



第一章 简介与动机

机器学习是指设计算法，自动从数据中提取有价值的信息。这里强调的是“自动”，即机器学习关注的是可应用于许多数据集的通用方法，同时产生有意义的东西。机器学习的核心有三个概念：数据、模型和学习。

由于机器学习本质上是数据驱动的，因此数据是机器学习的核心。机器学习的目标是设计通用方法，从数据中提取有价值的模式，理想情况下无需太多特定领域的专业知识。例如，给定一个大型文档语料库（如许多图书馆中的书籍），机器学习方法可用于自动查找文档中共享的相关主题（Hoffman et al., 2010）。为了实现这一目标，我们设计的模型通常与生成数据的过程相关，类似于我们得到的数据集。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。套用 Mitchell (1997) 的说法：如果一个模型在考虑了数据之后，在给定任务上的表现有所改善，那么这个模型可以说是从数据中学习的。我们的目标是找到能很好地泛化到我们将来可能会关注的未见数据的好模型。学习可以理解为一种通过优化模型参数来自动发现数据中的模式和结构的学习方法。

虽然机器学习已经有了很多成功案例，而且设计和训练丰富灵活的机器学习系统的软件也很容易获得，但我们认为，机器学习的数学基础对于理解构建更复杂机器学习系统的基本原理非常重要。了解这些原理有助于创建新的机器学习解决方案、理解和调试现有方法，以及了解我们正在使用的方法的固有假设和局限性。

[下一章节 >](#)
第二章 线性代数



1.1 为直觉寻找词语

我们在机器学习中经常面临的一个挑战是，概念和词语都很模糊，机器学习系统的某个特定组件可以抽象为不同的数学概念。例如，在机器学习中，“算法”一词至少有两种不同的含义。在第一种意义上，我们使用“机器学习算法”来指根据输入数据进行预测的系统。我们将这些算法称为预测器。在第二种意义上，我们使用完全相同的短语“机器学习算法”来指一种系统，它可以调整预测器的某些内部参数，从而使其在未来未见的输入数据上表现良好。在这里，我们将这种调整称为训练系统。

本书不会解决含混不清的问题，但我们希望预先强调，根据上下文的不同，相同的表达方式可能有不同的含义。不过，我们会努力使上下文足够清晰，以减少歧义的程度。

本书的第一部分讨论机器学习系统的三个主要组成部分：数据、模型和学习所需的数学概念和基础。我们将在此简要概述这些组成部分，在讨论完必要的数学概念后，我们将在第 8 章中再次讨论它们。

虽然并非所有数据都是数字，但考虑数字格式的数据往往是有用的。在本书中，我们假定数据已经被适当地转换成适合读入计算机程序的数字表示形式。因此，我们将数据视为向量。向量是数据的另一种表现形式，它说明了文字是多么微妙，我们可以（至少）用三种不同的方式来思考向量：向量是一个数字数组（计算机科学观点），向量是一个有方向和大小的箭头（物理学观点），向量是一个服从加法和缩放的对象（数学观点）。

模型通常用于描述生成数据的过程，与手头的数据集类似。因此，好的模型也可以被看作是真实（未知）数据生成过程的简化版本，捕捉对建模数据并提取隐藏模式中至关重要的特征。好的模型可以用来预测真实世界中会发生的事情，而无需进行真实世界的实验。

现在我们来谈谈问题的关键，即机器学习的学习部分。假设我们得到了一个数据集和一个合适的模型。训练模型意味着利用现有数据优化模型的某些参数，而模型的参数与效用函数相关，效用函数用于评估模型对训练数据的预测效果。大多数训练方法可以看作是一种类似于爬山到达山顶的方法。在这种类比中，山顶对应的是某种所需的性能指标的最大值。然而，在实践中，我们希望模型在未见过的数据上表现良好。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，这可能并不能很好地推广到未见过的数据上，而且在实际应用中，我们经常需要让机器学习系统面对它以前从未遇到过的情况。

让我们总结一下本书中涉及的机器学习的主要概念：

- 我们将数据表示为向量。
 - 我们选择一个合适的模型，或者使用概率观点，或者使用优化观点。
 - 我们使用数值优化方法从可用数据中学习，目的是让模型在未使用或未训练的数据上表现良好。
-

[下一章节 >](#)

1.2 阅读本书的两种方法



1.2 阅读本书的两种方法

我们可以考虑采用两种策略来理解机器学习的数学知识：一种是将数学知识应用于机器学习，另一种是将数学知识应用到机器学习。

- 自下而上：从基础概念到高级概念。在数学等技术性较强的领域，这通常是首选方法。这种策略的好处是，读者在任何时候都能依靠以前学过的概念。遗憾的是，对于实践者来说，许多基础概念本身并不特别有趣，而且缺乏动力，这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。这种以目标为导向的方法的优点是，读者随时都知道他们为什么需要学习某个特定的概念，而且所需知识的路径也很清晰。这种策略的缺点是，知识可能建立在不稳固的基础上，读者必须记住一系列他们根本无法理解的单词。

我们决定以模块化的方式编写本书，将基础（数学）概念与应用分开，这样就可以从两个方面阅读本书。本书分为两部分，第一部分奠定数学基础，第二部分将第一部分的概念应用于一系列基本的机器学习问题，如图 1.1 所示，这些问题构成了机器学习的四大支柱：回归、降维、密度估计和分类。第一部分的章节大多建立在前一章的基础上，但如果有必要，也可以跳过一章，向后学习。第二部分各章之间的联系并不紧密，可以按照任意顺序阅读。书中两部分之间有许多前后相互指引的内容，将数学概念与机器学习算法联系起来。

