示了更新均值 $\mu[k]$ 后的GMM密度。

[(11.20)]中均值参数的更新看起来相当直观。然而，注意责任度 $r[nk]$ 是函数

$\pi[j], \mu[j], \Sigma[j]$ 对于所有 $j = 1, \dots, K$ ，使得 [(11.20)] 中的更新依赖于 GMM 的所有参数，因此无法获得我们在第 [9.2] 节的线性回归或第 [10] 章的 PCA 中得到的闭式解。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[356] [[使用高斯混合模型进行密度估计]]

11.2.3 更新协方差

定理 11.2 (GMM 协方差的更新)。GMM 协方差参数 $\Sigma[k]$, $k = 1, \dots, K$ 的更新由下式给出:

$$\Sigma[k]_{\text{new}} = \frac{1}{N[k]} \sum_{n=1}^{N[k]} (x[n] - \mu[k])(x[n] - \mu[k])^T, \quad (11.30)$$

其中 $r[nk]$ 和 $N[k]$ 分别在 [(11.17)] 和 [(11.24)] 中定义。

证明 [为了证明定理 [11.2], 我们的方法是计算对数似然 L 关于协方差 $\Sigma[k]$ 的偏导数, 令其等于 0, 然后求解 $\Sigma[k]$ 。我们从一般方法开始:]

$$\frac{\partial}{\partial \Sigma[k]} L = \frac{\partial}{\partial \Sigma[k]} \log p(x | \theta) = \sum_{n=1}^N \frac{\partial}{\partial \Sigma[k]} \log p(x[n] | \theta) = \sum_{n=1}^N \frac{1}{p(x[n] | \theta)} \frac{\partial p(x[n] | \theta)}{\partial \Sigma[k]} \quad (11.31)$$

我们已经从 [(11.16)] 知道了 $1/p(x[n] | \theta)$ 。为了得到剩余的偏导数 $\partial p(x[n] | \theta) / \partial \Sigma[k]$, 我们写出高斯分布 $p(x[n] | \theta)$ 的定义 (见 [(11.9)]) 并删除除第 k 项之外的所有项。然后我们得到:

$$p(x[n] | \theta) = \pi[k] \frac{1}{(2\pi)^{D/2} \det(\Sigma[k])^{1/2}} \exp\left\{-\frac{1}{2} (x[n] - \mu[k])^T \Sigma[k]^{-1} (x[n] - \mu[k])\right\} \quad (11.32a)$$

$$\frac{\partial}{\partial \Sigma[k]} \log p(x[n] | \theta) = \frac{\partial}{\partial \Sigma[k]} \left\{ \log \pi[k] - \frac{D}{2} \log \det(\Sigma[k]) - \frac{1}{2} (x[n] - \mu[k])^T \Sigma[k]^{-1} (x[n] - \mu[k]) \right\} \quad (11.32b)$$

$$\frac{\partial}{\partial \Sigma[k]} \log \pi[k] = 0, \quad \frac{\partial}{\partial \Sigma[k]} \left(-\frac{D}{2} \log \det(\Sigma[k]) \right) = -\frac{D}{2} \frac{1}{\det(\Sigma[k])} \frac{\partial \det(\Sigma[k])}{\partial \Sigma[k]} \quad (11.32c)$$

$$\frac{\partial}{\partial \Sigma[k]} \left(-\frac{1}{2} (x[n] - \mu[k])^T \Sigma[k]^{-1} (x[n] - \mu[k]) \right) = \frac{1}{2} \Sigma[k]^{-1} (x[n] - \mu[k])(x[n] - \mu[k])^T \Sigma[k]^{-1} \quad (11.32d)$$

我们现在使用恒等式:

$$\frac{\partial}{\partial \Sigma[k]} \det(\Sigma[k]) = \det(\Sigma[k]) \text{tr}(\Sigma[k]^{-1}) \quad (11.33)$$

$$\frac{\partial}{\partial \Sigma[k]} \text{tr}(\Sigma[k]^{-1} (x[n] - \mu[k])(x[n] - \mu[k])^T) = -\Sigma[k]^{-1} (x[n] - \mu[k])(x[n] - \mu[k])^T \Sigma[k]^{-1} \quad (11.34)$$

经过一些重新整理后, 我们得到 [(11.31)] 中所需的偏导数:

$$\frac{\partial}{\partial \Sigma[k]} \log p(x[n] | \theta) = \frac{1}{2} \text{tr}(\Sigma[k]^{-1} (x[n] - \mu[k])(x[n] - \mu[k])^T \Sigma[k]^{-1}) - \frac{D}{2} \frac{1}{\det(\Sigma[k])} \det(\Sigma[k]) \text{tr}(\Sigma[k]^{-1}) \quad (11.35)$$

将所有内容组合在一起, 对数似然关于 $\Sigma[k]$ 的偏导数为:

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>] [11.2 通过最大似然进行参数学习] [357]

$$\frac{\partial}{\partial \Sigma[k]} L = \sum_{n=1}^N \frac{\partial}{\partial \Sigma[k]} \log p(x[n] | \theta) = \sum_{n=1}^N \left(\frac{1}{2} \text{tr}(\Sigma[k]^{-1} (x[n] - \mu[k])(x[n] - \mu[k])^T \Sigma[k]^{-1}) - \frac{D}{2} \frac{1}{\det(\Sigma[k])} \det(\Sigma[k]) \text{tr}(\Sigma[k]^{-1}) \right) \quad (11.36a)$$

$$\frac{\partial}{\partial \Sigma[k]} L = \sum_{n=1}^N \left(\frac{1}{2} \text{tr}(\Sigma[k]^{-1} (x[n] - \mu[k])(x[n] - \mu[k])^T \Sigma[k]^{-1}) - \frac{D}{2} \frac{1}{\det(\Sigma[k])} \det(\Sigma[k]) \text{tr}(\Sigma[k]^{-1}) \right) \quad (11.36b)$$

$$\frac{\partial}{\partial \Sigma[k]} L = \sum_{n=1}^N \left(\frac{1}{2} \text{tr}(\Sigma[k]^{-1} (x[n] - \mu[k])(x[n] - \mu[k])^T \Sigma[k]^{-1}) - \frac{D}{2} \frac{1}{\det(\Sigma[k])} \det(\Sigma[k]) \text{tr}(\Sigma[k]^{-1}) \right) \quad (11.36c)$$

$[\pi] [0][.][30] [0][.]$

$[25] 3 [N] [x][1]\mu[, \sigma] [2] [3]D] \pi[N][0][x][1]\mu[, \sigma][2] [3]D]$

[GMM 密度] [GMM 密度] [0] [.] [25]

$[0][.][20] D] D] [0][.][20] x \times [0]$

$[p] [0] [0][.] [15]p] [0][.][15]$

$[0][.][10] [0][.][10]$

$[0][.][05] [0][.][05]$

$[0][.][00] [0][.][00]$

$[-][4] [-][2] [0] [2] [4] 6 [8] [-][4] [-][2] [0] [2] [4] 6 [8]$

$x \times x$

[(a) 更新方差前的 GMM; (b) 更新方差后的 GMM, 同时保持均值和混合权重不变。] [[(a) GMM 密度和各个分量]] [(b) GMM 密度和各个分量]]

[在更新方差之前。] [在更新方差之后。]

与均值参数的更新类似, 我们可以将[(11.30)]解释为与第 k 个混合分量相关联的数据点 $x[n]$ 的加权协方差的Monte Carlo估计, 其中权重是responsibility $r[nk]$ 。与均值参数的更新一样, 这个更新通过responsibility $r[nk]$ 依赖于所有 $\pi[j]$ 、 $\mu[j]$ 、 $\Sigma[j]$, $j = 1, \dots, K$, 这阻止了闭式解的存在。

11.2.4 更新混合权重

定理11.3 (GMM混合权重的更新)。*GMM的混合权重更新为*

$$\pi[k][\text{new}] = N[k]/N, \quad k = 1, \dots, K, \quad (11.42)$$

其中 N 是数据点的数量, $N[k]$ 在[(11.24)]中定义。

证明 [为了找到log-likelihood关于权重参数 $\pi[k]$, $k = 1, \dots, K$ 的偏导数, 我们通过使用Lagrange乘数来考虑约束 $\sum \pi[k] = 1$ (见第[7.2]节)。]Lagrangian是

$$L = \sum_{k=1}^K L + \lambda (\sum_{k=1}^K \pi[k] - 1) \quad (11.43a)$$

$$= \sum_{n=1}^N \sum_{k=1}^K \log \pi[k] N(x[n] | \mu[k], \Sigma[k]) + \lambda (\sum_{k=1}^K \pi[k] - 1), \quad (11.43b)$$

其中 L 是来自[(11.10)]的log-likelihood, 第二项编码了所有混合权重需要和为1的等式约束。我们获得关于 $\pi[k]$ 的偏导数为

$$\partial L / \partial \pi[k] = \sum_{n=1}^N N(x[n] | \mu[k], \Sigma[k]) / (\sum_{j=1}^K \pi[j] N(x[n] | \mu[j], \Sigma[j])) + \lambda \quad (11.44a)$$

$$= 1/\pi[k] \sum_{n=1}^N \pi[k] N(x[n] | \mu[k], \Sigma[k]) / (\sum_{j=1}^K \pi[j] N(x[n] | \mu[j], \Sigma[j])) + \lambda = N[k]/\pi[k] + \lambda, \quad (11.44b)$$

以及关于Lagrange乘数 λ 的偏导数为

$$\partial L / \partial \lambda = \sum_{k=1}^K \pi[k] - 1. \quad [(11.45)]$$

将两个偏导数都设置为0 (最优的必要条件) 得到方程组

$$\pi[k] = -N[k]/\lambda, \quad (11.46) \quad 1 = \sum_{k=1}^K \pi[k]. \quad [(11.47)]$$

在[(11.47)]中使用[(11.46)]并求解 $\pi[k]$, 我们得到

$$\sum_{k=1}^K \pi[k] = 1 \iff -\sum_{k=1}^K N[k]/\lambda = 1 \iff -\sum_{k=1}^K N[k]/\lambda = 1 \iff -N/\lambda = 1. \quad (11.48)$$

这允许我们在[(11.46)]中用 $-N$ 替换 λ 来获得

$$\pi[k][\text{new}] = N[k]/N, \quad (11.49)$$

这给出了权重参数 $\pi[k]$ 的更新并证明了定理[11.3]。

我们可以将[(11.42)]中的混合权重识别为第 k 个聚类的总responsibility与数据点数量的比值。由于 $\sum_k N[k] = N$, 数据点的数量也可以解释为所有混合分量一起的总responsibility, 因此 $\pi[k]$ 是第 k 个混合分量对数据集的相对重要性。

注释。 由于 $N[k] = \sum_{i=1}^N r[nk]$, 混合权重 $\pi[k]$ 的更新方程[(11.42)]也通过responsibility $r[nk]$ 依赖于所有 $\pi[j]$ 、 $\mu[j]$ 、 $\Sigma[j]$, $j = 1, \dots, K$ 。◇

示例11.5 (权重参数更新)

图11.7 在GMM中更新混合权重的效果。(a) 更新混合权重前的GMM密度和各个分量；(b) 更新混合权重后的GMM密度和各个分量。注意垂直轴的不同尺度。

在我们来自图[11.3]的运行示例中，混合权重更新如下：

$$\pi_1 : 1/3 \rightarrow 0.29 \text{ (11.50)} \quad \pi_2 : 1/3 \rightarrow 0.29 \text{ (11.51)} \quad \pi_3 : 1/3 \rightarrow 0.42 \text{ (11.52)}$$

这里我们看到第三个分量获得了更多的权重/重要性，而其他分量变得稍微不那么重要。图[11.7]说明了更新混合权重的效果。图[11.7(a)]与图[11.6(b)]相同，显示了更新混合权重前的GMM密度及其各个分量。图[11.7(b)]显示了更新混合权重后的GMM密度。

总的来说，在更新了均值、方差和权重一次后，我们获得了图[11.7(b)]中显示的GMM。与图[11.3]中显示的初始化相比，我们可以看到参数更新使得GMM密度将一些质量向数据点移动。

在更新均值、方差和权重一次后，图[11.7(b)]中的GMM拟合已经比图[11.3]的初始化显著更好。这也被log-likelihood值所证实，它从28.3（初始化）增加到一个完整更新周期后的14.4。

11.3 EM算法

不幸的是，[(11.20)]、[(11.30)]和[(11.42)]中的更新并不构成混合模型参数 $\mu[k]$ 、 $\Sigma[k]$ 、 $\pi[k]$ 更新的闭式解，因为responsibility $r[nk]$ 以复杂的方式依赖于这些参数。然而，结果建议了一个简单的迭代

参数估计问题的方案，通过[EM算法]最大似然求解。期望最大化算法(EM算法)由[Dempster et al. (1977)]提出，是一种在混合模型以及更一般的潜变量模型中学习参数（最大似然或MAP）的通用迭代方案。

在我们的Gaussian混合模型示例中，我们为 $\mu[k]$, $\Sigma[k]$, $\pi[k]$ 选择初始值并交替进行直到收敛：

E步： 评估责任 $r[nk]$ （数据点 n 属于混合分量 k 的后验概率）。

M步： 使用更新的责任重新估计参数 $\mu[k]$, $\Sigma[k]$, $\pi[k]$ 。

EM算法的每一步都会增加对数似然函数[(Neal and Hinton, 1999)]。对于收敛性，我们可以直接检查对数似然或参数。用于估计GMM参数的EM算法的具体实例如下：

1. 初始化 $\mu[k]$, $\Sigma[k]$, $\pi[k]$ 。

2. **E步：** 使用当前参数 $\pi[k]$, $\mu[k]$, $\Sigma[k]$ 评估每个数据点 $x[n]$ 的责任 $r[nk]$ ：

$$\pi[k] N(x[n] | \mu[k], \Sigma[k]) r[nk] = \frac{\pi[k] N(x[n] | \mu[k], \Sigma[k])}{\sum_j \pi[j] N(x[n] | \mu[j], \Sigma[j])} \quad (11.53)$$

3. **M步：** 使用当前责任 $r[nk]$ （来自E步）重新估计参数 $\pi[k]$, $\mu[k]$, $\Sigma[k]$ ：[[在更新了均值 $\mu[k]$ 之后，它们随后用于(11.54)，然后用于(11.55)来更新相应的协方差。]]

$$\mu[k] = \frac{1}{N[k]} \sum_{n=1}^N r[nk] x[n] \quad (11.54)$$

$$\Sigma[k] = \frac{1}{N[k]} \sum_{n=1}^N r[nk] (x[n] - \mu[k])(x[n] - \mu[k])^T \quad (11.55)$$

$$\pi[k] = \frac{N[k]}{N} \quad (11.56)$$

示例11.6 (GMM拟合)



图11.8 应用于图11.2 GMM的EM算法

图11.8 应用于图11.2 GMM的EM算法。(a) 最终GMM拟合；(b) 负对数似然作为EM迭代的函数。

当我们在图11.3的示例上运行EM时，经过五次迭代后获得图11.8(a)所示的最终结果，图11.8(b)显示了负对数似然如何作为EM迭代的函数演化。最终的GMM为：

$$p(x) = 0.29 N(x | -2.75, 0.06) + 0.28 N(x | -0.50, 0.25) + 0.43 N(x | 3.64, 1.63) \quad (11.57)$$

我们将EM算法应用于图11.1所示的二维数据集，使用 $K = 3$ 个混合分量。图11.9展示了EM算法的一些步骤，并显示负对数似然作为EM迭代的函数（图11.9(b)）。图11.10(a)显示

图11.9 用EM算法拟合三分量Gaussian混合模型到二维数据集的说明

图11.9 用EM算法拟合三分量Gaussian混合模型到二维数据集的说明。(a) 数据集；(b) 负对数似然（越低越好）作为EM迭代的函数。红点表示显示相应GMM拟合的混合分量的迭代，如(c)到(f)所示。黄色圆盘表示Gaussian混合分量的均值。图11.10(a)显示最终的GMM拟合。

图11.10 EM收敛时的GMM拟合和责任

图11.10 EM收敛时的GMM拟合和责任。(a) EM收敛时的GMM拟合；(b) 每个数据点根据混合分量的责任着色。

相应的最终GMM拟合。图[11.10(b)]可视化了数据点混合成分的最终responsibilities。数据集根据EM收敛时混合成分的responsibilities进行着色。虽然单个混合成分明显负责左侧的数据，但右侧两个数据聚类的重叠可能由两个混合成分生成。很明显，有些数据点不能唯一地分配给单个成分（蓝色或黄色），因此这两个聚类对这些点的responsibilities约为0.5。

11.4 潜变量视角

我们可以从离散潜变量模型的角度来看GMM，即潜变量 z 只能取有限的值集合。这与PCA形成对比，在PCA中潜变量是 $[M] \times R$ 中的连续值数字。

概率视角的优势在于：(i) 它将证明我们在前面章节中做出的一些临时决定的合理性，(ii) 它允许将responsibilities具体解释为后验概率，(iii) 更新模型参数的迭代算法可以作为潜变量模型中最大似然参数估计的EM算法以原则性方式推导出来。

11.4.1 生成过程和概率模型

为了推导GMM的概率模型，考虑生成过程是有用的，即允许我们使用概率模型生成数据的过程。

我们假设一个具有 K 个成分的混合模型，并且数据点 x 可以由恰好一个混合成分生成。我们引入一个

二进制指示变量 $z[k] \in \{0, 1\}$ 具有两个状态（见第[6.2]节），它指示第 k 个混合成分是否生成了该数据点

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[364] [[使用高斯混合模型的密度估计]]

因此

$$p(x | z = 1) = N(x | \mu, \Sigma) \quad (11.58)$$

我们定义 $\tau \in \mathbb{R}^K$ 为 $\tau := [z_1, \dots, z_K] \in \mathbb{R}^K$ 由 $K-1$ 个0和恰好一个1组成的概率向量。例如，对于 $K=3$ ，一个有效的 τ 将是

$$\tau = [z_1, z_2, z_3] = [0, 1, 0], \text{ 这将选择第二个混合成分，因为 } z_2 = 1.$$

备注。有时这种概率分布被称为“多努利分布”(multinoulli)，是伯努利分布对超过两个值的推广[(Murphy, 2012).]◇

z 的性质意味着 $\sum_{k=1}^K z[k] = 1$ 。因此， z 是一个one-hot编码（也称：1-of- K 表示）。到目前为止，我们假设指示变量 $z[k]$ 是已知的。然而，在实践中并非如此，我们在潜变量 z 上放置一个先验分布

τ

$$p(\tau) = \prod_{k=1}^K \pi_k^{z[k]} \quad (11.59)$$

π_k

π_k

这个概率向量的第 k 个条目

$$\pi_k = p(z[k] = 1) \quad (11.60)$$

描述了第k个混合成分生成数据点x的概率。图11.11 [图形模型]

备注（从GMM采样）。这个潜变量模型的构建（见图11.11中相应的图形模型）使其[具有单个数据点的GMM]

[π] 适合一个非常简单的采样过程（生成过程）来生成数据：

[z] [1] 1. 采样 $z \sim p(z)$ 。

2. 采样 $x \sim p(x | z = 1)$ 。

[μ] 在第一步中，我们根据 $p(z) = \pi$ 随机选择一个混合成分i（通过one-hot编码z）；在第二步中，我们从相应的混合成分中抽取一个样本。当我们丢弃潜变量的样本，只保留 x 时，我们得到GMM的有效样本。这种采样方式，其中随机变量的样本依赖于图形模型中该变量父节点的样本，称为祖先采样。◇

[Σ] x

[k] $[= 1], \dots, K$

[ancestral sampling]

一般来说，概率模型由数据和潜变量的联合分布定义（见第8.4节）。通过在(11.59)和(11.60)中定义的先验 $p(z)$ 以及来自(11.58)的条件 $p(x | z)$ ，我们通过以下方式获得这个联合分布的所有K个成分：

$$p(x, z = 1) = p(x | z = 1) p(z = 1) = \pi N(x | \mu, \Sigma) \quad (11.61)$$

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] <https://mml-book.com> [11.4 潜变量视角] [365]

对于 $k = 1, \dots, K$ ，因此

$$p(x, z = 1) = \pi N(x | \mu, \Sigma)$$

$$p(x, z) = \dots, \quad (11.62)$$

$$p(x, z = 1) = \pi N(x | \mu, \Sigma)$$

这完全指定了概率模型。

11.4.2 似然

为了在潜变量模型中获得似然 $p(x | \theta)$ ，我们需要边缘化掉潜变量（见第8.4.3节）。在我们的情况下，这可以通过从(11.62)中的联合 $p(x, z)$ 中求和所有潜变量来完成，因此

$$p(x | \theta) = \sum_z p(x, z | \theta), \quad \theta := \{\mu, \Sigma, \pi : k = 1, \dots, K\}$$

[z]

(11.63)

我们现在明确地对概率模型的参数 θ 进行条件化，这在之前被省略了。在[(11.63)]中，我们对 z 的所有 K 个可能的 one-hot 编码求和，用 P 表示。由于每个 z 中只有一个非零条目， z 只有 K 种可能的配置/设置。例如，如果 $K = 3$ ，那么 z 可以有以下配置： $[z]$

$1 \ 0 \ 0$
 $0, 1, 0 \dots [(11.64)]$

$0 \ 0 \ 1$

在[(11.63)]中对 z 的所有可能配置求和等价于查看 z 向量的非零条目并写成

$$p(X(x|\theta)) = p(x|\theta, z)p(z|\theta) \quad (11.65a)$$

$[z]$

$[K]$

$$= \sum_{[z]} p(x|\theta, z)p(z|\theta) \quad [(11.65b)]$$

$[k] \ [k]$

$[k][=1]$

使得所需的边际分布为

$[K]$

$$p([(11.65b)]) X(x|\theta) = p(x|\theta, z[k]=1)p(z[k]=1|\theta) \quad (11.66a)$$

$[k][=1]$

$[K]$

$$= \sum_{[k]} \pi N(x|\mu, \Sigma), \quad (11.66b) \quad [k] \ [k] \ [k]$$

$[k][=1]$

我们将其识别为来自[(11.3)]的GMM模型。给定数据集 X ，我们立即得到似然函数

$[N] \ [N] \ [K]$

$$p(Y([(11.66b)]) X(X|\theta)) = p(x[n]|\theta) = \pi[k] N(x[n]|\mu[k], \Sigma[k]), \quad [(11.67)]$$

$[n][=1] \ [n][=1] \ [k][=1]$

[366] [[使用高斯混合模型进行密度估计]]

图11.12 $[\pi]$ [包含N个数据点的GMM图形模型]

$[z][n]$

$[\mu] [k]$

$[\Sigma] [x][n] [k]$

$[k] [= 1], \dots, K]$

$[n] [= 1], \dots, N]$

这正是来自[(11.9)]的GMM似然函数。因此，具有潜在指示器 $z[k]$ 的潜变量模型是思考高斯混合模型的等价方式。

11.4.3 后验分布

让我们简要看一下潜变量 z 的后验分布。根据贝叶斯定理，第 k 个分量生成数据点 x 的后验为

$$p(z[k] = 1)p(x | z[k] = 1) / p(x) = p(z[k] = 1 | x), \quad (11.68)$$

其中边际 $p(x)$ 在[(11.66b)]中给出。这产生了第 k 个指示变量 $z[k]$ 的后验分布

$$p(z[k] = 1)p(x | z[k] = 1) \pi[k] N(x | \mu[k], \Sigma[k]) / \sum_{j=1}^K p(z[j] = 1) \pi[j] N(x | \mu[j], \Sigma[j]) \quad (11.69)$$

我们将其识别为第 k 个混合分量对数据点 x 的责任度(responsibility)。注意我们省略了对GMM参数 $\pi[k], \mu[k], \Sigma[k]$ (其中 $k = 1, \dots, K$) 的显式条件。

11.4.4 扩展到完整数据集

到目前为止，我们只讨论了数据集仅包含单个数据点 x 的情况。然而，先验和后验的概念可以直接扩展到 N 个数据点 $X := \{x_1, \dots, x_N\}$ 的情况。

在GMM的概率解释中，每个数据点 $x[n]$ 都有自己的潜变量

$$z[n] \in \{1, \dots, K\} \quad (11.70)$$

之前（当我们只考虑单个数据点 x 时），我们省略了索引 n ，但现在这变得很重要。

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>] [11.4 潜变量视角] [[367]]

我们在所有潜变量 $z[n]$ 上共享相同的先验分布 π 。相应的图形模型如图[11.12]所示，其中我们使用了板符号(plate notation)。

条件分布 $p(x_1, \dots, x_N | z_1, \dots, z_N)$ 在数据点上分解，并给出为

$$\prod_{n=1}^N$$

$$p(x[n] | z[n]) \quad (11.71)$$

$$[n]=1$$

为了获得后验分布 $p(z[nk] = 1 | x[n])$ ，我们遵循第[11.4.3]节中相同的推理并应用贝叶斯定理得到

$$p(z[nk] = 1 | x[n]) = \frac{p(x[n] | z[nk] = 1) \pi(z[nk] = 1)}{\sum_{j=1}^K p(x[n] | z[nj] = 1) \pi(z[nj] = 1)} \quad (11.72a)$$

$$\pi(x | \mu, \Sigma)$$

$$= \prod_{k=1}^K r_k \quad (11.72b)$$

$$P(z[nk] = 1 | x[n]) \propto \pi(z[nk] = 1) \prod_{j=1}^K r_j$$

$$[j]=1$$

这意味着 $p(z[nk] = 1 | x[n])$ 是第 k 个混合分量生成数据点 $x[n]$ 的（后验）概率，对应于我们在[(11.17)]中引入的责任度 $r[k]$ 。现在责任度不仅具有直观解释，还具有数学上合理的后验概率解释。

11.4.5 重新审视EM算法

我们作为最大似然估计迭代方案引入的EM算法可以从潜变量视角以原则性方式推导出来。给定模型参数的当前设置 $\theta^{(t)}$ ，E步计算期望对数似然函数

$$Q(\theta | \theta^{(t)}) = E[\log p(x, z | \theta)] \quad (11.73a)$$

其中

$$= \int \log p(x, z | \theta) p(z | x, \theta^{(t)}) dz, \quad (11.73b)$$

其中 $\log p(x, z | \theta)$ 的期望是关于潜变量的后验 $p(z | x, \theta^{(t)})$ 求取的。M步通过最大化 $Q(\theta | \theta^{(t)})$ 选择更新的模型参数集 $\theta^{(t+1)}$ 。

虽然EM迭代确实会增加对数似然函数，但不能保证EM收敛到最大似然解。EM算法可能收敛到对数似然函数的局部最大值。可以在多次EM运行中使用参数 θ 的不同初始化来降低陷入糟糕局部最优的风险。我们这里不深入更多细节，但参考[Rogers and Girolami (2016)]和[Bishop (2006)]的优秀阐述。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[368] [[使用高斯混合模型进行密度估计]]

11.5 进一步阅读

GMM可以被认为是生成模型，因为使用祖先采样[Bishop, 2006]生成新数据是直接的。对于给定的GMM参数 $\pi[k], \mu[k], \Sigma[k], k = 1, \dots, K$ ，我们从概率向量 $[\pi_1, \dots, \pi_K]^T$ 中采样一个索引 k ，然后采样一个数据点 $x \sim N(\mu[k], \Sigma[k])$ 。如果我们重复这个过程 N 次，就得到了由GMM生成的数据集。图[11.1]就是使用这个过程生成的。

在本章中，我们假设分量数 K 是已知的。在实践中，这通常不是这种情况。然而，我们可以使用第[8.6.1]节讨论的嵌套交叉验证来找到好的模型。

高斯混合模型与K-means聚类算法密切相关。K-means也使用EM算法将数据点分配到

聚类。如果我们将GMM中的均值视为聚类中心并忽略协方差（或将其设置为 I ），我们就得到了K-means。正如[MacKay (2003)]很好地描述的那样，K-means对数据点到聚类中心 $\mu[k]$ 进行“硬”分配，而GMM通过责任度进行“软”分配。

我们只是简单介绍了GMM的潜变量视角和EM算法。注意EM可以用于一般潜变量模型的学习，例如非线性状态空间模型[Ghahramani and Roweis, 1999; Roweis and Ghahramani, 1999]以及[Barber (2012)]讨论的强化学习。因此，GMM的潜变量视角对于以原则性方式推导相应的EM算法很有用[Bishop, 2006; Barber, 2012; Murphy, 2012]。

我们只讨论了通过EM算法进行最大似然估计来寻找GMM参数。最大似然的标准批评也适用于此：

与线性回归一样，最大似然可能遭受严重的过拟合。在GMM情况下，当混合分量的均值与数据点相同且协方差趋向于0时就会发生这种情况。然后，似然趋向于无穷大。[Bishop (2006)]和[Barber (2012)]详细讨论了这个问题。

我们只获得参数 $\pi[k], \mu[k], \Sigma[k] (k = 1, \dots, K)$ 的点估计，这并不能给出参数值不确定性的任何指示。Bayesian方法会在参数上放置先验，可以用来获得参数的后验分布。这个后验使我们能够计算模型证据（边际似然），可以用于模型比较，这给了我们确定混合分量数量的原则性方法。不幸的是，在这种设置下不可能进行闭式推理，因为该模型没有共轭先验。然而，诸如variational inference等近似方法可以用来获得近似后验[Bishop, 2006]。

图11.13 数据直方图（橙色条）和核密度估计（蓝线）。核密度估计器产生底层密度的平滑估计，而直方图是有多少数据点（黑色）落入单个bin的非平滑计数测量。

在本章中，我们讨论了用于密度估计的混合模型。有大量可用的密度估计技术。在实践中，我们经常使用直方图和核密度估计。

直方图

直方图提供了表示连续密度的非参数方法，由[Pearson (1895)]提出。直方图通过对数据空间进行”分箱”并计算有多少数据点落入每个bin来构建。然后在每个bin的中心绘制一个条形，条形的高度与该bin内数据点的数量成正比。bin大小是一个关键的超参数，糟糕的选择可能导致过拟合和欠拟合。如第8.2.4节所讨论的，交叉验证可以用来确定好的bin大小。

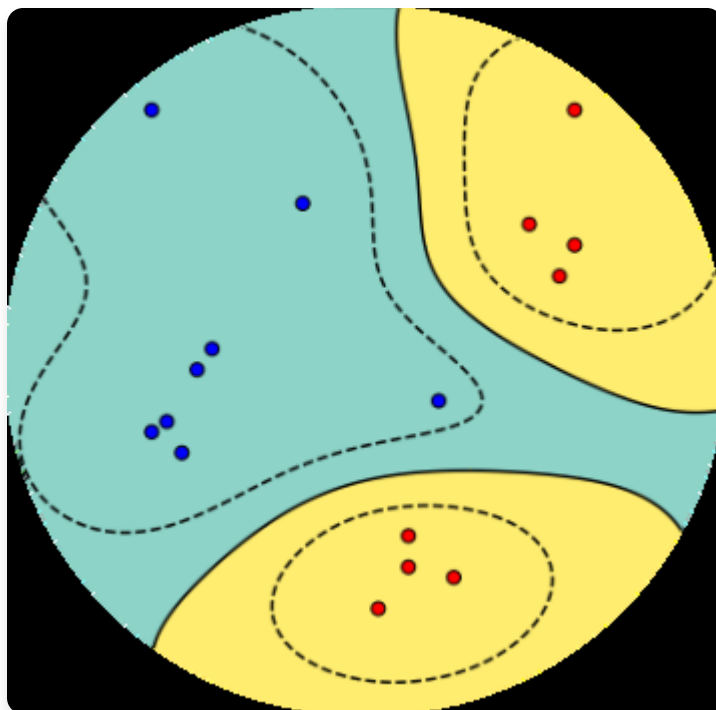
核密度估计

核密度估计由[Rosenblatt (1956)]和[Parzen (1962)]独立提出，是密度估计的非参数方法。给定N个i.i.d.样本，核密度估计器将底层分布表示为

$$\hat{p}_X(x) = (1/Nh) \sum_{n=1}^N k((x - x_n)/h), \quad (11.74)$$

其中k是核函数，即积分为1的非负函数， $h > 0$ 是平滑/带宽参数，它的作用类似于直方图中的bin大小。注意我们在数据集中的每个数据点 x_n 上放置一个核。常用的核函数是均匀分布和Gaussian分布。核密度估计与直方图密切相关，但通过选择合适的核，我们可以保证密度估计的平滑性。图11.13说明了对于给定的250个数据点数据集，直方图和核密度估计器（使用Gaussian形核）之间的差异。

12 支持向量机分类



在许多情况下，我们希望机器学习算法预测若干（离散）结果中的一个。例如，电子邮件客户端将邮件分类为个人邮件和垃圾邮件，这有两个结果。另一个例子是望远镜识别夜空中的物体是星系、恒星还是行星。通常有少数几个结果，更重要的是这些结果通常没有额外的结构。在本章中，我们考虑输出二进制值的预测器，即只有两个可能的结果。这种机器学习任务称为**二分类**。这与第9章形成对比，在第9章中我们考虑了具有连续值输出的预测问题。

对于二分类，标签/输出可以取的可能值集合是二进制的，在本章中我们用 $\{+1, -1\}$ 表示它们。换句话说，我们考虑形式为

$$f: \mathcal{R}^D \rightarrow \{+1, -1\} \quad (12.1)$$

的预测器。

结构的一个例子是如果结果是有序的，比如小号、中号和大号T恤的情况。

回顾第8章，我们将每个样本（数据点） $x[n]$ 表示为由 D 个实数组成的特征向量。标签通常分别称为正类和负类。需要注意的是，不应该从 $+1$ 类中推断出直观的正面属性。例如，在癌症检测任务中，患有癌症的患者通常被标记为 $+1$ 。原则上，可以使用任何两个不同的值，例如 $\{\text{True}, \text{False}\}$ 、 $\{0, 1\}$ 或 $\{\text{red}, \text{blue}\}$ 。二元分类问题已经得到了充分的研究，我们将其他方法的综述推迟到第12.6节。

我们介绍一种称为支持向量机(support vector machine, SVM)的方法，它解决二元分类任务。与回归类似，我们有一个监督学习任务，其中我们有一组样本 $x \in \mathcal{R}^D$ 以及相应的（二元）标签 $y[n] \in \{+1, -1\}$ 。给定由样本-标签对 $\{(x_1, y_1), \dots, (x_N, y_N)\}$ 组成的训练数据集，我们希望估计模型的参数，以获得最小的分类误差。类似于第9章，我们考虑线性模型，并将非线性隐藏在样本的变换 ϕ 中。我们将在第12.4节重新讨论 ϕ 。

SVM在许多应用中提供了最先进的结果，具有可靠的理论保证。我们选择使用SVM来说明二元分类有两个主要原因。首先，SVM允许以几何方式思考监督机器学习。虽然在第9章中我们从概率模型的角度考虑机器学习问题，并使用最大似然估计和贝叶斯推理来解决它，但在这里我们将考虑一种替代方法，从几何角度推理机器学习任务。它严重依赖于内积和投影等概念，我们在第3章中讨论过这些概念。我们发现SVM具有指导意义的第二个原因是，与第9章相比，SVM的优化问题没有解析解，因此我们需要使用第7章介绍的各种优化工具。

SVM的机器学习观点与第9章的最大似然观点略有不同。最大似然观点基于数据分布的概率视图提出模型，从中导出优化问题。相比之下，SVM观点从设计在训练期间要优化的特定函数开始，基于几何直觉。我们在第10章中已经看到了类似的东西，我们从几何原理导出了PCA(主成分分析)。在SVM的情况下，我们首先设计一个要在训练数据上最小化的损失函数，遵循经验风险最小化的原则。