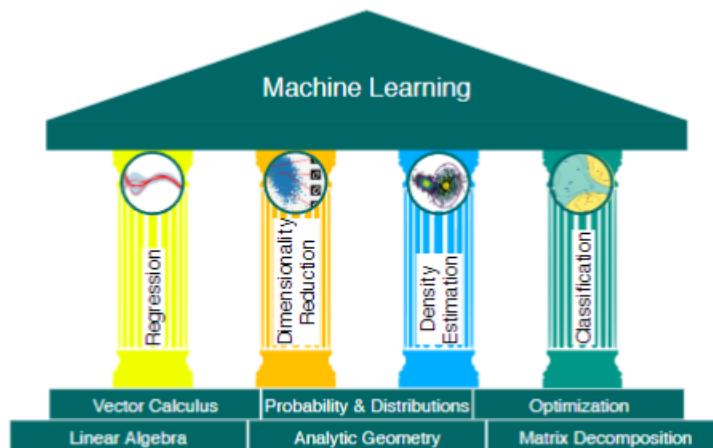


图1.1 机器学习的基础和四大支柱

当然，阅读本书的方法不止两种。大多数读者会采用自上而下和自下而上相结合的方法来学习，有时会先积累基本的数学技能，然后再尝试更复杂的概念，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

我们在本书中介绍的机器学习四大支柱（见图 1.1）需要坚实的数学基础，这将在第一部分中阐述。

我们用向量来表示数值数据，用矩阵来表示这些数据的表格。对向量和矩阵的研究称为线性代数，我们将在第 2 章介绍线性代数。该章还介绍了将向量集合为矩阵的方法。

给定代表现实世界中两个物体的两个向量，我们要对它们的相似性做出说明。我们的想法是，我们的机器学习算法（我们的预测器）应该预测相似的向量会有相似的输出。为了使向量间的相似性概念正规化，我们需要引入一些操作，将两个向量作为输入，并返回一个代表其相似性的数值。相似性和距离的构造是解析几何的核心，将在第 3 章中讨论。

在第 4 章中，我们将介绍有关矩阵和矩阵分解的一些基本概念。矩阵的一些运算在机器学习中非常有用，可以直观地解释数据，提高学习效率。

我们通常认为数据是对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，从噪声中识别出信号。这就要求我们有一种语言来量化“噪声”的含义。我们通常还希望预测器能让我们表达某种不确定性，例如，量化我们对特定测试数据点的预测值的信心。不确定性的量化属于概率论的范畴，将在第 6 章中介绍。

为了训练机器学习模型，我们通常要找到能最大化某些性能指标的参数。许多优化技术都需要梯度的概念，它告诉我们寻找解决方案的方向。第 5 章是关于向量微积分的内容，详细介绍了梯度的概念，随后我们将在第 7 章中使用梯度的概念来进行优化，找到函数的最大值/最小值。

第二部分 关于机器学习

本书第二部分介绍了机器学习的四大支柱，如图 1.1 所示。我们说明了本书第一部介绍的数学概念是如何为每个支柱奠定基础的。大体上，各章按难度排序（从高到低）。



在第 8 章中，我们以数学的方式重述了机器学习的三个组成部分（数据、模型和参数估计）。此外，我们就如何设计机器学习实验评估方案提供了若干指导原则，以防止对机器学习系统的评估过于乐观。回顾一下，我们的目标是建立一个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，我们的目标是找到将输入 $\mathbf{x} \in \mathbb{R}^D$ 映射到或对应的观测函数值 $y \in \mathbb{R}$ 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估计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，在贝叶斯线性回归中，我们对参数进行积分而不是优化。

第 10 章的重点是利用主成分分析法降维，即图 1.1 中的第二个支柱。降维的主要目的是为高维数据 $\mathbf{x} \in \mathbb{R}^D$ 找到一个紧凑的低维表示，它通常比原始数据更容易分析。与回归不同的是，降维只关注数据建模--数据点 \mathbf{x} 没有相关标签。

在第 11 章中，我们将转向第三个支柱：密度估计。密度估计的目标是找到描述给定数据集的概率分布。为此，我们将重点关注高斯混合模型，并讨论一种迭代方案来找到该模型的参数。与降维一样，数据点 $\mathbf{x} \in \mathbb{R}^D$ 没有相关标签。然而，我们并不寻求数据的低维表示。相反，我们感兴趣的是能描述数据的密度模型。

第 12 章以深入讨论第四个支柱：分类作为本书的结尾。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 \mathbf{x} 和相应的标签 y 。然而，与回归不同的是，回归中的标签是实值，而分类中的标签是整数，这就需要特别注意。

< 上一章节

1.1 为直觉寻找词语

下一章节 >

1.3 练习和反馈



1.3 练习和反馈

我们在第一部分提供了一些练习，这些练习主要可以通过纸笔完成。在第二部分中，我们提供了编程教程（jupyter notebook），用于探索本书讨论的机器学习算法的一些特性。

剑桥大学出版社大力支持我们实现教育和学习民主化的目标，在以下网址免费提供本书的下载，对此我们深表感谢：

<https://mml-book.com>

在此可找到教程、勘误表和其他资料。可使用前述 URL 报告错误并提供反馈。

◀ 上一章节

1.2 阅读本书的两种方法



线性代数

译者：马世拓、何瑞杰

这一章是后续很多概念的基础，我国工科生在本科阶段需要强制学习线性代数课程，但很多同学对学校的线性代数课程感到有些云里雾里。所以我这里对这一章进行了一些翻译与补充。

形成直观概念的一种常见方法是构建一系列的符号对象以及针对这些对象的规则。这就是我们所知道的**代数学**。线性代数是一门研究向量与向量运算法则的学科。我们在中学阶段所熟知的“向量”被称为**几何向量**，通常会用小箭头作标记，例如 \vec{x}, \vec{y} 等。在本书中我们讨论的是更为一般的向量概念并用粗体来表示它们，比如 \mathbf{x}, \mathbf{y} 等。

一般来说，向量这种特殊的对象可以进行叠加，并且乘以标量后会产生新的同类型对象。从抽象的数学角度来看，任何满足这两个属性的对象都可以被认为是一个向量。下面是一些这样的向量对象的例子：

1. **几何向量**。这种定义下的向量案例对于有中学数学和物理基础的人来说再熟悉不过了。如图2.1(a)所示，几何向量在图中被表示为一个至少有两个维度的有向线段。两个几何向量 \vec{x}, \vec{y} 可以相加，例如 $\vec{x} + \vec{y} = \vec{z}$ 就是一个新的几何向量。进一步地，一个几何向量 \vec{x} 乘上一个标量 $\lambda \in \mathbb{R}$ 变为 $\lambda\vec{x}$ ，结果仍然是一个几何向量。事实上，它是由原向量放缩 λ 倍得到的结果。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**。如图2.1(b)所示，两个多项式加在一起可以进而产生新的多项式；它们也可以用一个标量 $\lambda \in \mathbb{R}$ 去乘，结果同样是一个新的多项式。因此，多项式是（不太寻常的）向量实例。要注意到多项式与几何向量有很大不同。几何向量是具体的图形，而多项式是抽象概念。然而，它们都是我们前面描述的向量。
3. **音频信号是向量**。音频信号用一系列的数字来表示。我们可以把音频信号加在一起，它们的总和就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的类型。

4. \mathbb{R}^n (n个实数组成的元组) 中的元素也是向量 (译者注: 这里我们往往也叫做“n维 Euclid 空间”)。 \mathbb{R}^n 是比多项式更抽象的概念, 也是我们在这本书中会聚焦的概念。例如:

$$\mathbf{a} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \in \mathbb{R}^3 \quad (2.1)$$

就是一个三元数组的实例。对两个向量 \mathbf{a}, \mathbf{b} 按分量相加会得到一个新的向量 $\mathbf{a} + \mathbf{b} = \mathbf{c} \in \mathbb{R}^n$ 。进一步说, 用一个标量 $\lambda \in \mathbb{R}$ 乘一个向量 \mathbf{a} 会得到一个放缩后的新向量 $\lambda\mathbf{a} \in \mathbb{R}^n$ 。将向量作为 \mathbb{R}^n 的元素有一个额外的好处, 就是能够自然对应于计算机上的实数数组。许多编程语言都支持数组操作, 这允许方便地实现涉及向量操作的算法。

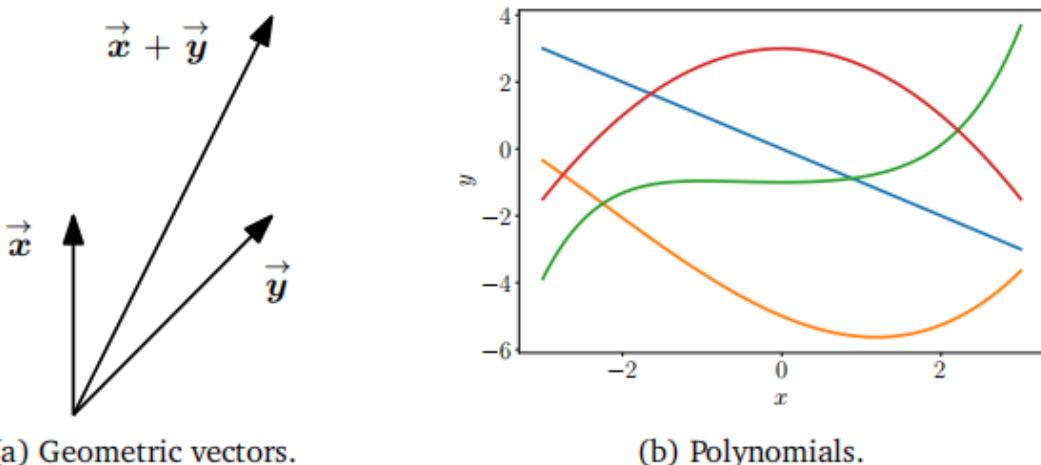


图2.1 不同类型的向量。向量可以是各种令人吃惊的对象, 包括(a)几何向量和(b)多项式

线性代数聚焦于这些向量概念之间的相似性。我们可以对这些向量进行加法或标量乘法。我们聚焦 \mathbb{R}^n 中的向量, 因为线性代数中的绝大部分算法都是在n维 Euclid 空间中形成的。我们会在第8章中看到我们也经常会把数据用 \mathbb{R}^n 中的向量来表示。在本书中, 我们会聚焦有限维线性空间, 在这种情况下任何一个向量在 \mathbb{R}^n 中存在唯一对应关系。在方便的时候, 我们也会使用有关几何向量的认知并考虑一些基于数组的算法。