让我们推导出对应于在样本-标签对上训练SVM的优化问题。直观上，我们想象二元分类数据，可以通过超平面分离，如图12.1所示。这里，每个样本 $x[n]$ (2维向量) 是一个二维位置($x[n]^{(1)}$ 和 $x[n]^{(2)}$)，相应的二元标签 y_n 是两个不同符号之一（橙色十字或蓝色圆盘）。“超平面”是机器学习中常用的词汇，我们在第2.8节已经遇到过超平面。超平面是维度为 $D-1$ 的仿射子空间（如果相应的向量空间的维度为 D ）。样本由两个类组成（有两个可能的标签），其特征（表示样本的向量的分量）以这样的方式排列，使我们能够通过画一条直线来分离/分类它们。



图12.1 二维数据示例，说明了数据的直觉，我们可以找到一个线性分类器来分离橙色十字和蓝色圆盘

图12.1 二维数据示例，说明了数据的直觉，我们可以找到一个线性分类器来分离橙色十字和蓝色圆盘

12.1 分离超平面

给定两个表示为向量 $x[i]$ 和 $x[j]$ 的样本，计算它们之间相似性的一种方法是使用内积 $\langle x[i], x[j] \rangle$ 。回顾第3.2节，内积与向量之间的角度密切相关。

在下面，我们将形式化找到两个类的线性分离器的想法。我们引入边距(margin)的概念，然后扩展线性分离器以允许样本落在“错误”的一边，产生分类误差。我们提出了形式化SVM的两种等价方法：几何视图（第12.2.4节）和损失函数视图（第12.2.5节）。我们使用拉格朗日乘数法推导SVM的对偶版本（第7.2节）。对偶SVM允许我们观察形式化SVM的第三种方法：根据每个类的样本的凸包(convex hulls)（第12.3.2节）。我们最后简要描述核函数(kernels)以及如何数值求解非线性核SVM优化问题。

两个向量的内积值取决于每个向量的长度（范数）。此外，内积允许我们严格定义几何概念，如正交性和投影。

许多分类算法背后的主要思想是在 $\mathbb{R}^D$ 中表示数据，然后对这个空间进行分割，理想情况下，具有相同标签（且没有其他示例）的示例位于同一分割中。

在二元分类的情况下，空间将被分为两部分，分别对应正类和负类。我们考虑一种特别方便的分割方式，即使用超平面(hyperplane)将空间（线性地）分割成两半。设示例 $x \in \mathbb{R}^D$ 是数据空间的一个元素。考虑函数

$$f: \mathbb{R}^D \rightarrow \mathbb{R} \quad (12.2a) \quad x \mapsto f(x) := \langle w, x \rangle + b, \quad (12.2b)$$

由 $w \in \mathbb{R}^D$ 和 $b \in \mathbb{R}$ 参数化。回忆第2.8节，超平面是仿射子空间。因此，我们将在二元分类问题中分离两个类别的超平面定义为

$$x \in \mathbb{R}^D : f(x) = 0. \quad (12.3)$$

超平面的图示如图12.2所示，其中向量 w 是超平面的法向量， b 是截距。我们可以通过选择超平面上的任意两个示例 x_a 和 x_b ，并证明它们之间的向量与 w 正交，来推导出 w 是方程(12.3)中超平面的法向量。用方程的形式表示：

$$f(x_a) - f(x_b) = \langle w, x_a \rangle + b - (\langle w, x_b \rangle + b) \quad (12.4a) = \langle w, x_a - x_b \rangle, \quad (12.4b)$$

图12.2 分离超平面方程(12.3)的标准表示方式。(a) 3D中的标准表示方式。(b) 为便于绘制，我们查看超平面在平面上的投影。

其中第二行通过内积的线性性（第3.2节）得到。由于我们选择 x_a 和 x_b 在超平面上，这意味着 $f(x_a) = 0$ 和 $f(x_b) = 0$ ，因此 $\langle w, x_a - x_b \rangle = 0$ 。回忆当两个向量的内积为零时，它们是正交的。因此，我们得出 w 与超平面上的任何向量都正交。

备注。回忆第2章，我们可以用不同的方式思考向量。在本章中，我们将参数向量 w 视为指示方向的箭头，即我们将 w 视为几何向量。相反，我们将示例向量 x 视为数据点（由其坐标表示），即我们将 x 视为相对于标准基的向量坐标。

◇

当给出测试示例时，我们根据示例所在的超平面的哪一侧将其分类为正类或负类。注意方程(12.3)不仅定义了一个超平面；它还定义了一个方向。换句话说，它定义了超平面的正侧和负侧。因此，为了对测试示例 x_{test} 进行分类，我

们计算函数 $f(x_{\text{test}})$ 的值，如果 $f(x_{\text{test}}) \geq 0$ 则将示例分类为+1，否则分类为-1。从几何角度思考，正示例位于超平面的”上方”，负示例位于超平面的”下方”。

在训练分类器时，我们希望确保具有正标签的示例位于超平面的正侧，即

$$\langle w, x_n \rangle + b \geq 0 \text{ 当 } y_n = +1 \text{ (12.5)}$$

具有负标签的示例位于负侧，即

$$\langle w, x_n \rangle + b < 0 \text{ 当 } y_n = -1 \text{ (12.6)}$$

参考图12.2了解正负示例的几何直觉。这两个条件通常用单一方程表示

$$y_n(\langle w, x_n \rangle + b) \geq 0 \text{ (12.7)}$$

当我们将方程(12.5)和(12.6)的两边分别乘以 $y_n = 1$ 和 $y_n = -1$ 时，方程(12.7)等价于(12.5)和(12.6)。

图12.3 可能的分离超平面。有许多线性分类器（绿线）可以将橙色叉号与蓝色圆盘分开。

12.2 原始支持向量机

基于点到超平面距离的概念，我们现在可以讨论支持向量机。对于线性可分的数据集 $\{(x_1, y_1), \dots, (x_N, y_N)\}$ ，我们有无限多个候选超平面（参考图12.3），因此有无限多个分类器可以解决我们的分类问题而不出现任何（训练）错误。为了找到唯一解，一个想法是选择能够最大化正负示例之间margin（间隔）的分离超平面。换句话说，我们希望正负示例被大的margin分离（第12.2.1节）。在下文中，我们计算示例与超平面之间的距离以推导margin。回忆到超平面上距离给定点（示例 x_n ）最近的点是通过正交投影（第3.8节）获得的。

12.2.1 Margin的概念

margin的概念直观上很简单：它是分离超平面到数据集中最近示例的距离，假设数据集是线性可分的。然而，当试图形式化这个concept时，

12.2 原始支持向量机

考虑一个超平面 $\langle w, x \rangle + b$ ，以及一个样本 $x[a]$ ，如图12.4所示。不失一般性，我们可以认为样本 $x[a]$ 位于超平面的正侧，即 $\langle w, x[a] \rangle + b > 0$ 。我们希望计算 $x[a]$ 到超平面的距离 $r > 0$ 。我们通过考虑 $x[a]$ 到超平面的正交投影(第3.8节)来实现这一点，记为 $x'[a]$ 。由于 w 垂直于超平面，我们知道距离 r 就是这个向量 w 的一个缩放。如果已知 w 的长度，那么我们可以使用这个缩放因子 r 来计算 $x'[a]$ 和 $x[a]$ 之间的绝对距离。为了方便，我们选择使用单位长度的向量(其范数为1)，通过将 w 除以其范数 $\|w\|$ 来获得。使用向量加法(第2.4节)，我们得到

$$x'[a] = x[a] + r(w / \|w\|) \quad (12.8)$$

另一种理解 r 的方式是，它是 $x[a]$ 在由 $w / \|w\|$ 张成的子空间中的坐标。现在我们已经将 $x[a]$ 到超平面的距离表示为 r ，如果我们选择 $x[a]$ 为最接近超平面的点，这个距离 r 就是间隔(margin)。

回想一下，我们希望正样本到超平面的距离超过 r ，负样本到超平面的距离(在负方向上)超过 r 。类似于将(12.5)和(12.6)合并成(12.7)，我们将这个目标表述为

$$y[n] (\langle w, x[n] \rangle + b) \geq r \quad (12.9)$$

换句话说，我们将样本至少距离超平面 r (在正负方向上)的要求合并成一个不等式。

由于我们只对方向感兴趣，我们在模型中添加一个假设，即参数向量 w 具有单位长度，即 $\|w\| = 1$ ，其中我们使用欧几里得范数 $\|w\| = \sqrt{w^T w}$ (第3.1节)。这个假设也允许对距离(12.8)有更直观的解释，因为它是长度为1的向量的缩放因子。

注释. 熟悉间隔其他表述的读者会注意到，我们定义的 $\|w\| = 1$ 与SVM的标准表述不同，例如Schölkopf和Smola(2002)提供的表述。在第12.2.3节中，我们将展示两种方法的等价性。◇

将三个要求收集到一个约束优化问题中，我们得到目标

$$\max r, b, r$$

$$\text{subject to } y_n \geq r, \|w\| = 1, r > 0 \quad (12.10)$$

这意味着我们希望最大化间隔 r ，同时确保数据位于超平面的正确一侧。

注释. 间隔的概念在机器学习中极其普遍。Vladimir Vapnik和Alexey Chervonenkis使用它来证明，当间隔很大时，函数类的“复杂度”很低，因此学习是可能的(Vapnik, 2000)。事实证明，这个概念对于理论上分析泛化误差的各种不同方法都很有用(Steinwart and Christmann, 2008; Shalev-Shwartz and Ben-David, 2014)。◇

12.2.2 间隔的传统推导

在前一节中，我们通过观察到我们只对 w 的方向而不是其长度感兴趣，导致假设 $\|w\| = 1$ ，从而推导出(12.10)。在本节中，我们通过做不同的假设来推导间隔最大化问题。我们不选择参数向量被归一化，而是为数据选择一个尺度。

我们选择这个尺度，使得预测器 $\langle w, x \rangle + b$ 在最接近的样本处的值为1。让我们也将数据集中最接近超平面的样本记为 $x[a]$ 。

图12.5与图12.4相同，除了现在我们重新缩放坐标轴，使得样本 $x[a]$ 恰好位于间隔上，即 $\langle w, x[a] \rangle + b = 1$ 。由于 $x'[a]$ 是 $x[a]$ 到超平面的正交投影，根据定义它必须位于超平面上，即

$$\langle w, x'[a] \rangle + b = 0 \quad (12.11)$$

将(12.8)替换到(12.11)中，我们得到

$$\langle w, x[a] - r(w/\|w\|) \rangle + b = 0 \quad (12.12)$$

利用内积的双线性(见第3.2节)，我们得到



图12.4 到超平面的距离： $x[a] = x'[a] + r(w/\|w\|)$

图12.4 到超平面的距离： $x[a] = x'[a] + r(w/\|w\|)$



图12.5 间隔的推导： $r = (\langle w, x[a] \rangle + b)/\|w\|$

图12.5 间隔的推导： $r = (\langle w, x[a] \rangle + b)/\|w\|$

我们当前考虑线性可分数据。

回想我们希望正样本到超平面的距离超过 r ，负样本到超平面的距离(在负方向上)超过 r 。

$$\langle w, x[a] \rangle + b - r = 0 \quad (12.13) \quad \|w\|$$

观察到第一项根据我们的比例假设为1，即 $\langle w, x[a] \rangle + b = 1$ 。从第[3.1]节的[(3.16)]我们知道 $\langle w, w \rangle = \|w\|^2$ 。因此，第二项简化为 $r\|w\|$ 。使用这些简化，我们得到

$$r = 1/\|w\| \quad (12.14)$$

这意味着我们用超平面的法向量 w 推导出了距离 r 。乍一看，这个方程式是反直觉的，因为我们似乎用向量 w 的长度推导出了到超平面的距离，但我们还不知道这个向量。思考这个问题的一种方法是将距离 r 视为一个临时变量，我们只在这个推导中使用它。因此，在本节的其余部分，我们将用 $1/\|w\|$ 表示到超平面的距离。在第[12.2.3]节中，我们将看到边距等于1的选择等价于我们在第[12.2.1]节中之前的假设 $\|w\| = 1$ 。

类似于获得[(12.9)]的论证，我们希望正例和负例距离超平面至少为1，这产生了条件

$$y[n] (\langle w, x[n] \rangle + b) \geq 1. \quad (12.15)$$

将边距最大化与样本需要在超平面正确一侧（基于其标签）的事实相结合，我们得到

$$\max 1 / \|w\| \quad (12.16) \quad w, b$$

$$\text{受约束 } y[n] (\langle w, x[n] \rangle + b) \geq 1 \text{ 对于所有 } n = 1, \dots, N. \quad (12.17)$$

我们通常不是像[(12.16)]中那样最大化范数的倒数，而是最小化平方范数。我们还经常包含一个常数1/2，它不影响最优的 w, b ，但在我们计算梯度时产生更整洁的形式。然后，我们的目标变为

$$\min 1/2 \|w\|^2 \quad (12.18) \quad w, b$$

$$\text{受约束 } y_n \geq 1 \text{ 对于所有 } n = 1, \dots, N. \quad (12.19)$$

方程[(12.18)]被称为 *hard margin SVM*。表达式“hard”的原因是因为该公式不允许任何违反边距条件的情况。我们将在第[12.2.4]节中看到，如果数据不是线性可分的，这个“hard”条件可以放宽以适应违规情况。

12.2.3 为什么我们可以将边距设置为1

在第[12.2.1]节中，我们论证了我们想要最大化某个值 r ，它表示最近样本到超平面的距离。在第[12.2.2]节中，我们缩放数据使得最近样本到超平面的距离为1。在本节中，我们关联这两个推导，并证明它们是等价的。

定理 12.1. 最大化边距 r ，其中我们考虑如[(12.10)]中的归一化权重，

$$\max r \quad (12.20) \quad w, b, r \mid \{z\} \text{ 边距}$$

$$\text{受约束 } y[n] (\langle w, x[n] \rangle + b) \geq r, \|w\| = 1, r > 0, \{z\} \mid \{z\} \text{ 数据拟合 归一化}$$

等价于缩放数据，使得边距为单位：

$$\min 1/2 \|w\|^2 \quad (12.21) \quad w, b \mid \{z\} \text{ 边距}$$

$$\text{受约束 } y_n \geq 1. \mid \{z\} \text{ 数据拟合}$$

证明 考虑[(12.20)]。由于平方对于非负参数是严格单调变换，如果我们在目标中考虑 r^2 ，最大值保持不变。由于 $\|w\| = 1$ ，我们可以用一个未归一化的新权重向量 w' 重新参数化方程，通过显式使用 $w' / \|w'\|$ 。我们得到

$$\max r^2 \quad (12.22) \quad w', b, r$$

$$\text{受约束 } y[n] \langle w' / \|w'\|, x[n] \rangle + b \geq r, r > 0.$$

方程[(12.22)]明确声明距离 r 是正的。因此，我们可以将第一个约束除以 r ，这产生

$$\max r^2 \quad (12.23) \quad w', b, r$$

$$\text{受约束 } y[n] \langle w' / r \|w'\|, x[n] \rangle + b/r \geq 1, r > 0 \mid \{z\} \mid \{z\} w' \quad b'$$

重命名参数为 w' 和 b' 。由于 $w' = w' / r \|w'\|$ ，重新排列 r 得到

$$\|w'\| = 1/(r \|w\|) = 1/r \cdot 1/\|w\|. \quad (12.24)$$

通过将此结果代入[(12.23)], 我们得到

$$\max 1/\|w'\| \quad (12.25) \quad w' \in \mathcal{B}$$

受约束 $y[n] \langle w', x[n] \rangle + b' \geq 1$.