“闭包”是数学中一个很重要的概念。这基于一个问题: 根据我设定的操作规则所得到的元素构成了一个怎样的集合? 对于向量而言, 将一个很小的向量集合经过相加与放缩操作后会得到一个怎样的向量集? 这就引出了线性空间的概念 (详见2.4节)。线

性空间的概念及其正确性是很大一部分机器学习的基础。这一章中要介绍的一些概念总结如图2.2所示。

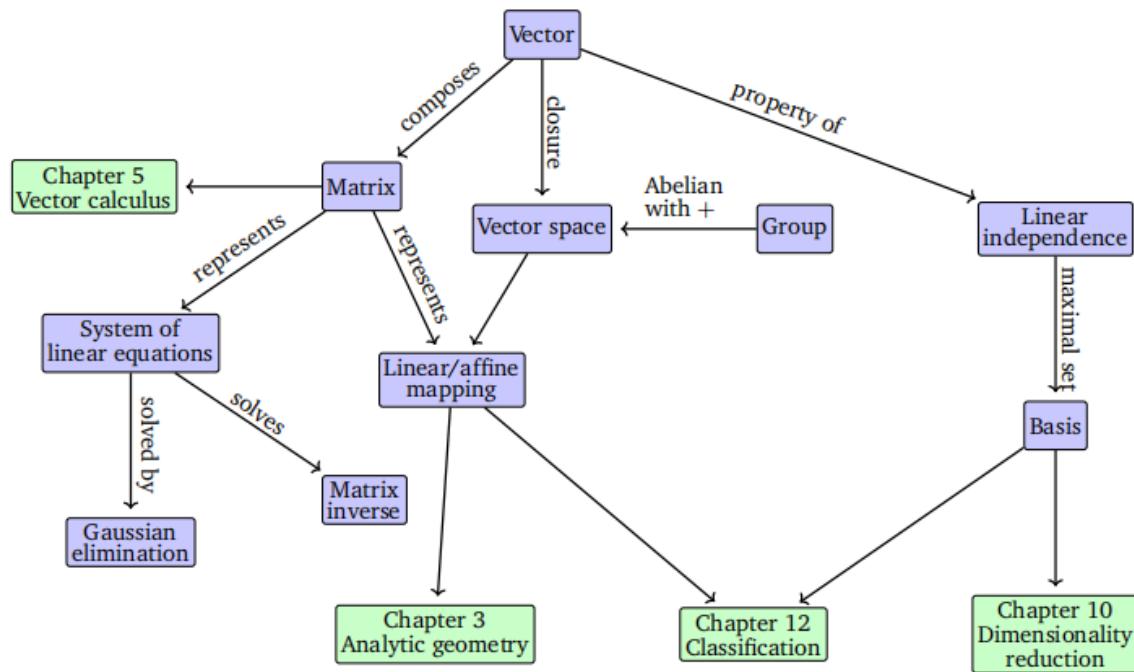


图2.2 本章的概念地图及与其他章节的联系

本章主要是基于下列作者的课堂笔记与著作: Drumm & Weil (2001), Strang (2003), Hogben (2013), Liesen & Mehrmann(2015), 还有Pavel Grinfeld的线性代数系列, 以及其他的好资源例如Gilbert Strang在MIT的线性代数课程以及3Blue1Brown的线性代数系列。

线性代数在机器学习与基础数学中扮演着重要角色。这一章引入的概念是对第3章中几何概念的更高扩充。在第5章中, 我们将讨论向量微积分, 这里对于矩阵运算法则的知识就很有必要了。在第10章中, 我们会使用投影 (在3.8节中会介绍) 通过主成分分析 (PCA) 进行降维。在第9章中, 我们会讨论线性回归, 线性代数在这里起到了了解最小二乘问题的主要作用。

< 上一章节

下一章节 >

第一章 简介与动机

第三章 解析几何



2.1 线性方程组

线性方程组是线性代数的核心部分。很多问题都可以用线性方程组表示，线性代数也为我们提供了解这类问题的方法。

例2.1 一家公司生产产品 N_1, N_2, \dots, N_n 需要用到原料 R_1, R_2, \dots, R_m 。生产一单位产品 N_j 所需要原料 R_i 的用量为 a_{ij} ，这里 $i=1,2,\dots,m; j=1,2,\dots,n$ 。

问题的目标是找到一组最优的生产方案：在原料 R_i 总可用量为 b_i 的条件下生产产品 N_j 的量 x_j 为多少时没有任何原料剩余。

如果我们生产 x_1, x_2, \dots, x_n 单位的对应品种产品，我们需要原材料 R_i 的总用量为：

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n. \quad (2.2)$$

可行解 $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ ，因此也就需要满足下列条件：

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned} \quad (2.3)$$

这里 $a_{ij} \in \mathbb{R}, b_i \in \mathbb{R}$

式2.3是线性方程组的通用表达形式， x_1, x_2, \dots, x_n 是方程组中的未知量。每个满足2.3式的n-元组 $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ 都是这个线性方程组的一个解。