最后一步是观察到最大化 $1/\|w'\|$ 产生与最小化 $1/2 \|w'\|^2$ 相同的解, 这完成了定理[12.1]的证明。

12.2.4 软边距SVM：几何视角

在数据不是线性可分的情况下, 我们可能希望允许一些样本落在边距区域内, 甚至在超平面的错误一侧, 如图[12.6]所示。

允许一些分类错误的模型称为 *soft margin SVM*。在本节中, 我们使用几何论证推导出结果优化问题。在第[12.2.5]节中, 我们将使用 *loss function* 的思想推导出等价的优化问题。使用 *Lagrange* 乘数 (第[7.2]节), 我们将推导出对偶优化

图12.6 (a) 线性可分和 (b) 非线性可分数据。

$x^{(2)}$ $x^{(1)}$

$x^{(1)}$ $x^{(1)}$ (a) 线性可分数据, 具有大边距 (b) 非线性可分数据

SVM的问题在第12.3节中讨论。这个对偶优化问题使我们能够观察到SVM的第三种解释: 作为一个超平面, 它平分对应于正样本和负样本的凸包之间的线段 (第12.3.2节)。

关键的几何思想是引入一个松弛变量 $\xi[n]$, 对应每个样本-标签对 $(x[n], y[n])$, 允许特定样本位于边界内或甚至在超平面的错误一侧 (参见

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

380 支持向量机分类

图12.7 软边界SVM允许样本位于边界内或超平面的错误一侧。松弛变量 ξ 衡量正样本 x_+ 到正边界超平面 $\langle w, x \rangle + b = 1$ 的距离, 当 x_+ 位于错误一侧时为0。

图12.7)。我们从边界中减去 $\xi[n]$ 的值, 约束 $\xi[n]$ 为非负。为了鼓励样本的正确分类, 我们将 $\xi[n]$ 添加到目标函数中

$$\min_{w, b, \xi} (1/2) \|w\|^2 + C \sum_{n=1}^N \xi[n] \quad (12.26a)$$

$$\text{subject to } y_n \geq 1 - \xi[n] \quad (12.26b) \quad \xi[n] \geq 0 \quad (12.26c)$$

对于 $n = 1, \dots, N$ 。与硬边界SVM的优化问题(12.18)相比, 这个被称为 *软边界SVM*。参数 $C > 0$ 权衡边界大小和松弛总量。这个参数被称为 *正则化参数*, 因为正如我们将在下一节中看到的, 目标函数(12.26a)中的边界项是一个正则化项。边界项 $\|w\|^2$ 被称为 *正则化项*, 在许多数值优化书籍中, 正则化参数与此项相乘 (第8.2.3节)。这与我们在本节中的公式化形成对比。这里 C 的大值意味着低正则化, 因为我们给松弛变量更大的权重, 因此对不位于边界正确一侧的例子给予更多优先级。

备注。在软边界SVM的公式化(12.26a)中， w 被正则化，但 b 没有被正则化。我们可以通过观察正则化项不包含 b 来看这一点。未正则化的项 b 使理论分析复杂化(Steinwart and Christmann, 2008, 第1章)并降低计算效率(Fan et al., 2008)。有这种正则化的替代参数化，这就是为什么(12.26a)也经常被称为C-SVM。◇

12.2.5 软边界SVM：损失函数视角

让我们考虑推导SVM的不同方法，遵循经验风险最小化原则（第8.2节）。对于SVM，我们

Mathematics for Machine Learning草稿（2023-10-18）。反馈：<https://mml-book.com> 12.2 原始支持向量机 381

选择超平面作为假设类，即

$$f(x) = \langle w, x \rangle + b. \quad (12.27)$$

我们将在本节中看到边界对应于正则化项。剩下的问题是，什么是损失函数？与第9章不同，我们考虑回归问题（预测器的输出是实数），在本章中，我们考虑二元分类问题（预测器的输出是两个标签 $\{+1, -1\}$ 中的一个）。因此，每个单独样本-标签对的误差/损失函数需要适合二元分类。例如，用于回归的平方损失(9.10b)不适合二元分类。

备注。二元标签之间的理想损失函数是计算预测和标签之间不匹配的数量。这意味着对于应用于样本 $x[n]$ 的预测器 f ，我们将输出 $f(x[n])$ 与标签 $y[n]$ 进行比较。如果它们匹配，我们定义损失为零，如果不匹配则为一。这表示为 $1(f(x[n]) \neq y[n])$ ，称为零一损失。不幸的是，零一损失导致寻找最佳参数 w, b 的组合优化问题。组合优化问题（与第7章讨论的连续优化问题相比）通常更难解决。◇

对应于SVM的损失函数是什么？考虑预测器 $f(x[n])$ 的输出与标签 $y[n]$ 之间的误差。损失描述了在训练数据上产生的误差。推导(12.26a)的等价方法是使用合页损失

$$\ell(t) = \max\{0, 1 - t\} \text{ 其中 } t = yf(x) = y(\langle w, x \rangle + b). \quad (12.28)$$

如果 $f(x)$ 在超平面的正确一侧（基于对应标签 y ），并且距离超过1，这意味着 $t \geq 1$ ，合页损失返回零值。如果 $f(x)$ 在正确一侧但太靠近超平面（ $0 < t < 1$ ），样本 x 在边界内，合页损失返回正值。当样本在超平面的错误一侧（ $t < 0$ ）时，合页损失返回更大的值，并线性增加。换句话说，一旦我们比边界更接近超平面，即使预测是正确的，我们也要付出惩罚，并且惩罚线性增加。表达合页损失的另一种方式是将其视为两个线性片段

$$\ell(t) = \begin{cases} 0 & \text{if } t \geq 1 \\ 1 - t & \text{if } t < 1 \end{cases} \quad (12.29)$$

如图12.8所示。对应于硬边界SVM 12.18的损失定义为

$$\ell(t) = \begin{cases} 0 & \text{if } t \geq 1 \\ \infty & \text{if } t < 1 \end{cases} \quad (12.30)$$

第12章 支持向量机分类

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[382] [[支持向量机分类]]

图 12.8

hinge损失是零一损失的凸上界。这种损失可以解释为不允许任何样本出现在边界内部(margin)。

对于给定的训练集 $\{(x_1, y_1), \dots, (x[N], y[N])\}$ ，我们寻求最小化总损失，同时用 ℓ_2 正则化对目标函数进行正则化（参见第8.2.3节）。使用hinge损失(12.28)给出无约束优化问题

$$\frac{1}{2} \|w\|^2 + C \sum_{n=1}^N \max\{0, 1 - y_n\}. \quad (12.31)$$

第一项在(12.31)中称为**正则化项**或**正则化器**（参见第8.2.3节），第二项称为**损失项**或**误差项**。回顾第12.2.4节，项 $\frac{1}{2} \|w\|^2$ 直接来源于边界(margin)。换句话说，边界最大化可以解释为**正则化**。

原则上，(12.31)中的无约束优化问题可以直接用第7.1节描述的（次）梯度下降方法求解。要看到(12.31)和(12.26a)是等价的，观察到hinge损失(12.28)本质上由两个线性部分组成，如(12.29)所表达的。考虑单个样本-标签对的hinge损失(12.28)。我们可以等价地用一个松弛变量 ξ 和两个约束来替换对 t 的hinge损失最小化。用方程形式表示：

$$\min \max\{0, 1 - t\} \quad (12.32)$$

等价于

$$\min_{\xi, t} \xi \quad (12.33)$$

subject to $\xi \geq 0, \xi \geq 1 - t$.

通过将此表达式代入(12.31)并重新排列其中一个约束，我们正好得到软边界SVM (12.26a)。

备注。 让我们将本节中损失函数的选择与第9章线性回归的损失函数进行对比。回顾第9.2.1节，为了寻找最大似然估计器，我们通常最小化负对数似然。此外，由于具有高斯噪声的线性回归的似然项是高斯的，每个样本的负对数似然是平方误差函数。平方误差函数是寻找最大似然解时最小化的损失函数。◇

12.3 对偶支持向量机

前面几节中以变量 w 和 b 表示的 SVM 描述被称为原始 SVM。回顾我们考虑输入 $x \in \mathbb{R}^D$ ，有 D 个特征。由于 w 与 x 具有相同的维度，这意味着优化问题的参数数量（ w 的维度）随特征数量线性增长。

在下面，我们考虑一个等价的优化问题（所谓的对偶视图），它独立于特征数量。相反，参数数量随训练集中样本数量增加。我们在第 10 章中看到了类似的想法，其中我们以不随特征数量缩放的方式表达学习问题。这对于特征数量多于训练数据集中样本数量的问题很有用。对偶 SVM 还有一个额外的优势，即可以轻松应用核(kernel)，正如我们将在本章末尾看到的。“对偶”一词经常出现在数学文献中，在这个特定情况下，它指的是凸对偶性。以下小节本质上是凸对偶性的应用，我们在第 7.2 节中讨论过。

12.3.1 通过拉格朗日乘数的凸对偶性

回顾原始软边界 SVM (12.26a)。我们称对应于原始 SVM 的变量 w 、 b 和 ξ 为原始变量。我们使用 $\alpha[n] \geq 0$ 作为对应于约束 (12.26b) 的拉格朗日乘数，即样本被正确分类，使用 $\gamma[n] \geq 0$ 作为对应于松弛变量非负约束的拉格朗日乘数；参见 (12.26c)。拉格朗日函数为

$$L(w, b, \xi, \alpha, \gamma) = 1/2 \|w\|^2 + C \sum_{n=1}^N \xi[n] - \sum_{n=1}^N \alpha[n] - \sum_{n=1}^N \gamma[n] \xi[n].$$

通过分别对三个原始变量 w 、 b 和 ξ 对拉格朗日函数 (12.34) 求偏导数，我们得到

$$\partial L / \partial w = w - \sum_{n=1}^N \alpha[n] y[n] x[n], \quad (12.35)$$

$$\partial L / \partial b = - \sum_{n=1}^N \alpha[n] y[n], \quad (12.36)$$

$$\partial L / \partial \xi[n] = C - \alpha[n] - \gamma[n]. \quad (12.37)$$

我们现在通过将这些偏导数设为零来寻找拉格朗日函数的最大值。通过将 (12.35) 设为零，我们发现

$$w = \sum_{n=1}^N \alpha[n] y[n] x[n], \quad (12.38)$$

这是**表示定理**(representer theorem)的一个特例(Kimeldorf and Wahba, 1970)。方程 (12.38) 表明原始问题中的最优权重向量是样本 $x[n]$ 的线性组合。回顾第 2.6.1 节，这意味着优化问题的解

[解的位置] [在训练数据的生成空间内。此外，通过]

[最小化] [将 [(12.36)] 设为零获得的约束意味着最优权重向量是一个仿射]

[经验风险位于] [样本的组合。表示定理对于正则化经验风险最小化的非常一般的设置都成立 [(Hof-]]

[子空间] [[mann et al., 2008]; [Argyriou and Dinuzzo, 2014)。] 该定理有更]

[(第 [2.4.3 节]] [由示例定义的]

[示例。] 一般的版本 [(Sch ¨) [olkopf et al., 2001),] 以及其存在的必要充分

条件可以在 [Yu et al. (2013)] 中找到。

注释。表示定理 [(12.38)] 也为”支持向量机”这个名称提供了解释。

对应参数 $\alpha[n] = 0$ 的示例 $x[n]$ 对解 w 没有任何贡献。其他 $\alpha[n] > 0$ 的示例被称为*支持向量*，因为它们”支持”超平面。[◇]

[支持向量]

通过将 w 的表达式代入拉格朗日函数 [(12.34)]，我们得到对偶形式

$$\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N D(X_i X_j X_i X_j \xi, \alpha, \gamma) = y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle - y_i y_j \alpha_i \alpha_j x_i^2 \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N \gamma_{kl} \xi_{kl} \quad (12.39)$$

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N X_i X_j X_k X_l C \xi_{kl} - b y_i \alpha_i + \alpha_i - \alpha_i \xi_i - \gamma_i \xi_i \quad (12.39)$$

(12.39)

注意这里不再有涉及原始变量 w 的项。

通过将 [(12.36)] 设为零，我们得到 $\sum_{i=1}^N \alpha_i = 0$ 。因此，涉及 b 的项也消失了。 $y_i \alpha_i = 0$ $\sum_{i=1}^N \alpha_i = 0$ $\sum_{i=1}^N \alpha_i$

回想内积是对称的且

[《机器学习数学基础》草稿 (2023-10-18)。反馈：] [https://mml-book.com][。] [12.3 对偶支持向量机] [385]

双线性的（见第 [3.2] 节）。] 因此，[(12.39)] 中的前两项针对相同的对象。这些项（蓝色部分）可以简化，我们得到拉格朗日函数

$$\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N D(X_i X_j X_i X_j \xi, \alpha, \gamma) = - y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle + \alpha_i + (C - \alpha_i - \gamma_i) \xi_i \quad (12.40)$$

(12.40)

该方程的最后一项是包含松弛变量 ξ_i 的所有项的集合。通过将 [(12.37)] 设为零，我们看到 [(12.40)] 中的最后一项也为零。此外，通过使用相同的方程并回忆拉格朗日乘数 γ_i 是非负的，我们得出 $\alpha_i \leq C$ 。我们现在得到 SVM 的对偶优化问题，它完全用拉格朗日乘数 α_i 表示。根据拉格朗日对偶性（定义 [7.1]），我们最大化对偶问题。这等价于最小化负对偶问题，这样我们得到*对偶 SVM*

[对偶 SVM]

$$\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \min X_i X_j y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle - \alpha_i \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N \gamma_{kl} \xi_{kl} \quad (12.41)$$

$\sum_{i=1}^N \alpha_i = 0$ subject to $\sum_{i=1}^N y_i \alpha_i = 0$ $\sum_{i=1}^N \alpha_i = 0$ $\sum_{i=1}^N \alpha_i$

$0 \leq \alpha_i \leq C$ for all $i = 1, \dots, N$.

[(12.41)] 中的等式约束是通过将 [(12.36)] 设为零得到的。不等式约束 $\alpha_i \geq 0$ 是对不等式约束的拉格朗日乘数施加的条件（第 [7.2] 节）。] 不等式约束 $\alpha_i \leq C$ 在前面的段落中讨论过。

SVM 中的不等式约束集合被称为“盒约束”，因为它们将拉格朗日乘数向量 $\alpha = [\alpha_1, \dots, \alpha_N] \in \mathbb{R}^N$ 限制在每个轴上的 0 和 C 定义的盒子内。这些轴对齐的盒子在数值求解器中特别有效 (Dost' al, 2009,] 第5章)。

[[事实证明]]

一旦我们得到对偶参数 α ，我们可以通过使用表示定理 (12.38) 恢复原始参数 w 。让我们称最优原始参数为 w 。然而，仍然存在如何获得参数 b 的问题。考虑一个恰好位于

[[其对偶参数位于] [边界上的]]

边界上的样本 $x[n]$ ，即 $\langle w, x[n] \rangle + b = y[n]$ 。回忆 $y[n]$ 要么是 +1 要么是 -1。因此，唯一的未知数是 b ，它可以通过以下方式计算 $b = y[n] - \langle w, x[n] \rangle$ 。这是

$b = y[n] - \langle w, x[n] \rangle$ [使用 Karush Kuhn Tucker] [条件推导的，例如 w, x 。(12.42) [在

注释。原则上，可能没有恰好位于边界上的样本。在这种情况下，我们应该对所有支持向量计算 $|y[n] - \langle w, x[n] \rangle|$ ，并取这个绝对值差的中位数作为 b 。

[[[Smola (2002).]]]

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由 Cambridge University Press 出版 (2020).]

[386] [[支持向量机分类]]

图 12.9 [凸包]

凸包。[(a)] 点的凸包，其中一些点位于边界内；[(b)] 正样本和负样本周围的凸包。

[c]

[d]

[(a) 凸包。] [(b) 正样本（蓝色）和负样本（橙色）周围的凸包。两个凸集之间的距离是差向量] [c] [-] [d] [的长度。]

b 的值。相关推导可以在 <http://fouryears.eu/2012/06/07/the-svm-bias-term-conspiracy/> 中找到。[◇]

12.3.2 对偶 SVM：凸包视角

获得对偶 SVM 的另一种方法是考虑替代的几何论证。考虑具有相同标签的样本集合 $x[n]$ 。我们希望构建一个包含所有样本的凸集，使其是可能的最小集合。这被称为凸包，如图 [12.9] 所示。

让我们首先建立关于点的凸组合的一些直觉。

考虑两个点 x_1 和 x_2 以及对应的非负权重

$\alpha_1, \alpha_2 \geq 0$ ，使得 $\alpha_1 + \alpha_2 = 1$ 。方程 $\alpha_1 x_1 + \alpha_2 x_2$ 描述了

x_1 和 x_2 之间直线上的每个点。考虑当我们添加

第三个点 x_3 以及权重 $\alpha \geq 0$ 使得 $\alpha = 1$ 时会发生什么。

$\sum_{n=1}^3 \alpha_n = 1$

这三个点 x_1, x_2, x_3 的凸组合跨越一个二维区域。这个区域的 *凸包* (convex hull) 是由

对应于每对点的边形成的三角形。当我们添加更多点时，

当点的数量变得大于维数时，一些点将位于凸包内部，正如我们在

图 [12.9(a)] 中看到的那样。

一般来说，构建凸包可以通过引入

对应于每个样本 $x[n]$ 的非负权重 $\alpha[n] \geq 0$ 来完成。然后

凸包可以描述为集合

$\{ \sum_{n=1}^N \alpha_n x[n] \mid \sum_{n=1}^N \alpha_n = 1, \alpha_n \geq 0 \}$

$\text{conv}(\{x_1, \dots, x_N\}) = \{ \sum_{n=1}^N \alpha_n x_n \mid \sum_{n=1}^N \alpha_n = 1, \alpha_n \geq 0 \}$ (12.43)

$\sum_{n=1}^N \alpha_n = 1$

[《机器学习数学》草稿 (2023-10-18)。反馈：] <https://mml-book.com> [12.3 对偶支持向量机] [[387]]

对于所有 $n = 1, \dots, N$ 。如果对应于正类和负类的两个点云是分离的，那么凸包不重叠。给定训练数据 $(x_1, y_1), \dots, (x_N, y_N)$ ，我们形成两个凸包，分别对应于正类和负类。我们选择一个点 c ，它在正样本集合的凸包中，并且最接近负类分布。类似地，我们在负样本集合的凸包中选择一个点 d ，并且最接近

正类分布；见图 [12.9(b)]。我们定义 d 和 c 之间的差向量为

$w := c - d$. (12.44)

如前述情况选择点 c 和 d ，并要求它们彼此最接近，这等价于最小化 w 的长度/范数，因此我们得到相应的优化问题

$$\arg \min \|w\| = \arg \min \|w\|_1. \quad (12.45)$$

[2]

由于 c 必须在正凸包中，它可以表示为正样本的凸组合，即，对于非负系数 α

[n]

$$c \in \text{conv}(\{x_n : y_n = +1\}). \quad (12.46)$$

[n]

[n] : $y_n = +1$

在 [(12.46)] 中，我们使用记号 $n : y_n = +1$ 来表示 $y_n = +1$ 的索引 n 的集合。类似地，对于负标签的样本，我们得到

$$d \in \text{conv}(\{x_n : y_n = -1\}). \quad (12.47)$$

[n]

[n] : $y_n = -1$

通过将 [(12.44)], [(12.46)] 和 [(12.47)] 代入 [(12.45)]，我们得到目标函数

[2]

$$\min \sum_{n : y_n = +1} \alpha_n \|x_n - d\|_1 + \sum_{n : y_n = -1} \alpha_n \|x_n - c\|_1. \quad (12.48)$$

设 α 为所有系数的集合，即 α 和 α 的连接。

回想我们要求对于每个凸包，它们的系数和为一，

$$\sum_{n : y_n = +1} \alpha_n = 1 \text{ 和 } \sum_{n : y_n = -1} \alpha_n = 1. \quad (12.49)$$

[n]

[n] : $y_n = +1$ 和 [n] : $y_n = -1$

这意味着约束

[N]

$$\sum_{n : y_n = +1} \alpha_n = 0. \quad (12.50)$$

[n] : $y_n = -1$

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由剑桥大学出版社出版 (2020).]

[388] [[支持向量机分类]]

这个结果可以通过展开各个类别看到

[N]

$$\sum_{n=1}^N \sum_{y \in \mathcal{Y}} \alpha_n (+1)^y = \sum_{n=1}^N \alpha_n (12.51a)$$

$$\sum_{n=1}^N \alpha_n (-1)^y = \sum_{n=1}^N \alpha_n (-1)^y (12.51b)$$

$$\sum_{n=1}^N \alpha_n = \sum_{n=1}^N \alpha_n = 1 - 1 = 0 \quad (12.51b)$$

$$\sum_{n=1}^N \alpha_n (-1)^y = \sum_{n=1}^N \alpha_n (-1)^y = 1 - 1 = 0$$

目标函数 [(12.48)] 和约束 [(12.50)], 连同

$\alpha_n \geq 0$ 的假设, 给出了一个约束 (凸) 优化问题。这个优化问题可以证明与

硬间隔 SVM 的对偶问题相同 [(Bennett and Bredensteiner, 2000a)]。

备注。为了得到软间隔对偶, 我们考虑 *缩减包(reduced hull)*。缩减包类似于凸包, 但对系数 α 的大小有限制。 α 元素的最大可能值

限制了凸包可以取的大小。换句话说,

α 的界限将凸包缩小到更小的体积 [(Bennett and

[Bredensteiner, 2000b)]。[◇]

12.4 核函数

考虑对偶 SVM 的表述 [(12.41)]。注意目标函数中的内积只发生在样本 $x[i]$ 和 $x[j]$ 之间。

样本和参数之间没有内积。

因此，如果我们考虑一组特征 $\phi(x[i])$ 来表示 $x[i]$ ，对偶 SVM 中唯一的

变化将是替换内积。这种模块化，即分类方法（SVM）的选择和

特征表示 $\phi(x)$ 的选择可以分别考虑，

为我们独立探索这两个问题提供了灵活性。在

本节中，我们讨论表示 $\phi(x)$ 并简要介绍

核函数的概念，但不深入技术细节。

由于 $\phi(x)$ 可能是非线性函数，我们可以使用 SVM（它

假设线性分类器）来构造在

样本 $x[n]$ 中非线性的分类器。这提供了第二条途径，除了软

间隔之外，让用户处理非线性可分的数据集。事实证明，有许多算法和统计方法具有

我们在对偶 SVM 中观察到的这种性质：唯一的内积

是发生在样本之间的那些。我们不是显式定义

非线性特征映射 $\phi(\cdot)$ 并计算样本 $x[i]$ 和 $x[j]$ 之间的结果内积，而是定义 $x[i]$ 和 $x[j]$ 之间的相似性函数 $k(x[i], x[j])$ 。对于某类相似性函数，称为核函数(kernels)，

相似性函数 隐式定义非线性特征映射 $\phi(\cdot)$ 。核函数根据定义是函数 $k: X \times X \rightarrow \mathbb{R}$ ，对于这些函数存在

kernel function can] 一个 Hilbert 空间 H 和 $\phi: X \rightarrow H$ 一个特征映射，使得[be very general and]

[are not necessarily] $k(x[i], x[j]) = \langle \phi(x[i]), \phi(x[j]) \rangle$. (12.52) [H] [D] [restricted to] [R] [.]

[Draft (2023-10-18) of “Mathematics for Machine Learning” . Feedback:] <https://mml-book.com/> [.] [12.4 Kernels] [389]

图 12.10 使用不同 kernels 的 SVM。注意虽然决策边界是非线性的，但[underlying problem]正在解决的是线性分离的[feature]问题[Second] [Second] [hyperplane (albeit with a nonlinear kernel).]

[First feature] [First feature]

a. 使用线性 kernel 的 SVM (b) 使用 RBF kernel 的 SVM

[feature] [feature]

[Second] [Second]

[First feature] [First feature]

c. 使用多项式(2次)kernel的SVM (d) 使用多项式(3次)kernel的SVM

每个kernel k 都有一个唯一的 reproducing kernel Hilbert space 与之关联 [(Aronszajn, 1950); (Berlinet and Thomas-Agnan, 2004).] 在这种唯一关联中, $\phi(x) = k(\cdot, x)$ 被称为 *canonical feature map*。

从内积到kernel函数[(12.52)]的推广被称为 *kernel trick* [(Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004),] 因为它隐藏了显式的非线性特征映射。

从内积或将 $k(\cdot, \cdot)$ 应用到数据集得到的矩阵 $K \in \mathbb{R}^{[N] \times [N]}$ 被称为 *Gram matrix*, 通常简称为 *kernel matrix*。Kernels必须是对称且正半定函数, 以便每个kernel matrix K 都是对称且正半定的(Section [3.2.3].)

$$\forall z \in \mathbb{R}^{[N]} : z^\top K z \geq 0. \quad (12.53)$$

对于多变量实值数据 $x[i] \in \mathbb{R}^{[D]}$ 的一些流行kernel示例包括polynomial kernel、Gaussian radial basis function kernel和rational quadratic kernel [(Schölkopf and Smola, 2002); (Rasmussen and Williams, 2006).]

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[and Williams, 2006).] 图[12.10]展示了不同kernels对示例数据集上分离超平面的影响。注意我们仍在求解超平面, 也就是说, 函数的假设类仍然是线性的。非线性表面是由于kernel函数造成的。

备注。不幸的是, 对于机器学习初学者来说, “kernel” 这个词有多重含义。在本章中, “kernel” 一词来自reproducing kernel Hilbert space (RKHS)的概念[(Aronszajn, 1950); (Saitoh, 1988).]我们已经讨论了线性代数中kernel的概念(Section [2.7.3].)其中kernel是null space的另一种说法。在机器学习中”kernel”的第三种常见用法是kernel density estimation中的smoothing kernel (Section [11.5].)◇

由于显式表示 $\phi(x)$ 在数学上等价于kernel表示 $k(x[i], x[j])$, 实践者通常会设计kernel函数, 使其比显式特征映射之间的内积计算更高效。例如, 考虑polynomial kernel [(Schölkopf and Smola, 2002).]当输入维度很大时, 显式展开中的项数增长得非常快(即使对于低次多项式)。kernel函数每个输入维度只需要一次乘法, 这可以提供显著的计算节省。另一个例子是Gaussian radial basis function kernel [(Schölkopf and Smola, 2002); (Rasmussen and Williams, 2006).]其对应的特征空间是无限维的。在这种情况下, 我们无法显式表示特征空间, 但[The choice of]仍可以使用kernel计算一对样本之间的相似性。

kernel trick的另一个有用方面是不需要[kernel, as well as]原始数据已经表示为多变量实值[the parameters of]数据。注意内积定义在函数[the kernel, is often] $\phi(\cdot)$ 的输出上, 但不限制输入为实数。因此, 函数[chosen using nested] $\phi(\cdot)$ 和kernel函数 $k(\cdot, \cdot)$ 可以定义在任何对象上, 例如[cross-validation]集合、序列、字符串、图和分布[(Ben-Hur et al., 2008); (Section [8.6.1].)][(Gertner et al., 2008); (Shi et al., 2009); (Sriperumbudur et al., 2010; Vishwanathan et al., 2010).]

12.5 数值解

我们通过研究如何根据第7章提出的概念来表达本章中导出的问题来结束对SVM的讨论。我们考虑两种不同的方法来找到SVM的最优解。首先我们考虑SVM的损失函数观点[8.2.2]并将其表达为无约束优化问题。然后我们将primal和dual SVM的约束版本表达为标准形式的二次规划[7.3.2.]

考虑SVM的损失函数观点[(12.31).]这是一个凸无约束优化问题，但hinge loss [(12.28)]不可

[Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback:] <https://mml-book.com>][.] [12.5 Numerical Solution] [391]

微分。因此，我们应用subgradient方法来解决它。然而，hinge loss几乎在所有地方都可微分，除了在hinge $t = 1$ 的单一。在这一点上，梯度是位于0和-1之间的一组可能值。因此，hinge loss的subgradient g 由下式给出：

$$g(t) = \begin{cases} 1 & \text{if } t < -1 \\ -1, 0 & \text{if } t = 1 \\ 0 & \text{if } t > 1 \end{cases} \quad (12.54)$$

使用这个子梯度，我们可以应用第7.1节中介绍的优化方法。

原始SVM和对偶SVM都导致凸二次规划问题（约束优化）。注意(12.26a)中的原始SVM具有与输入样本维数 D 大小相同的优化变量。(12.41)中的对偶SVM具有与样本数量 N 大小相同的优化变量。

为了将原始SVM表示为二次规划的标准形式(7.45)，我们假设使用点积(3.5)作为内积。我们重新整理原始SVM的方程(12.26a)，使得优化变量都在右边，约束的不等式符合标准形式。这产生了优化问题：

回忆第3.2节中我们使用点积这个术语来表示欧几里得向量空间上的内积。

$$\min_{w, b, \xi} \frac{1}{2} \|w\|^2 + C \sum_{n=1}^N \xi_n \quad (12.55)$$

$$\text{约束条件: } -y_n x_n^T w - y_n b - \xi_n \leq -1 \quad -\xi_n \leq 0$$

$n = 1, \dots, N$ 。通过将变量 w, b, x_n 连接成单个向量，并仔细收集项，我们得到软间隔SVM的以下矩阵形式：

$$\min_{w, b, \xi} \frac{1}{2} \begin{bmatrix} w & b & \xi \end{bmatrix}^T \begin{bmatrix} I_D & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w \\ b \\ \xi \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} w \\ b \\ \xi \end{bmatrix} \quad (12.56)$$

$$\text{约束条件: } \begin{bmatrix} -YX & -y & -I_N \end{bmatrix} \begin{bmatrix} w \\ b \\ \xi \end{bmatrix} \leq \begin{bmatrix} -1 & 0 & 0 \end{bmatrix} \quad (12.56)$$

在前面的优化问题中，最小化是关于参数 $[w^T, b, \xi^T] \in \mathbb{R}^{D+1+N}$ ，我们使用记号： I_m 表示大小为 $m \times m$ 的单位矩阵， $0_{m,n}$ 表示大小为 $m \times n$ 的零矩阵， $1_{m,n}$ 表示大小为 $m \times n$ 的全一矩阵。此外， y 是标签向量 $[y_1, \dots, y_N]^T$ ， $Y = \text{diag}(y)$

© 2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

Classification with Support Vector Machines

是一个 $N \times N$ 矩阵，其对角线元素来自 y ， $X_{N \times D} \in \mathbb{R}$ 是通过连接所有样本得到的矩阵。

我们可以类似地对SVM的对偶版本(12.41)进行项的收集。为了将对偶SVM表示为标准形式，我们首先必须表达kernel矩阵 K ，使得每个元素为 $K_{ij} = k(x_i, x_j)$ 。如果我们有显式的特征表示 x_i ，那么我们定义 $K_{ij} = \langle x_i, x_j \rangle$ 。

为了记号方便，我们引入一个除对角线外都为零的矩阵，在其对角线上存储标签，即 $Y = \text{diag}(y)$ 。

对偶SVM可以写为：

$$\min_{\alpha} \frac{1}{2} \alpha^T Y K Y \alpha - 1_{\{N,1\}}^T \alpha \quad (12.57)$$

$$\text{约束条件: } \begin{bmatrix} y^T \\ I_N \\ -I_N \end{bmatrix} \alpha \leq \begin{bmatrix} 0 \\ C1_{\{N,1\}} \\ 0_{\{N,1\}} \end{bmatrix}$$

备注。 在第7.3.1节和第7.3.2节中，我们介绍了约束的标准形式为不等式约束。我们将对偶SVM的等式约束表示为两个不等式约束，即

$$Ax = b \text{ 被替换为 } Ax \leq b \text{ 和 } Ax \geq b \quad (12.58)$$

凸优化方法的特定软件实现可能提供表达等式约束的能力。◇

由于SVM有许多不同的可能视角，因此有许多方法来解决所得的优化问题。这里提出的方法，即将SVM问题表示为标准凸优化形式，在实践中并不经常使用。SVM求解器的两个主要实现是Chang and Lin (2011)（开源）和Joachims (1999)。由于SVM有清晰且定义良好的优化问题，许多基于数值优化技术的方法(Nocedal and Wright, 2006)都可以应用(Shawe-Taylor and Sun, 2011)。

12.6 进一步阅读

SVM是研究二元分类的许多方法之一。其他方法包括perceptron、logistic回归、Fisher判别、最近邻、naive Bayes和random forest (Bishop, 2006; Murphy, 2012)。关于离散序列上SVM和kernel的简短教程可以在Ben-Hur et al. (2008)中找到。SVM的发展与第8.2节讨论的经验风险最小化密切相关。

因此，SVM具有强理论性质(Vapnik, 2000; Steinwart and Christmann, 2008)。关于kernel方法的书籍(Schölkopf and Smola, 2002)包含了support vector machine的许多细节以及

Draft (2023-10-18) of “Mathematics for Machine Learning”. Feedback: <https://mml-book.com>. 12.6 Further Reading 393

如何优化它们。一本关于kernel方法的更广泛的书籍(Shawe-Taylor and Cristianini, 2004)也包含了不同机器学习问题的许多线性代数方法。

对偶SVM的另一种推导可以使用Legendre-Fenchel变换的思想(第7.3.3节)来获得。该推导分别考虑SVM无约束公式(12.31)的每一项，并计算它们的凸共轭(Rifkin and Lippert, 2007)。对SVM的泛函分析视角（也是正则化方法视角）感兴趣的读者可参考Wahba (1990)的工作。kernel的理论阐述(Aronszajn, 1950; Schwartz, 1964; Saitoh, 1988; Manton and Amblard, 2015)需要线性算子的基本基础(Akhiezer and Glazman, 1993)。kernel的思想已被推广到Banach空间(Zhang et al., 2009)和Krein空间(Ong et al., 2004; Loosli et al., 2016)。

观察到hinge损失有三种等价表示，如(12.28)和(12.29)所示，以及(12.33)中的约束优化问题。公式(12.28)在比较时经常使用

ing the SVM loss function with other loss functions [(Steinwart, 2007).]

两段式表述 [(12.29)] 便于计算子梯度，因为每一段都是线性的。第三种表述 [(12.33)]，如第 [12.5] 节所示，能够使用凸二次规划（第 [7.3.2] 节）工具。

由于二元分类是机器学习中一个被深入研究的任务，有时也会使用其他术语，如discrimination、separation和decision。此外，二元分类器可以产生三种不同的输出。首先是线性函数本身的输出（通常称为score），可以取任何实数值。这个输出可以用于对样本进行排序，二元分类可以被视为在排序样本中选择一个阈值 [(Shawe-Taylor and Cristianini, 2004)]。二元分类器输出的第二种量通常被认为是经过非线性函数处理后确定的输出，将其值约束到有界范围内，例如区间 $[0, 1]$ 。常见的非线性函数是sigmoid函数 [(Bishop, 2006)]。当非线性处理产生校准良好的概率时 [(Gneiting and Raftery, 2007; Reid and Williamson, 2011)]，这称为类概率估计。二元分类器的第三种输出是最终的二元决策 $\{+1, -1\}$ ，这是最常被假定为分类器输出的形式。

SVM是一种二元分类器，本质上不容易进行概率解释。有几种方法可以将线性函数的原始输出（score）转换为校准的类概率估计 $(P(Y = 1|X = x))$ ，这些方法涉及额外的校准步骤 [(Platt, 2000; Zadrozny and Elkan, 2001; Lin et al., 2007)]。从训练的角度来看，有许多相关的概率方法。我们在第 [12.2.5] 节末尾提到了损失函数与似然函数之间的关系（也可比较第 [8.2] 和 [8.3] 节）。

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[394] [[Classification with Support Vector Machines]]

训练期间对应于校准良好转换的 maximum likelihood 方法称为 logistic regression，它来自于一类称为 generalized linear models 的方法。从这个角度来看 logistic regression 的细节可以在 [Agresti (2002, chapter 5)] 和 [McCullagh and Nelder (1989, chapter 4)] 中找到。

自然地，我们可以采用更加 Bayesian 的分类器输出观点，通过使用 Bayesian logistic regression 估计后验分布。Bayesian 观点还包括先验的规定，其中包括诸如与似然函数共轭（第 [6.6.1] 节）等设计选择。此外，还可以考虑将潜在函数作为先验，这会产生 Gaussian process classification [(Rasmussen and Williams, 2006, chapter 3)]。

[Draft (2023-10-18) of “Mathematics for Machine Learning” . Feedback: <https://mml-book.com>.]

References

- [Abel, Niels H. 1826.] [*Démonstration de l' Impossibilité de la Résolution Algébrique des Équations Générales qui Passent le Quatrième Degré.* Grøndahl and Son.]
- [Adhikari, Ani, and DeNero, John. 2018.] [*Computational and Inferential Thinking: The Foundations of Data Science.* Gitbooks.]
- [Agarwal, Arvind, and Daumé III, Hal. 2010. A Geometric View of Conjugate Priors. *Machine Learning*, **81**(1), 99 – 113.]
- [Agresti, A. 2002.] [*Categorical Data Analysis.* Wiley.]
- [Akaike, Hirotugu. 1974. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, **19**(6), 716 – 723.]
- [Akhiezer, Naum I., and Glazman, Izrail M. 1993.] [*Theory of Linear Operators in Hilbert Space.* Dover Publications.]
- [Alpaydin, Ethem. 2010.] [*Introduction to Machine Learning.* MIT Press.]
- [Amari, Shun-ichi. 2016.] [*Information Geometry and Its Applications.* Springer.]
- [Argyriou, Andreas, and Dinuzzo, Francesco. 2014. A Unifying View of Representer Theorems. In: *Proceedings of the International Conference on Machine Learning.*]
- [Aronszajn, Nachman. 1950. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, **68**, 337 – 404.]
- [Axler, Sheldon. 2015.] [*Linear Algebra Done Right.* Springer.]
- [Bakir, Gökhan, Hofmann, Thomas, Schölkopf, Bernhard, Smola, Alexander J., Taskar, Ben, and Vishwanathan, S. V. N. (eds). 2007.] [*Predicting Structured Data.* MIT Press.]
- [Barber, David. 2012.] [*Bayesian Reasoning and Machine Learning.* Cambridge University Press.]
- [Barndorff-Nielsen, Ole. 2014.] [*Information and Exponential Families: In Statistical Theory.* Wiley.]
- [Bartholomew, David, Knott, Martin, and Moustaki, Irini. 2011.] [*Latent Variable Models and Factor Analysis: A Unified Approach.* Wiley.]
- [Baydin, Atılım G., Pearlmutter, Barak A., Radul, Alexey A., and Siskind, Jeffrey M. 2018. Automatic Differentiation in Machine Learning: A Survey. *Journal of Machine Learning Research*, **18**, 1 – 43.]
- [Beck, Amir, and Teboulle, Marc. 2003. Mirror Descent and Nonlinear Projected Subgradient Methods for Convex Optimization. *Operations Research Letters*, **31**(3), 167 – 175.]
- [Belabbas, Mohamed-Ali, and Wolfe, Patrick J. 2009. Spectral Methods in Machine Learning and New Strategies for Very Large Datasets. *Proceedings of the National Academy of Sciences*, 0810600105.]