例2.2 线性方程组

$$\begin{aligned} x_1 + x_2 + x_3 &= 3 \\ x_1 - x_2 + 2x_3 &= 2 \\ 2x_1 + 3x_3 &= 1 \end{aligned} \quad (2.4)$$



是没有解的。因为把第一个方程和第二个方程相加得到 $2x_1 + 3x_3 = 5$ 与第三个方程发生了冲突。

我们再来看这样一个方程组:

$$\begin{aligned}x_1 + x_2 + x_3 &= 3 \\x_1 - x_2 + 2x_3 &= 2 \\x_2 + x_3 &= 2\end{aligned}\tag{2.5}$$

第一个方程减掉第三个方程可以得到 $x_1 = 1$ 。第一个方程加第二个方程可以得到 $2x_1 + 3x_3 = 5$, 因此 $x_3 = 1$ 。根据第三个方程, 我们又得到 $x_2 = 1$ 。因此, $(1,1,1)$ 是唯一的可行解也就是唯一解 (可以通过代入法验证 $(1,1,1)$ 是方程组的一个解)。

第三个案例我们再来看这样一个方程组:

$$\begin{aligned}x_1 + x_2 + x_3 &= 3 \\x_1 - x_2 + 2x_3 &= 2 \\2x_1 + 3x_3 &= 5\end{aligned}\tag{2.6}$$

因为第一个方程和第二个方程相加得到第三个方程, 我们可以把第三个多余的方程消掉。从前两个方程中我们可以得到 $2x_1 = 5 - 3x_3$, $2x_2 = 1 + x_3$ 。我们定义 $x_3 = a \in \mathbb{R}^3$ 作为自由变量, 任意一个满足下列形式的三元组都是方程组的解:

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

因此, 我们得到了一个包含无穷个解的解集。

总的来说, 对于一个实数域内的线性方程组, 它要么无解, 要么有唯一解, 要么有无穷个解。线性回归 (第9章) 提供了一个求像例2.1这样无解线性方程组的 (近似) 解的一个方法。

注: 线性方程组的几何意义。在一个只有 x_1, x_2 两个变量的方程组中, 每个方程都被代表了 x_1-x_2 平面内的一条直线。线性方程组的解要分别满足其中所有方程里任意一个方程, 所以它同时也是这些直线的交点。交点可以组成一条直线 (如果两个方程描述的是同一条直线), 可以组成一个点, 或为空 (两条直线平行)。图2.3描述了下面这个线性方程组的几何表示:

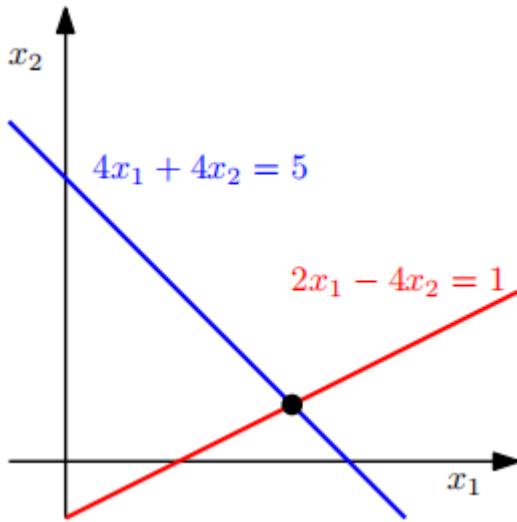


图2.3 两个变量线性方程组的解空间在几何意义上表示为两条线的交点。每个线性方程都代表一条直线

最终的解为 $(x_1, x_2) = (1, 1/4)$ 。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。这些平面相交形成的结果同时满足所有的线性方程，它们可以得到一个解集，可能是一个平面、一条线、一个点或为空（在这些平面没有公共的交点的情况下）。

为了引出解线性方程组的符号方法，我们介绍一种有效的缩写方法。我们将系数 a_{ij} 写作向量并将向量构造为矩阵。换而言之，我们将线性方程组改写为如下形式：

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

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

接下来，我们将对这些**矩阵**及其定义的运算规则作出进一步的探究。我们将在第2.3节中讲述线性方程组的解法。

[下一章节 >](#)

2.2 矩阵





2.2 矩阵

矩阵在线性代数中起到了关键作用。它们不仅可以表示线性方程组，还可以表示线性函数（或者线性映射），我们将在2.7节中看到。在我们讨论这些有趣的话题之前，我们首先要定义什么是矩阵以及我们可以对矩阵进行怎样的操作。我们会在第4章看到更多有关矩阵的性质。

定义2.1（矩阵）：对于 $m, n \in \mathbb{Z}_{>0}$ ，一个形状为 (m, n) 的实矩阵 \mathbf{A} 是一个关于元素 a_{ij} 的 $m \times n$ 元组，其中 $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ ，按照 m 行 n 列的方式进行排布。

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, a_{ij} \in \mathbb{R} \quad (2.11)$$

按照惯例，形状为 $(1, n)$ 的矩阵成为行向量，形状为 $(m, 1)$ 的矩阵称为列向量。

$\mathbb{R}^{m \times n}$ 是所有实值 (m, n) 矩阵的集合。通过将矩阵的所有 n 列叠加成一个长向量，一个 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 可以等价地表示为一个 $\mathbf{a} \in \mathbb{R}^{mn}$ ，如图2.4所示。