[Belkin, Mikhail, and Niyogi, Partha. 2003. Laplacian Eigenmaps for Dimensionality

Reduction and Data Representation. [Neural Computation][,] [15][6], 1373 – 1396.] [Ben-Hur, Asa, Ong, Cheng Soon, Sonnenburg, Sören] [Joren, Schölkopf, Bernhard, and Rätsch, Gunnar. 2008. Support Vector Machines and Kernels for Computational Biology.]

[PLoS Computational Biology][,] [4][10), e1000173.]

395

本材料由 Cambridge University Press 出版，书名为 *Mathematics for Machine Learning*，作者为 Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong (2020)。此版本仅供个人使用免费查看和下载。不得重新分发、转售或用于衍生作品。

©by M. P. Deisenroth, A. A. Faisal, and C. S. Ong, 2023. <https://mml-book.com>.

396 [参考文献]

Bennett, Kristin P., and Bredensteiner, Erin J. 2000a. SVM分类器中的对偶性和几何。在：机器学习国际会议论文集。

Bennett, Kristin P., and Bredensteiner, Erin J. 2000b. 学习中的几何。第132 – 145页：几何在工作中。美国数学协会。

Berlinet, Alain, and Thomas-Agnan, Christine. 2004. 概率与统计中的再生核Hilbert空间。Springer。

Bertsekas, Dimitri P. 1999. 非线性规划。Athena Scientific。

Bertsekas, Dimitri P. 2009. 凸优化理论。Athena Scientific。

Bickel, Peter J., and Doksum, Kjell. 2006. 数理统计：基本思想和精选主题。第1卷。Prentice Hall。

Bickson, Danny, Dolev, Danny, Shental, Ori, Siegel, Paul H., and Wolf, Jack K. 2007. 通过置信传播的线性检测。在：通信、控制和计算年度Allerton会议论文集。

Billingsley, Patrick. 1995. 概率与测度。Wiley。

Bishop, Christopher M. 1995. 模式识别神经网络。Clarendon Press。

Bishop, Christopher M. 1999. 贝叶斯PCA。在：神经信息处理系统进展。

Bishop, Christopher M. 2006. 模式识别与机器学习。Springer。

Blei, David M., Kucukelbir, Alp, and McAuliffe, Jon D. 2017. 变分推断：统计学家的综述。美国统计协会杂志，112(518), 859 – 877。

Blum, Arvind, and Hardt, Moritz. 2015. 阶梯：机器学习竞赛的可靠排行榜。在：国际机器学习会议。

Bonnans, J. Frédéric, Gilbert, J. Charles, Lemaréchal, Claude, and Sagastizábal, Claudia A. 2006. 数值优化：理论与实践方面。Springer。

- Borwein, Jonathan M., and Lewis, Adrian S. 2006. *凸分析与非线性优化*。第2版。加拿大数学学会。
- Bottou, Léon. 1998. 在线算法和随机逼近。第9–42页：*在线学习与神经网络*。Cambridge University Press。
- Bottou, Léon, Curtis, Frank E., and Nocedal, Jorge. 2018. 大规模机器学习的优化方法。*SIAM评论*, **60**(2), 223–311。
- Boucheron, Stephane, Lugosi, Gabor, and Massart, Pascal. 2013. *集中不等式：独立性的非渐近理论*。Oxford University Press。
- Boyd, Stephen, and Vandenberghe, Lieven. 2004. *凸优化*。Cambridge University Press。
- Boyd, Stephen, and Vandenberghe, Lieven. 2018. *应用线性代数导论*。Cambridge University Press。
- Brochu, Eric, Cora, Vlad M., and de Freitas, Nando. 2009. 昂贵成本函数的贝叶斯优化教程，及其在主动用户建模和分层强化学习中的应用。技术报告TR-2009-023。不列颠哥伦比亚大学计算机科学系。
- Brooks, Steve, Gelman, Andrew, Jones, Galin L., and Meng, Xiao-Li (eds). 2011. *Markov链Monte Carlo手册*。Chapman and Hall/CRC。
- Brown, Lawrence D. 1986. *统计指数族基础：在统计决策理论中的应用*。数理统计研究所。
- Bryson, Arthur E. 1961. 优化多阶段分配过程的梯度方法。在：*哈佛大学数字计算机及其应用研讨会论文集*。
- Bubeck, Sébastien. 2015. 凸优化：算法与复杂性。*机器学习基础与趋势*, **8**(3-4), 231–357。
- Bühlmann, Peter, and Van De Geer, Sara. 2011. *高维数据统计*。Springer。
- “Mathematics for Machine Learning” 草稿(2023-10-18)。反馈：<https://mml-book.com>。参考文献 [397]
- Burges, Christopher. 2010. 降维：导览。*机器学习基础与趋势*, **2**(4), 275–365。
- Carroll, J Douglas, and Chang, Jih-Jie. 1970. 通过“Eckart-Young”分解的N向泛化分析多维标度中的个体差异。*心理测量学*, **35**(3), 283–319。
- Casella, George, and Berger, Roger L. 2002. *统计推断*。Duxbury。
- Çinlar, Erhan. 2011. *概率与随机过程*。Springer。
- Chang, Chih-Chung, and Lin, Chih-Jen. 2011. LIBSVM：支持向量机库。*ACM智能系统与技术汇刊*, **2**, 27:1–27:27。
- Cheeseman, Peter. 1985. 为概率辩护。在：*国际人工智能联合会议论文集*。
- Chollet, Francois, and Allaire, J. J. 2018. *R语言深度学习*。Manning Publications。
- Codd, Edgar F. 1990. *数据库管理的关系模型*。Addison-Wesley Longman Publishing。
- [Cunningham, John P., and Ghahramani, Zoubin. 2015. Linear Dimensionality Reduc-
[tion: Survey, Insights, and Generalizations.] [*机器学习研究期刊*][,]

[16][, 2859 – 2900.]

[Datta, Biswa N. 2010.] [数值线性代数及其应用][. SIAM.] [Davidson, Anthony C., and Hinkley, David V. 1997.] [Bootstrap方法及其应用]

[Cambridge University Press.]

[Dean, Jeffrey, Corrado, Greg S., Monga, Rajat, and Chen, et al. 2012. Large Scale]

[Distributed Deep Networks. In:] [神经信息处理系统进展][.] [Deisenroth, Marc P., and Mohamed, Shakir. 2012. Expectation Propagation in Gaus-]

[sian Process Dynamical Systems. Pages 2618 – 2626 of:] [神经信息处理系统进展][.]

[Deisenroth, Marc P., and Ohlsson, Henrik. 2011. A General Perspective on Gaussian]

[Filtering and Smoothing: Explaining Current and Deriving New Algorithms. In:]

[美国控制会议论文集][.]

[Deisenroth, Marc P., Fox, Dieter, and Rasmussen, Carl E. 2015. Gaussian Processes]

[for Data-Efficient Learning in Robotics and Control.] [IEEE模式分析与机器智能汇刊][,] [37][[(2), 408 – 423.] [Dempster, Arthur P., Laird, Nan M., and Rubin, Donald B. 1977. Maximum Likelihood]

[from Incomplete Data via the EM Algorithm.] [皇家统计学会期刊][,]

[39][[(1), 1 – 38.]

[Deng, Li, Seltzer, Michael L., Yu, Dong, Acero, Alex, Mohamed, Abdel-rahman, and]

[Hinton, Geoffrey E. 2010. Binary Coding of Speech Spectrograms Using a Deep]

[Auto-Encoder. In:] [Interspeech会议论文集][.]

[Devroye, Luc. 1986.] [非均匀随机变量生成][. Springer.] [Donoho, David L., and Grimes, Carrie. 2003.] [[Hessian特征映射：高维数据的局部线性]]

[嵌入技术.] [美国国家科学院院刊][,] [100][[(10), 5591 – 5596.]

[Dostál, Zdeněk. 2009.] [最优二次规划算法：变分不等式应用][. Springer.]

[Douven, Igor. 2017. Abduction. In:] [斯坦福哲学百科全书][. Meta-]

[physics Research Lab, Stanford University.]

[Downey, Allen B. 2014.] [统计思维：探索性数据分析][. 2nd edn. O’ Reilly]

[Media.]

[Dreyfus, Stuart. 1962. The Numerical Solution of Variational Problems.] [数学分析与应用期刊][.] [5] [(1), 30 – 45.] [Drumm, Volker, and Weil, Wolfgang. 2001.] [线性代数与解析几何][.]

[Lecture Notes, Universität Karlsruhe (TH).]

[Dudley, Richard M. 2002.] [实分析与概率][. Cambridge University Press.]

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[[参考文献]]

[Eaton, Morris L. 2007.] [多元统计：向量空间方法][. Institute of]

[Mathematical Statistics Lecture Notes.]

[Eckart, Carl, and Young, Gale. 1936. The Approximation of One Matrix by Another of]

[Lower Rank.] [心理计量学][.] [1] [(3), 211 – 218.]

[Efron, Bradley, and Hastie, Trevor. 2016.] [计算机时代的统计推断：算法、证据与数据科学][. Cambridge University Press.]

[Efron, Bradley, and Tibshirani, Robert J. 1993.] [Bootstrap方法导论][. Chap-]

[man and Hall/CRC.]

[Elliott, Conal. 2009. Beautiful Differentiation. In:] [函数式编程国际会议][.]

[Evgeniou, Theodoros, Pontil, Massimiliano, and Poggio, Tomaso. 2000. Statistical]

[Learning Theory: A Primer.] [国际计算机视觉期刊][.] [38] [(1), 9 – 13.]

[Fan, Rong-En, Chang, Kai-Wei, Hsieh, Cho-Jui, Wang, Xiang-Rui, and Lin, Chih-Jen.]

[2008. LIBLINEAR: A Library for Large Linear Classification.] [机器学习研究期刊][.] [9], 1871 – 1874.]

[Gal, Yarin, van der Wilk, Mark, and Rasmussen, Carl E. 2014. Distributed Variational]

[Inference in Sparse Gaussian Process Regression and Latent Variable Models. In:] [神经信息处理系统进展][.]

[Gärtner, Thomas. 2008.] [结构化数据的核方法][. World Scientific.]

[Gavish, Matan, and Donoho, David L. 2014. The Optimal Hard Threshold for Singular] [✓]

[Values is] [4] [3][.] [IEEE信息论汇刊][.] [60] [(8), 5040 – 5053.]

[Gelman, Andrew, Carlin, John B., Stern, Hal S., and Rubin, Donald B. 2004.] [贝叶斯数据分析][. Chapman and Hall/CRC.]

[Gentle, James E. 2004.] [[随机数生成与Monte Carlo方法]][.]

[Springer.]

- [Ghahramani, Zoubin. 2015. Probabilistic Machine Learning and Artificial Intelligence.]
- [自然][,] [521][, 452 – 459.]
- [Ghahramani, Zoubin, and Roweis, Sam T. 1999. Learning Nonlinear Dynamical Sys-tems Using an EM Algorithm. In:] [神经信息处理系统进展][.] [MIT Press.]
- [Gilks, Walter R., Richardson, Sylvia, and Spiegelhalter, David J. 1996.] [Markov链Monte Carlo实践][. Chapman and Hall/CRC.]
- [Gneiting, Tilmann, and Raftery, Adrian E. 2007. Strictly Proper Scoring Rules, Pre-diction, and Estimation.] [美国统计协会期刊][,] [102][[(477),] [359 – 378.]
- [Goh, Gabriel. 2017. Why Momentum Really Works.] [Distill][.]
- [Gohberg, Israel, Goldberg, Seymour, and Krupnik, Nahum. 2012.] [线性算子的迹与行列式][. Birkhäuser.]
- [Golan, Jonathan S. 2007.] [研究生应掌握的线性代数][. Springer.]
- [Golub, Gene H., and Van Loan, Charles F. 2012.] [矩阵计算][. JHU Press.]
- [Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. 2016.] [深度学习][. MIT Press.]
- [Graepel, Thore, Candela, Joaquin Quiñonero-Candela, Borchert, Thomas, and Her-Brich, Ralf. 2010. Web-Scale Bayesian Click-through Rate Prediction for Sponsored Search Advertising in Microsoft’ s Bing Search Engine. In: *Proceedings of the International Conference on Machine Learning*.
- Griewank, Andreas, and Walther, Andrea. 2003. Introduction to Automatic Differentiation. In: *Proceedings in Applied Mathematics and Mechanics*.
- Griewank, Andreas, and Walther, Andrea. 2008. *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*. SIAM.
- Grimmett, Geoffrey R., and Welsh, Dominic. 2014. *Probability: An Introduction*. Oxford University Press.
- Draft (2023-10-18) of “Mathematics for Machine Learning” . Feedback: <https://mml-book.com>. *References* [399]
- Grinstead, Charles M., and Snell, J. Laurie. 1997. *Introduction to Probability*. American Mathematical Society.
- Hacking, Ian. 2001. *Probability and Inductive Logic*. Cambridge University Press.
- Hall, Peter. 1992. *The Bootstrap and Edgeworth Expansion*. Springer.
- Hallin, Marc, Paindaveine, Davy, and Šiman, Miroslav. 2010. Multivariate Quantiles and Multiple-Output Regression Quantiles: From ℓ_1 Optimization to Halfspace Depth. *Annals of Statistics*, **38**, 635 – 669.

- Hasselblatt, Boris, and Katok, Anatole. 2003. *A First Course in Dynamics with a Panorama of Recent Developments*. Cambridge University Press.
- Hastie, Trevor, Tibshirani, Robert, and Friedman, Jerome. 2001. *The Elements of Statistical Learning – Data Mining, Inference, and Prediction*. Springer.
- Hausman, Karol, Springenberg, Jost T., Wang, Ziyu, Heess, Nicolas, and Riedmiller, Martin. 2018. Learning an Embedding Space for Transferable Robot Skills. In: *Proceedings of the International Conference on Learning Representations*.
- Hazan, Elad. 2015. Introduction to Online Convex Optimization. *Foundations and Trends in Optimization*, **2**(3 – 4), 157 – 325.
- Hensman, James, Fusi, Nicolò, and Lawrence, Neil D. 2013. Gaussian Processes for Big Data. In: *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
- Herbrich, Ralf, Minka, Tom, and Graepel, Thore. 2007. TrueSkill(TM): A Bayesian Skill Rating System. In: *Advances in Neural Information Processing Systems*.
- Hiriart-Urruty, Jean-Baptiste, and Lemaréchal, Claude. 2001. *Fundamentals of Convex Analysis*. Springer.
- Hoffman, Matthew D., Blei, David M., and Bach, Francis. 2010. Online Learning for Latent Dirichlet Allocation. *Advances in Neural Information Processing Systems*.
- Hoffman, Matthew D., Blei, David M., Wang, Chong, and Paisley, John. 2013. Stochastic Variational Inference. *Journal of Machine Learning Research*, **14**(1), 1303 – 1347.
- Hofmann, Thomas, Schölkopf, Bernhard, and Smola, Alexander J. 2008. Kernel Methods in Machine Learning. *Annals of Statistics*, **36**(3), 1171 – 1220.
- Hogben, Leslie. 2013. *Handbook of Linear Algebra*. Chapman and Hall/CRC.
- Horn, Roger A., and Johnson, Charles R. 2013. *Matrix Analysis*. Cambridge University Press.
- Hotelling, Harold. 1933. Analysis of a Complex of Statistical Variables into Principal Components. *Journal of Educational Psychology*, **24**, 417 – 441.
- Hyvarinen, Aapo, Oja, Erkki, and Karhunen, Juha. 2001. *Independent Component Analysis*. Wiley.
- Imbens, Guido W., and Rubin, Donald B. 2015. *Causal Inference for Statistics, Social and Biomedical Sciences*. Cambridge University Press.
- Jacod, Jean, and Protter, Philip. 2004. *Probability Essentials*. Springer.
- Jaynes, Edwin T. 2003. *Probability Theory: The Logic of Science*. Cambridge University Press.
- Jefferys, William H., and Berger, James O. 1992. Ockham’s Razor and Bayesian Analysis. *American Scientist*, **80**, 64 – 72.
- Jeffreys, Harold. 1961. *Theory of Probability*. Oxford University Press.

Jimenez Rezende, Danilo, and Mohamed, Shakir. 2015. Variational Inference with Normalizing Flows. In: *Proceedings of the International Conference on Machine Learning*.

Jimenez Rezende, Danilo, Mohamed, Shakir, and Wierstra, Daan. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In: *Proceedings of the International Conference on Machine Learning*.

Joachims, Thorsten. 1999. *Advances in Kernel Methods – Support Vector Learning*. MIT Press. Chap. Making Large-Scale SVM Learning Practical, pages 169 – 184.

Jordan, Michael I., Ghahramani, Zoubin, Jaakkola, Tommi S., and Saul, Lawrence K. 1999. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, **37**, 183 – 233.

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

References [400]

Julier, Simon J., and Uhlmann, Jeffrey K. 1997. A New Extension of the Kalman Filter to Nonlinear Systems. In: *Proceedings of AeroSense Symposium on Aerospace/Defense Sensing, Simulation and Controls*.