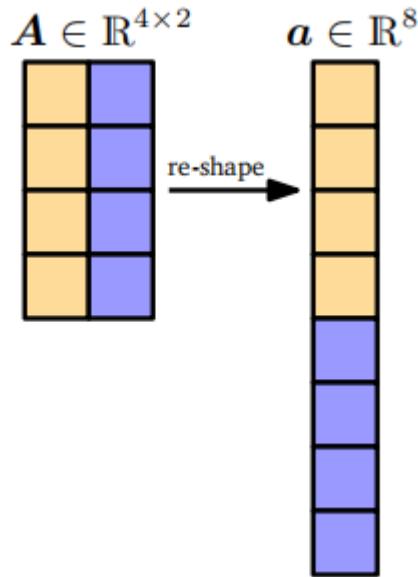


图2.4 通过叠加矩阵 \boldsymbol{A} 的列，矩阵 \boldsymbol{A} 可以表示为长向量 \boldsymbol{a} 。



2.2.1 矩阵的加法与乘法

两个矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{m \times n}$ 的和被定义为两个矩阵按对应元素的相加得到的新矩阵, 即:

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n} \quad (2.12)$$

对于矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times k}$ 的乘积矩阵 $\mathbf{C} = \mathbf{AB} \in \mathbb{R}^{m \times k}$ (注意这里矩阵的大小) 中元素的计算法则为:

$$c_{ij} = \sum_{l=1}^n a_{il}b_{lj}, (i = 1, 2, \dots, m, j = 1, 2, \dots, k) \quad (2.13)$$

也就是说, 为了计算元素 c_{ij} 我们用 \mathbf{A} 的第*i*行和 \mathbf{B} 的第*j*列元素逐项相乘并求和。在后续的3.2节中, 我们把这种操作称作对应行与列的**内积**。在我们需要显式地执行乘法的情况下, 我们使用符号 $\mathbf{A} \cdot \mathbf{B}$ 来表示乘法 (显式地表示“.”)。

注意: 矩阵只有在“相邻”的尺寸匹配时才可以相乘。例如, 一个大小为 $n \times k$ 的矩阵 \mathbf{A} 可以与一个 $k \times m$ 大小的矩阵 \mathbf{B} 相乘, 但只能从左边乘:

$$\underbrace{\mathbf{A} \quad \mathbf{B}}_{n \times k \quad k \times m} = \underbrace{\mathbf{C}}_{n \times m} \quad (2.14)$$

而乘积 \mathbf{BA} 如果 $m \neq n$ 时则是不被定义的, 因为相邻的维度无法匹配。

注意: 矩阵乘法并不是对矩阵的逐元素乘法, 即: $c_{ij} \neq a_{ij}b_{ij}$ (即使 \mathbf{A}, \mathbf{B} 的尺寸被正确选择)。当我们进行多维数组之间的乘法时, 这种按位的乘法往往也出现在编程语言中, 它被称作 Hadamard 积。(译者注: Hadamard 积是指两个矩阵对应元素相乘得到的新矩阵, 例如 $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ 和 $\mathbf{B} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$ 的 Hadamard 积为 $\mathbf{A} \circ \mathbf{B} = \begin{bmatrix} 5 & 12 \\ 21 & 32 \end{bmatrix}$)

例2.3 对于矩阵 $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{bmatrix} \in \mathbb{R}^{2 \times 3}$, $\mathbf{B} = \begin{bmatrix} 0 & 2 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{3 \times 2}$, 我们有

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

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

从这个例子中我们可以看到，矩阵的乘法并不具备交换律，即： $\mathbf{AB} \neq \mathbf{BA}$ 。图2.5给出了它的几何解释。

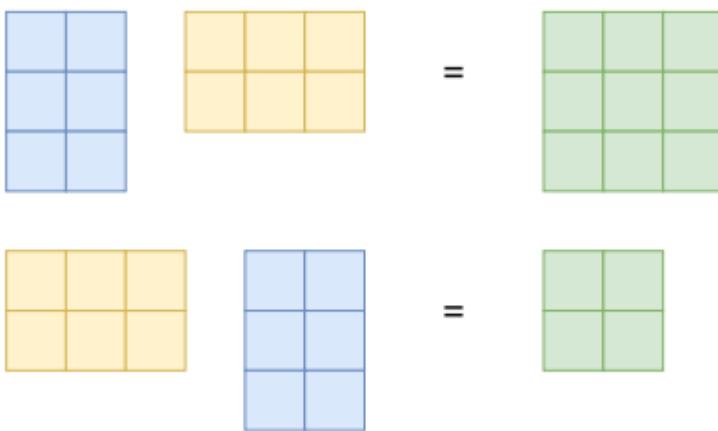


图2.5 即使同时定义了矩阵乘法 \mathbf{AB} 和 \mathbf{BA} ，结果的维数也可能是不同的。

定义（单位矩阵）：在 $\mathbb{R}^{n \times n}$ 中，定义 **单位矩阵**

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{n \times n} \quad (2.17)$$

为对角线上全部为1，其他位置全部为0的 $n \times n$ 维矩阵。

现在，我们定义了矩阵的加法、乘法和单位矩阵，让我们来看看它们的运算性质：

- **结合律：**

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

- **分配律：**

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

$$\mathbf{A}(\mathbf{C} + \mathbf{D}) = \mathbf{AC} + \mathbf{AD} \quad (2.19b)$$

- 与单位矩阵相乘:

$$\forall \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{I}_m \mathbf{A} = \mathbf{A} \mathbf{I}_n = \mathbf{A} \quad (2.20)$$