Kaiser, Marcus, and Hilgetag, Claus C. 2006. Nonoptimal Component Placement, but Short Processing Paths, Due to Long-Distance Projections in Neural Systems. *PLoS Computational Biology*, **2**(7), e95.

Kalman, Dan. 1996. A Singularly Valuable Decomposition: The SVD of a Matrix. *College Mathematics Journal*, **27**(1), 2 – 23.

Kalman, Rudolf E. 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME – Journal of Basic Engineering*, **82**(Series D), 35 – 45.

Kamthe, Sanket, and Deisenroth, Marc P. 2018. Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

[Katz, Victor J. 2004.] [数学史][. Pearson/Addison-Wesley.]

[Kelley, Henry J. 1960. 最优飞行路径的梯度理论.] [Ars Journal][,] [30] [(10),]

[947 – 954.]

[Kimeldorf, George S., and Wahba, Grace. 1970. Bayesian估计在随机过程中的对应关系]

[和样条平滑.] [数学统计年鉴][,] [41] [(2), 495 – 502.]

[Kingma, Diederik P., and Welling, Max. 2014. Auto-Encoding Variational Bayes. 载于:]

[学习表示国际会议论文集][.]

[Kittler, Josef, and Föglein, Janos. 1984. 多光谱像素的上下文分类]

[数据.] [图像与视觉计算][,] [2] [(1), 13 – 29.]

[Kolda, Tamara G., and Bader, Brett W. 2009. 张量分解及其应用.]

[SIAM评论] [,] [51][(3), 455 – 500.]

[Koller, Daphne, and Friedman, Nir. 2009.] [概率图模型][. MIT Press.]

[Kong, Linglong, and Mizera, Ivan. 2012. 分位数层析成像：在多变量数据中使用分位数
[数据.] [中国统计学][,] [22][, 1598 – 1610.]

[Lang, Serge. 1987.] [线性代数][. Springer.]

[Lawrence, Neil D. 2005. 使用Gaussian过程潜变量模型的概率非线性主成分分析
[Gaussian过程潜变量模型.] [机器学习研究期刊][,] [6][(Nov.), 1783 – 1816.]

[Leemis, Lawrence M., and McQueston, Jacquelyn T. 2008. 单变量分布
[关系.] [美国统计学家][,] [62][(1), 45 – 53.]

[Lehmann, Erich L., and Romano, Joseph P. 2005.] [统计假设检验][.]
[Springer.]

[Lehmann, Erich Leo, and Casella, George. 1998.] [点估计理论][. Springer.]

[Liesen, Jörg, and Mehrmann, Volker. 2015.] [线性代数][. Springer.]

[Lin, Hsuan-Tien, Lin, Chih-Jen, and Weng, Ruby C. 2007. 关于Platt的概率性的注记
[支持向量机的输出.] [机器学习][,] [68][, 267 – 276.]

[Ljung, Lennart. 1999.] [系统辨识：用户理论][. Prentice Hall.]

[Loosli, Gaëlle, Canu, Stéphane, and Ong, Cheng Soon. 2016. 在Kreĭn空间中学习SVM
[空间.] [IEEE模式分析与机器智能汇刊][,] [38][(6), 1204 –] [1216.]

[Luenberger, David G. 1969.] [向量空间方法优化][. Wiley.]

[MacKay, David J. C. 1992. Bayesian插值.] [神经计算][,] [4][, 415 – 447.]

[MacKay, David J. C. 1998. Gaussian过程介绍. 第133 – 165页载于:]

[Bishop, C. M. (编),] [神经网络与机器学习][. Springer.]

[MacKay, David J. C. 2003.] [信息论、推理和学习算法][.]
[Cambridge University Press.]

[Magnus, Jan R., and Neudecker, Heinz. 2007.] [矩阵微分学及其应用]

[统计学和计量经济学中的应用][. Wiley.]

[《机器学习数学》草稿 (2023-10-18). 反馈:] [<https://mml-book.com>][.] [参考文献] [[401]]

[Manton, Jonathan H., and Amblard, Pierre-Olivier. 2015. 再生入门]

[核Hilbert空间.] [信号处理基础与趋势][.] [8] [(1 - 2), 1 - 126.]

[Markovsky, Ivan. 2011.] [低秩近似: 算法、实现、应用]

[应用][. Springer.]

[Maybeck, Peter S. 1979.] [随机模型、估计与控制][. Academic Press.]

[McCullagh, Peter, and Nelder, John A. 1989.] [广义线性模型][. CRC Press.]

[McEliece, Robert J., MacKay, David J. C., and Cheng, Jung-Fu. 1998. Turbo解码]

[作为Pearl的”置信传播”算法的实例.] [IEEE通信选定领域期刊]

[通信领域][.] [16] [(2), 140 - 152.]

[Mika, Sebastian, Rätsch, Gunnar, Weston, Jason, Schölkopf, Bernhard, and Müller,]

[Klaus-Robert. 1999. 使用核的Fisher判别分析. 第41 - 48页载于:]

[神经网络信号处理研讨会论文集][.]

[Minka, Thomas P. 2001a.] [近似Bayesian推理算法族][.]

[博士论文, 麻省理工学院.]

[Minka, Tom. 2001b. PCA维度的自动选择. 载于:] [神经信息处理系统进展]

[神经信息处理系统][.]

[Mitchell, Tom. 1997.] [机器学习][. McGraw-Hill.]

[Mnih, Volodymyr, Kavukcuoglu, Koray, and Silver, David, et al. 2015. 人类水平]

[通过深度强化学习进行控制.] [自然][.] [518][, 529 - 533.]

[Moonen, Marc, and De Moor, Bart. 1995.] [SVD与信号处理, III: 算法,]

[架构与应用][. Elsevier.]

[Moustaki, Irini, Knott, Martin, and Bartholomew, David J. 2015.] [潜变量建模]

[建模][. American Cancer Society. 第1 - 10页.]

[Müller, Andreas C., and Guido, Sarah. 2016.] [*Python机器学习入门*]

[*Python: 数据科学家指南*][. O’ Reilly Publishing.]

[Murphy, Kevin P. 2012.] [*机器学习：概率视角*][. MIT Press.]

[Neal, Radford M. 1996.] [*神经网络的Bayesian学习*][. 博士论文, 多伦多大学计算机科学系.]

[Neal, Radford M., and Hinton, Geoffrey E. 1999. EM算法的一种观点]

[证明增量、稀疏和其他变体的合理性. 第355 – 368页载于:] [*图模型中的学习*

[*图形模型*][. MIT Press.]

[Nelsen, Roger. 2006.] [*Copulas介绍*][. Springer.]

[Nesterov, Yuri. 2018.] [*凸优化讲义*][. Springer.]

[Neumaier, Arnold. 1998. 求解病态和奇异线性系统：正则化教程]

[正则化教程.] [*SIAM评论*][.] [40][, 636 – 666.]

[Nocedal, Jorge, and Wright, Stephen J. 2006.] [*数值优化*][. Springer.]

[Nowozin, Sebastian, Gehler, Peter V., Jancsary, Jeremy, and Lampert, Christoph H.]

[(eds). 2014.] [*Advanced Structured Prediction*][. MIT Press.] [O’ Hagan, Anthony. 1991. Bayes-Hermite Quadrature.] [*Journal of Statistical Planning*

[and Inference][.] [29][, 245 – 260.]

[Ong, Cheng Soon, Mary, Xavier, Canu, Stéphane, and Smola, Alexander J. 2004. 使用非正定核学习. 见:] [*国际机器学习会议论文集*][.]

[Ormonieit, Dirk, Sidenblad, Hedvig, Black, Michael J., and Hastie, Trevor. 2001.]

[学习和跟踪循环人体运动. 见:] [*神经信息处理系统进展*][.]

[Page, Lawrence, Brin, Sergey, Motwani, Rajeev, and Winograd, Terry. 1999.] [*PageRank引用排名：为Web带来秩序*][. 技术报告. Stanford Info-]

[Lab.]

[Paquet, Ulrich. 2008.] [*潜变量模型的贝叶斯推断*][. 博士论文, 剑桥大学.]

[Parzen, Emanuel. 1962. 关于概率密度函数和模态的估计.]

[数理统计年鉴][.] [33][(3), 1065 – 1076.]

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).]

[402] [[参考文献]]

[Pearl, Judea. 1988.] [智能系统中的概率推理：似然推断网络][. Morgan Kaufmann.]

[Pearl, Judea. 2009.] [因果关系：模型、推理和推断][. 第2版. Cambridge]

[University Press.]

[Pearson, Karl. 1895. 对进化数学理论的贡献. II. 均匀材料中的偏态变异.][皇家学会哲学学报] [A：数学、物理和工程科学][.][186][, 343 – 414.]

[Pearson, Karl. 1901. 关于空间点系统最佳拟合的直线和平面.]

[哲学杂志][.][2][[(11), 559 – 572.]

[Peters, Jonas, Janzing, Dominik, and Schölkopf, Bernhard. 2017.] [因果推断要素：基础和学习算法][. MIT Press.]

[Petersen, Kaare B., and Pedersen, Michael S. 2012.] [矩阵手册][. 技术报告]

[丹麦技术大学.]

[Platt, John C. 2000. 支持向量机的概率输出及与正则化似然方法的比较. 见:] [大边距分类器进展][.]

[Pollard, David. 2002.] [测度论概率用户指南][. Cambridge]

[University Press.]

[Polyak, Roman A. 2016. 现代优化中的Legendre变换. 第]

[437 – 507页: Goldengorin, B. (ed).][控制和数据科学中的优化及其应用][. Springer.]

[Press, William H., Teukolsky, Saul A., Vetterling, William T., and Flannery, Brian P.]

[2007.] [数值算法：科学计算的艺术][. Cambridge University] [Press.]

[Proschan, Michael A., and Presnell, Brett. 1998. 从条件期望中预料意外.]

[美国统计学家][.][52][[(3), 248 – 252.]

[Raschka, Sebastian, and Mirjalili, Vahid. 2017.] [Python Machine Learning：使用Python、scikit-learn和TensorFlow进行机器学习和深度学习][. Packt出版社.]

[Rasmussen, Carl E., and Ghahramani, Zoubin. 2001. Occam剃刀原理. 见:] [神经信息处理系统进展][.]

[Rasmussen, Carl E., and Ghahramani, Zoubin. 2003. 贝叶斯Monte Carlo. 见:] [神经信息处理系统进展][.]

[Rasmussen, Carl E., and Williams, Christopher K. I. 2006.] [机器学习的Gaussian过程][. MIT Press.]

[Reid, Mark, and Williamson, Robert C. 2011. 二元实验的信息、散度和风险.]

- [机器学习研究期刊][,] [12][, 731 – 817.]
- [Rifkin, Ryan M., and Lippert, Ross A. 2007. 值正则化和Fenchel对偶性.]
- [机器学习研究期刊][,] [8][, 441 – 479.]
- [Rockafellar, Ralph T. 1970.] [凸分析][. Princeton University Press.]
- [Rogers, Simon, and Girolami, Mark. 2016.] [机器学习第一课程][. Chapman and Hall/CRC.]
- [Rosenbaum, Paul R. 2017.] [观察与实验：因果推断导论][. Harvard University Press.]
- [Rosenblatt, Murray. 1956. 关于密度函数某些非参数估计的评论.]
- [数理统计年鉴][,] [27][(3), 832 – 837.]
- [Roweis, Sam T. 1998. PCA和SPCA的EM算法. 第626 – 632页:] [神经信息处理系统进展][.]
- [Roweis, Sam T., and Ghahramani, Zoubin. 1999. 线性Gaussian模型的统一综述.]
- [神经计算][,] [11][(2), 305 – 345.]
- [Roy, Anindya, and Banerjee, Sudipto. 2014.] [统计学的线性代数和矩阵分析][. Chapman and Hall/CRC.]
- [Rubinstein, Reuven Y., and Kroese, Dirk P. 2016.] [仿真和Monte Carlo方法][. Wiley.]
- [《机器学习数学》草稿 (2023-10-18). 反馈:] [<https://mml-book.com>][,] [参考文献] [[403]]
- [Ruffini, Paolo. 1799.] [方程的一般理论，其中证明了四次以上一般方程代数解的不可能性][. S. Tommaso d’ Aquino印刷厂.]
- [Rumelhart, David E., Hinton, Geoffrey E., and Williams, Ronald J. 1986. 通过反向传播误差学习表示.] [自然][,] [323][(6088), 533 – 536.]
- [Sæmundsson, Steindór, Hofmann, Katja, and Deisenroth, Marc P. 2018. 使用潜变量Gaussian过程的元强化学习. 见:] [人工智能不确定性会议论文集][.]
- [Saitoh, Saburou. 1988.] [再生核理论及其应用][. Longman Scientific and Technical.]
- [Sims, John A., and Simon, David. 2013.] [贝叶斯滤波和平滑][. Cambridge University Press.]
- [Schölkopf, Bernhard, and Smola, Alexander J. 2002.] [核学习——支持
Vector Machines, Regularization, Optimization, and Beyond. MIT Press.
- Schölkopf, Bernhard, Smola, Alexander J., and Müller, Klaus-Robert. 1997. Kernel Principal Component Analysis. In: *Proceedings of the International Conference on Artificial Neural Networks*.

- Schölkopf, Bernhard, Smola, Alexander J., and Müller, Klaus-Robert. 1998. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, **10**(5), 1299 – 1319.
- Schölkopf, Bernhard, Herbrich, Ralf, and Smola, Alexander J. 2001. A Generalized Representer Theorem. In: *Proceedings of the International Conference on Computational Learning Theory*.
- Schwartz, Laurent. 1964. Sous Espaces Hilbertiens d' Espaces Vectoriels Topologiques et Noyaux Associés. *Journal d' Analyse Mathématique*, **13**, 115 – 256.
- Schwarz, Gideon E. 1978. Estimating the Dimension of a Model. *Annals of Statistics*, **6**(2), 461 – 464.
- Shahriari, Bobak, Swersky, Kevin, Wang, Ziyu, Adams, Ryan P., and De Freitas, Nando. 2016. Taking the Human out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, **104**(1), 148 – 175.
- Shalev-Shwartz, Shai, and Ben-David, Shai. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.
- Shawe-Taylor, John, and Cristianini, Nello. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.
- Shawe-Taylor, John, and Sun, Shiliang. 2011. A Review of Optimization Methodologies in Support Vector Machines. *Neurocomputing*, **74**(17), 3609 – 3618.
- Shental, Ori, Siegel, Paul H., Wolf, Jack K., Bickson, Danny, and Dolev, Danny. 2008. Gaussian Belief Propagation Solver for Systems of Linear Equations. Pages 1863 – 1867 of: *Proceedings of the International Symposium on Information Theory*.
- Shewchuk, Jonathan R. 1994. *An Introduction to the Conjugate Gradient Method without the Agonizing Pain*.
- Shi, Jianbo, and Malik, Jitendra. 2000. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8), 888 – 905.
- Shi, Qinfeng, Petterson, James, Dror, Gideon, Langford, John, Smola, Alexander J., and Vishwanathan, S. V. N. 2009. Hash Kernels for Structured Data. *Journal of Machine Learning Research*, 2615 – 2637.
- Shiryayev, Albert N. 1984. *Probability*. Springer.
- Shor, Naum Z. 1985. *Minimization Methods for Non-Differentiable Functions*. Springer.
- Shotton, Jamie, Winn, John, Rother, Carsten, and Criminisi, Antonio. 2006. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. In: *Proceedings of the European Conference on Computer Vision*.
- Smith, Adrian F. M., and Spiegelhalter, David. 1980. Bayes Factors and Choice Criteria for Linear Models. *Journal of the Royal Statistical Society B*, **42**(2), 213 – 220.
- ©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. Published by Cambridge University Press (2020).

参考文献

Snoek, Jasper, Larochelle, Hugo, and Adams, Ryan P. 2012. Practical Bayesian Optimization of Machine Learning Algorithms. In: *Advances in Neural Information Processing Systems*.

Spearman, Charles. 1904. "General Intelligence," Objectively Determined and Measured. *American Journal of Psychology*, **15**(2), 201 – 292.

Sriperumbudur, Bharath K., Gretton, Arthur, Fukumizu, Kenji, Schölkopf, Bernhard, and Lanckriet, Gert R. G. 2010. Hilbert Space Embeddings and Metrics on Probability Measures. *Journal of Machine Learning Research*, **11**, 1517 – 1561.

Steinwart, Ingo. 2007. How to Compare Different Loss Functions and Their Risks. *Constructive Approximation*, **26**, 225 – 287.

Steinwart, Ingo, and Christmann, Andreas. 2008. *Support Vector Machines*. Springer.

Stoer, Josef, and Burlirsch, Roland. 2002. *Introduction to Numerical Analysis*. Springer.

Strang, Gilbert. 1993. The Fundamental Theorem of Linear Algebra. *The American Mathematical Monthly*, **100**(9), 848 – 855.

Strang, Gilbert. 2003. *Introduction to Linear Algebra*. Wellesley-Cambridge Press.

Stray, Jonathan. 2016. *The Curious Journalist's Guide to Data*. Tow Center for Digital Journalism at Columbia's Graduate School of Journalism.

Strogatz, Steven. 2014. Writing about Math for the Perplexed and the Traumatized. *Notices of the American Mathematical Society*, **61**(3), 286 – 291.

Sucar, Luis E., and Gillies, Duncan F. 1994. Probabilistic Reasoning in High-Level Vision. *Image and Vision Computing*, **12**(1), 42 – 60.

Szeliski, Richard, Zabih, Ramin, and Scharstein, Daniel, et al. 2008. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(6), 1068 – 1080.

Tandra, Haryono. 2014. The Relationship between the Change of Variable Theorem and the Fundamental Theorem of Calculus for the Lebesgue Integral. *Teaching of Mathematics*, **17**(2), 76 – 83.

Tenenbaum, Joshua B., De Silva, Vin, and Langford, John C. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, **290**(5500), 2319 – 2323.