注意, 由于 $m \neq n$ 所以 $\mathbf{I}_m \neq \mathbf{I}_n$

2.2.2 矩阵的逆与转置

定义2.3 (逆矩阵) : 考虑一个方阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$, 令矩阵 \mathbf{B} 满足性质: $\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n$, \mathbf{B} 被称作 \mathbf{A} 的逆并记作 \mathbf{A}^{-1} 。

不幸的是, 并不是每个矩阵 \mathbf{A} 都存在逆矩阵 \mathbf{A}^{-1} 。如果这个逆存在, 矩阵 \mathbf{A} 被称作可逆矩阵/非奇异矩阵, 否则就叫作不可逆矩阵/奇异矩阵。如果一个矩阵的逆存在, 那么它也必然唯一。在2.3节中, 我们将会讨论一种通过求解线性方程组的解计算矩阵逆的通用方法。

注意: (2×2 矩阵逆的存在性) 考虑一个矩阵

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

如果我们对矩阵 \mathbf{A} 乘上:

$$\mathbf{A}' = \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \in \mathbb{R}^{2 \times 2} \quad (2.22)$$

我们就会得到:

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

因此,

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

当且仅当 $a_{11}a_{22} - a_{12}a_{21} \neq 0$ 。

在4.1节中，我们会看到 $(a_{11}a_{22} - a_{12}a_{21})$ 是这个 2×2 矩阵的行列式。此外，我们通常可以使用这个行列式来检查一个矩阵是否可逆。

例2.4 矩阵

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

互为逆矩阵，因为 $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$ 。

定义2.4 (转置)：对于矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ ，满足 $b_{ij} = a_{ji}$ 的矩阵 $\mathbf{B} \in \mathbb{R}^{n \times m}$ 被称作 \mathbf{A} 的转置。我们记 $\mathbf{B} = \mathbf{A}^\top$

总的来说， \mathbf{A}^\top 可以通过把 \mathbf{A} 的行作为 \mathbf{A}^\top 的对应列得到（译者注：这里其实通俗来讲就是行列互换）。下面是一些有关逆与转置的重要性质：

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

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

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

$$(\mathbf{A}^\top)^\top = \mathbf{A} \quad (2.29)$$

$$(\mathbf{A} + \mathbf{B})^\top = \mathbf{A}^\top + \mathbf{B}^\top \quad (2.30)$$

$$(\mathbf{AB})^\top = \mathbf{B}^\top \mathbf{A}^\top \quad (2.31)$$

注意

- 矩阵的**主对角线**是从左上角到右下角的对角线，包含元素 a_{ii} , $i = 1, 2, \dots, n$

2. 式 (2.28) 的实数版本是诸如 $\frac{1}{2+4} \neq \frac{1}{2} + \frac{1}{4}$ 这样的不等式。

定义2.5 (对称矩阵)：一个矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 若满足 $\mathbf{A} = \mathbf{A}^\top$ ，我们称其为**对称矩阵**。

注意只有 $n \times n$ 矩阵才可能具备对称性。通常来说，我们也把 $n \times n$ 矩阵叫**方阵**因为它具备相同的行数和列数。进一步的，如果矩阵 \mathbf{A} 可逆， \mathbf{A}^\top 也可逆，那么 $(\mathbf{A}^{-1})^\top = (\mathbf{A}^\top)^{-1}$ ，记作 \mathbf{A}^{-T}

注意（对称矩阵的和与积）。对称矩阵 $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ 的和矩阵也是对称的。但是，尽管二者的积存在，结果却通常是不对称的。例如：

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

2.2.3 矩阵的标量乘

让我们来看看如果矩阵乘上一个标量 $\lambda \in \mathbb{R}$ 会发生什么吧。令 $\mathbf{A} \in \mathbb{R}^{m \times n}, \lambda \in \mathbb{R}$ ，那么 $\lambda \mathbf{A} = \mathbf{K}, K_{ij} = \lambda a_{ij}$ 。实际上， λ 对矩阵 \mathbf{A} 中每个元素进行了放缩。对于 $\lambda, \psi \in \mathbb{R}$ ，有如下性质：

- 结合律1

$$(\lambda\psi)\mathbf{C} = \lambda(\psi\mathbf{C}), \mathbf{C} \in \mathbb{R}^{m \times n}$$

- 结合律2

$$\lambda(\mathbf{BC}) = (\lambda\mathbf{B})\mathbf{C} = \mathbf{B}(\lambda\mathbf{C}) = (\mathbf{BC})\lambda, \mathbf{B} \in \mathbb{R}^{m \times n}, \mathbf{C} \in \mathbb{R}^{n \times k}$$

- 转置

$$(\lambda\mathbf{C})^\top = \mathbf{C}^\top \lambda^\top = \mathbf{C}^\top \lambda = \lambda\mathbf{C}^\top$$

因为对于 $\forall \lambda \in \mathbb{R}, \lambda^\top = \lambda$

- 分配律

$$\begin{aligned} (\lambda + \psi)\mathbf{C} &= \lambda\mathbf{C} + \psi\mathbf{C}, \\ \lambda(\mathbf{B} + \mathbf{C}) &= \lambda\mathbf{B} + \lambda\mathbf{C} \end{aligned} \quad \mathbf{B} \in \mathbb{R}^{m \times n}, \mathbf{C} \in \mathbb{R}^{m \times n},$$