Tibshirani, Robert. 1996. Regression Selection and Shrinkage via the Lasso. *Journal of the Royal Statistical Society B*, **58**(1), 267 – 288.

Tipping, Michael E., and Bishop, Christopher M. 1999. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society: Series B*, **61**(3), 611 – 622.

Titsias, Michalis K., and Lawrence, Neil D. 2010. Bayesian Gaussian Process Latent

[可变模型。见：][*人工智能与统计国际会议论文集*]

- [Toussaint, Marc. 2012.] [梯度下降的一些笔记]. <https://ipvs.informatik.uni-stuttgart.de/mlr/marc/notes/gradientDescent.pdf>.
- [Trefethen, Lloyd N., and Bau III, David. 1997.] [数值线性代数]. SIAM.
- [Tucker, Ledyard R. 1966. 三模因子分析的一些数学笔记.] [心理测量学], [31] [(3), 279 – 311.]
- [Vapnik, Vladimir N. 1998.] [统计学习理论]. Wiley.
- [Vapnik, Vladimir N. 1999. 统计学习理论概述.] [IEEE神经网络学报], [10] [(5), 988 – 999.]
- [Vapnik, Vladimir N. 2000.] [统计学习理论的本质]. Springer.
- [Vishwanathan, S. V. N., Schraudolph, Nicol N., Kondor, Risi, and Borgwardt, Karsten M. 2010. 图核(Graph Kernels).] [机器学习研究杂志], [11], 1201 – 1242.
- [von Luxburg, Ulrike, and Schölkopf, Bernhard. 2011. 统计学习理论：模型、概念和结果. 651 – 706页: D. M. Gabbay, S. Hartmann,] [J. Woods (编),] [逻辑史手册], 第10卷. Elsevier.
- [《机器学习数学》草稿 (2023-10-18). 反馈:] [<https://mml-book.com/>][.] [参考文献] [[405]]
- [Wahba, Grace. 1990.] [观测数据的样条模型]. 工业与应用数学学会.
- [Walpole, Ronald E., Myers, Raymond H., Myers, Sharon L., and Ye, Keying. 2011.] [工程师和科学家的概率与统计]. Prentice Hall.
- [Wasserman, Larry. 2004.] [统计学全览]. Springer.
- [Wasserman, Larry. 2007.] [非参数统计学全览]. Springer.
- [Whittle, Peter. 2000.] [通过期望理解概率]. Springer.
- [Wickham, Hadley. 2014. 整洁数据.] [统计软件杂志], [59], 1 – 23.
- [Williams, Christopher K. I. 1997. 无限网络计算. 见:] [神经信息处理系统进展].
- [Yu, Yaoliang, Cheng, Hao, Schuurmans, Dale, and Szepesvári, Csaba. 2013. 表示定理的特征化. 见:] [国际机器学习会议论文集].
- [Zadrozny, Bianca, and Elkan, Charles. 2001. 从决策树和朴素贝叶斯分类器获得校准概率估计. 见:] [国际机器学习会议论文集].
- [Zhang, Haizhang, Xu, Yuesheng, and Zhang, Jun. 2009. 机器学习的再生核Banach空间.] [机器学习研究杂志], [10], 2741 – 2775.
- [Zia, Royce K. P., Redish, Edward F., and McKay, Susan R. 2009. 理解Legendre变换.] [美国物理学杂志], [77] [(614), 614 – 622.]
- [©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 剑桥大学出版社出版 (2020).]

索引

[1][-of-][K] [表示, [364]] [标准基, [45]]

ℓ [范数, [71]] [标准特征映射, [389]]

$\ell[2]$ [范数, [72]] [标准链接函数, [315]]

[溯因推理, [258]] [类别变量, [180]]

[Abel-Ruffini定理, [334]] [Cauchy-Schwarz不等式, [75]]

[阿贝尔群, [36]] [变量变换技术, [219]]

[绝对齐次, [71]] [特征多项式, [104]]

[激活函数, [315]] [Cholesky分解, [114]]

[仿射映射, [63]] [Cholesky因子, [114]]

[仿射子空间, [61]] [Cholesky分解, [114]]

[赤池信息准则, [288]] [类, [370]]

[代数, [17]] [分类, [315]]

[代数重数, [106]] [闭包, [36]]

[解析的, [143]] [代码, [343]]

[祖先采样, [340,] [364]] [同向, [105]]

[角度, [76]] [余域, [58,] [139]]

[结合律, [24,] [26,] [36]] [共线, [105]]

[属性, [253]] [列, [22]]

[增广矩阵, [29]] [列空间, [59]]

[自编码器, [343]] [列向量, [22,] [38]]

[自动微分, [161]] [配方, [307]]

[自同构, [49]] [凹函数, [236]]

[条件数, [230]]

[反向传播, [159]]

[条件概率, [179]]

[贝叶斯规则, [185]] [凸函数, [236]]

[凸包, [386]]

[贝叶斯定理, [185]] [凸优化问题, [236,] [239]]

[贝叶斯GP-LVM, [347]] [凸集, [236]]

[贝叶斯推理, [274]] [坐标, [50]]

[贝叶斯信息准则, [288]] [坐标表示, [50]]

[贝叶斯线性回归, [303]] [坐标向量, [50]]

[贝叶斯模型选择, [286]] [相关性, [191]]

[贝叶斯网络, [278,] [283]] [协方差, [190]]

[贝叶斯PCA, [346]] [共轭, [208]]

[基向量, [45]] [共轭先验, [208]]

[贝叶斯因子, [287]] [凸共轭, [242]]

[贝叶斯定律, [185]] [基, [44]]

[条件独立, [195]]

[基本变量, [30]]

[伯努利分布, [205]] [协方差矩阵, [190,] [198]]

[协变量, [253]]

[Beta分布, [206]] [CP分解, [136]]

[双线性映射, [72]] [交叉协方差, [191]]

[双射, [48]] [交叉验证, [258,] [263]]

[二元分类, [370]] [累积分布函数, [178,] [181]]

[二项分布, [206]]

[盲源分离, [346]]

[Borel σ -代数, [180]]

[d-分离, [281]]

[407]

[本材料由剑桥大学出版社出版为] [机器学习数学] [作者] [Marc Peter Deisenroth, A. Aldo Faisal, 和 Cheng Soon Ong (2020). 本版本仅供个人使用免费查看和下载。不得重新分发、转售或用于衍生作品.]

[©M. P. Deisenroth, A. A. Faisal, 和 C. S. Ong, 2023.] [<https://mml-book.com>][.]

[408] [索引]

[数据协方差矩阵, [318]] [特征, [253]]

[数据点, [253]] [特征映射, [254]]

[数据拟合项, [302]] [特征矩阵, [296]]

[解码器, [343]] [特征向量, [295]]

[深度自编码器, [347]] [Fisher判别分析, [136]]

[缺陷的, [111]] [Fisher-Neyman定理, [210]]

[分母布局, [151]] [前向模式, [161]]

[导数, [141]] [自由变量, [30]]

[设计矩阵, [294,] [296]] [满秩, [47]]

[行列式, [99]] [完整SVD, [128]]

[对角矩阵, [115]] [线性映射基本定理, [60]]

[可对角化的, [116]] [高斯消元法, [31]]

[对角化, [116]] [高斯混合模型, [349]]

[差商, [141]] [高斯过程, [316]]

[维数, [45]] [高斯过程潜变量模型, [347]]

[有向图模型, [278,] [283]] [一般线性群, [37]]

[方向, [61]] [通解, [28,] [30]]

[方向空间, [61]] [广义线性模型, [272,] [315]]

[距离, [75]] [生成集, [44]]

[分布, [177]] [生成过程, [272,] [286]]

[分配律, [24,] [26]] [生成器, [344]]

[定义域, [58,] [139]] [几何重数, [108]]

[点积, [72]] [Givens旋转, [94]]

[对偶SVM, [385]] [全局最小值, [225]]

[Eckart-Young定理, [131,] [334]] [GP-LVM, [347]]

[特征分解, [116]] [梯度, [146]]

[特征空间, [106]] [Gram矩阵, [389]]

[特征谱, [106]] [Gram-Schmidt正交化, [89]]

[特征值, [105]] [图模型, [278]]

[特征值方程, [105]] [群, [36]]

[特征向量, [105]] [Hadamard乘积, [23]]

[初等变换, [28]] [硬间隔SVM, [377]]

[EM算法, [360]] [Hessian矩阵, [164]]

[尴尬并行(embarrassingly parallel), [264]] [Hessian特征映射, [136]]

[经验协方差, [192]] [Hessian矩阵, [165]]

[经验均值, [192]] [铰链损失, [381]]

[经验风险, [260]] [直方图, [369]]

[经验风险最小化, [257,] [260]] [超参数, [258]]

[编码器, [343]] [超平面, [61,] [62]]

[自同态, [49]] [超先验, [281]]

[上图(epigraph), [236]] [i.i.d., [195]]

[等价的, [56]] [ICA, [346]]

[误差函数, [294]] [恒等自同态, [49]]

[误差项, [382]] [恒等映射, [49]]

[欧几里得距离, [72,] [75]] [单位矩阵, [23]]

[欧几里得范数, [72]] [像, [58,] [139]]

[欧几里得向量空间, [73]] [独立同分布, [195,] [260,] [266]]

[事件空间, [175]] [独立成分分析, [346]]

[证据, [186,] [285,] [306]] [推理网络, [344]]

[样本, [253]] [单射的, [48]]

[期望风险, [261]] [内积, [73]]

[期望值, [187]] [内积空间, [73]]

[指数族, [205,] [211]] [中间变量, [162]]

[扩展卡尔曼滤波器, [170]] [逆, [24]]

[因子分析, [346]] [逆元, [36]]

[因子图, [283]] [可逆的, [24]]

[草稿 (2023-10-18) 《机器学习数学》。反馈:] [<https://mml-book.com>] 索引 [409]

[Isomap, [136]] [间隔, [374]]

[同构, [49]] [边际, [190]]

[Jacobian矩阵, [146,] [150]] [边际似然, [186,] [286,] [306]]

[Jacobian行列式, [152]] [边际概率, [179]]

[Jeffreys-Lindley悖论, [287]] [边际化性质, [184]]

[Jensen不等式, [239]] [马尔可夫随机场, [283]]

[联合概率, [178]] [矩阵, [22]]

[矩阵分解, [98]]

[Karhunen-Loève变换, [318]] [最大后验, [300]]

[核, [33,] [47,] [58,] [254,] [388]] [最大后验估计, [269]]

[核密度估计, [369]] [最大似然, [257]]

[核矩阵, [389]] [最大似然估计, [296]]

[核PCA, [347]] [最大似然估计, [265,] [293]]

[核技巧, [316,] [347,] [389]] [均值, [187]]

[标签, [253]] [均值函数, [309]]

[拉格朗日乘数, [234]] [均值向量, [198]]

[拉格朗日函数, [234]] [测度, [180]]

[拉格朗日对偶问题, [234]] [中位数, [188]]

[拉普拉斯近似, [170]] [度量, [76]]

[拉普拉斯展开, [102]] [最小的, [44]]

[拉普拉斯特征映射, [136]] [极小极大不等式, [234]]

[LASSO, [303,] [316]] [失拟项, [302]]

[潜变量, [275]] [混合模型, [349]]

[定律, [177,] [181]] [混合权重, [349]]

[全方差定律, [203]] [众数, [188]]

[首项系数, [30]] [模型, [251]]

[最小二乘问题, [261]] [模型证据, [286]]

[最小二乘解, [88]] [Moore-Penrose伪逆, [35]]

[最小二乘损失, [154]] [多维标度, [136]]

[左奇异向量, [119]] [模型选择, [258]]

[勒让德变换, [242]] [标量乘法, [37]]

[勒让德-芬切尔变换, [242]] [多元的, [178]]

[长度, [71]] [多元高斯分布, [198]]

[似然, [185,] [265,] [269,] [291]] [多元泰勒级数, [166]]

[直线, [61,] [82]] [自然参数, [212]]

[线性组合, [40]] [负对数似然, [265]]

[线性流形, [61]] [嵌套交叉验证, [258,] [284]]

[线性映射, [48]] [中性元, [36]]

[线性规划, [239]] [不可逆的, [24]]

[线性子空间, [39]] [非奇异的, [24]]

[线性变换, [48]] [范数, [71]]

[线性相关, [40]] [正态分布, [197]]

[线性无关, [40]] [正规方程, [86]]

[链接函数, [272]] [法向量, [80]]

[载荷, [322]] [零空间, [33,] [47,] [58]]

[局部最小值, [225]] [分子布局, [150]]

[对数配分函数, [211]] [奥卡姆剃刀, [285]]

[逻辑回归, [315]] [ONB, [79]]

[逻辑sigmoid函数, [315]] [独热编码, [364]]

[损失函数, [260,] [381]] [有序基, [50]]

[损失项, [382]] [正交的, [77]]

[下三角矩阵, [101]] [正交基, [79]]

[麦克劳林级数, [143]] [正交补, [79]]

[曼哈顿范数, [71]] [正交矩阵, [78]]

[MAP, [300]] [标准正交的, [77]]

[MAP估计, [269]] [标准正交基, [79]]

©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 剑桥大学出版社出版 (2020)。

[410] 索引

[外积, [38]] [[正则化参数, [263,] [302,] [380]]]

[过拟合, [262,] [271,] [299]] [正则化最小二乘, [302]]

[PageRank, [114]] [正则化器, [263,] [302,] [380,] [382]]

[参数, [61]] [representer theorem, [384]]

[参数方程, [61]] [责任, [352]]

[偏导数, [146]] [反向模式, [161]]

[特解, [27,] [30]] [右奇异向量, [119]]

[PCA, [317]] [RMSE, [298]]

[pdf, [181]] [均方根误差, [298]]

[惩罚项, [263]] [旋转, [91]]

[主元, [30]] [旋转矩阵, [92]]

[平面, [62]] [行, [22]]

[plate, [281]] [行向量, [22,] [38]]

[总体均值和协方差, [191]] [行阶梯形式, [30]]

[正定, [71,] [73,] [74,] [76]] [样本均值, [192]]

[后验, [185,] [269]] [样本空间, [175]]

[后验比, [287]] [标量, [37]]

[幂迭代, [334]] [标量积, [72]]

[幂级数表示, [145]] [sigmoid, [213]]

[PPCA, [340]] [相似, [56]]

[预条件子, [230]] [奇异, [24]]

[预测器, [12,] [255]] [奇异值分解, [119]]

[原问题, [234]] [奇异值方程, [124]]

[主成分, [322]] [奇异值矩阵, [119]]

[主成分分析, [136,] [317]] [奇异值, [119]]

[主子空间, [327]] [松弛变量, [379]]

[先验, [185,] [269]] [软间隔SVM, [379,] [380]]

[先验比, [287]] [解, [20]]

[概率逆, [186]] [张成, [44]]

[概率PCA, [340]] [特解, [27]]

[概率编程, [278]] [谱聚类, [136]]

[概率, [175]] [谱范数, [131]]

[概率密度函数, [181]] [谱定理, [111]]

[概率分布, [172]] [谱, [106]]

[概率积分变换, [217]] [方阵, [25]]

[概率质量函数, [178]] [标准基, [45]]

[乘积规则, [184]] [标准差, [190]]

[投影, [82]] [标准正态分布, [198]]

[投影误差, [88]] [标准化, [336]]

[投影矩阵, [82]] [统计独立性, [194]]

[伪逆, [86]] [统计学习理论, [265]]

[随机变量, [172,] [175]] [随机梯度下降, [231]]

[值域, [58]] [强对偶性, [236]]

[秩, [47]] [充分统计量, [210]]

[秩亏, [47]] [和规则, [184]]

[秩-] [支撑点, [61]] [k] [近似, [130]]

[秩-零化子定理, [60]] [支持向量, [384]]

[方差原始分数公式, [193]] [支撑超平面, [242]]

[识别网络, [344]] [满射, [48]]

[重构误差, [88,] [327]] [SVD, [119]]

[简化凸包, [388]] [SVD定理, [119]]

[简化行阶梯形式, [31]] [对称, [73,] [76]]

[简化SVD, [129]] [对称矩阵, [25]]

[REF, [30]] [对称正定, [74]]

[回归, [289]] [[对称半正定, [74]]]

[正则, [24]] [线性方程组, [20]]

[正则化, [262,] [302,] [382]] [目标空间, [175]]

[《机器学习数学基础》草稿(2023-10-18)。反馈：] [<https://mml-book.com>][.] [索引] [411]

[Taylor多项式, [142,] [166]]

[Taylor级数, [142]]

[测试误差, [300]]

[测试集, [262,] [284]]

[Tikhonov正则化, [265]]

[迹, [103]]

[训练, [12]]

[训练误差, [300]]

[训练集, [260,] [292]]

[传递函数, [315]]

[变换矩阵, [51]]

[平移向量, [63]]

[转置, [25,] [38]]

[三角不等式, [71,] [76]]

[截断SVD, [129]]

[Tucker分解, [136]]

[欠拟合, [271]]

[无向图模型, [283]]

[均匀分布, [182]]

[单变量, [178]]

[无迹变换, [170]]

[上三角矩阵, [101]]

[验证集, [263,] [284]]

[变量选择, [316]]

[方差, [190]]

[向量, [37]]

[向量加法, [37]]

[向量空间, [37]]

[向量空间同态, [48]]

[带内积的向量空间, [73]]

[向量量子空间, [39]]

[弱对偶性, [235]]

[零一损失, [381]]

[©2023 M. P. Deisenroth, A. A. Faisal, C. S. Ong. 由Cambridge University Press出版 (2020).]

...

1. $n \times n \leftrightarrow$

2. $m \times n \leftrightarrow$

3. $m \times n \leftrightarrow$

4. $m \times n \leftrightarrow$

5. $n \times k \leftrightarrow$

6. $m \times n \leftrightarrow$

7. $m \times n \leftrightarrow$

8. $4 \times 3 \leftrightarrow$

9. $m \times n \leftrightarrow$

10. $2 \leftrightarrow$

11.3↩

12.1↩