例2.5（分配律） 如果我们令

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

，对于任意 $\lambda, \psi \in \mathbb{R}$ 都有：

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

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

2.2.4 线性方程组的矩阵表示

如果我们考虑这样一个线性方程组：

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

利用矩阵乘法的规则，我们可以把这个方程组写成更紧凑的形式：

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

注意， x_1 缩放了第一列， x_2 是第二列， x_3 是第三列。

一般的，一个线性方程组可以缩写为矩阵形式 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 。参考2.3节，乘积 $\mathbf{A}\mathbf{x}$ 是对 \mathbf{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。

< 上一章节

下一章节 >

2.1 线性方程组

2.3 解线性方程组



2.3 解线性方程组

在(2.3)中，我们介绍了方程组的一般形式，即

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

其中 $a_{i,j} \in \mathbb{R}$ 和 $b_i \in \mathbb{R}$ 是已知常数，而 x_j 是未知数， $i = 1, \dots, m$, $j = 1, \dots, n$ 。到目前为止，我们已经看到矩阵可以作为一种紧凑的方式来表述线性方程组，从而可以写成 $Ax = b$ ，见(2.10)。此外，我们还定义了矩阵的基本运算，例如矩阵的加法和乘法。在接下来的内容中，我们将专注于解线性方程组，并提供一种求矩阵逆的算法。

2.3.1 特解和通解

在讨论如何一般性地解线性方程组之前，我们先来看一个例子。考虑以下方程组：

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

该方程组有两条方程和四个未知数。因此，一般来说，我们期望有无穷多解。这个方程组的形式特别简单，前两列分别由1和0组成。我们希望找到标量 x_1, \dots, x_4 ，使得 $\sum_{i=1}^4 x_i c_i = b$ ，其中我们将矩阵的第 i 列定义为 c_i ，而 b 是(2.38)的右侧。通过取42倍的第一列和8倍的第二列，可以立即找到(2.38)的一个解：

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

因此，一个解是 $[42, 8, 0, 0]^\top$ 。这个解被称为**特解**（或**特殊解**）。然而，这并不是这个线性方程组的唯一解。为了找到所有其他解，我们需要创造性地用矩阵的列以非平凡的方式生成0：将0加到特解上不会改变特解。为此，我们用前两列（它们的形式非常简单）来表示第三列：

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

(2.40) 

因此如果我们将原矩阵 $\begin{bmatrix} 1 & 0 & 8 & -4 \\ 0 & 1 & 2 & 12 \end{bmatrix}$ 的所有列从左到右分别记作向量 c_1, c_2, c_3, c_4 , 我们可以写出:

$$0 = 8c_1 + 2c_2 - c_3 + 0c_4,$$

并且 $(x_1, x_2, x_3, x_4) = (8, 2, -1, 0)$ 。

事实上, 将这个解按任意标量 $\lambda_1 \in \mathbb{R}$ 缩放都会产生 $\mathbf{0}$ 向量, 即

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

按照相同的思路, 我们将矩阵的第四列用前两列表示, 并生成另一组非平凡的 $\mathbf{0}$:

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

对于任意 $\lambda_2 \in \mathbb{R}$ 。将所有内容放在一起, 我们得到(2.38)方程组的所有解, 称为通解:

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

注: 我们遵循的一般方法包括以下三个步骤:

1. 找到一个特解 $Ax = b$ 。
2. 找到所有解 $Ax = \mathbf{0}$ 。
3. 将步骤1和步骤2的解结合起来得到通解。

通解和特解都不是唯一的。 (译者注: 假如矩阵 A 的列向量线性相关, 可能会有无穷多的特解和通解。)

在前面的例子中，方程组很容易解，因为矩阵(2.38)具有这种特别方便的形式，允许我们通过观察找到特解和通解。然而，一般方程组并不具有这种简单形式。幸运的是，存在一种构造性算法，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**初等变换**，这些变换可以将方程组转换为更简单的形式，同时保持解集不变。然后，我们可以将这种简单形式应用到我们在(2.38)的例子中讨论的三个步骤。

2.3.2 初等变换

解线性方程组的关键是**初等变换**，这些变换可以保持方程组的解集不变，但可以将方程组转换为更简单的形式：

- 交换两个方程（矩阵中的行）。
- 将一个方程（行）乘以一个非零常数 $\lambda \in \mathbb{R} \setminus \{0\}$ 。
- 将两个方程（行）相加。

例2.6 对于 $a \in \mathbb{R}$ ，我们寻找以下方程组的所有解：

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

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

$$\left[\begin{array}{ccccc|c} -2 & 4 & -2 & -1 & 4 & -3 \\ 4 & -8 & 3 & -3 & 1 & 2 \\ 1 & -2 & 1 & -1 & 1 & 0 \\ 1 & -2 & 0 & -3 & 4 & a \end{array} \right].$$

其中我们用竖线分隔了(2.44)的左侧和右侧。我们用 \Rightarrow 表示通过初等变换对增广矩阵的转换。增广矩阵 $[A | b]$ 紧凑地表示了线性方程组 $Ax = b$ 。

交换第 1 行和第 3 行后得到：



$$\left[\begin{array}{ccccc|c} 1 & -2 & 1 & -1 & 1 & 0 \\ 4 & -8 & 3 & -3 & 1 & 2 \\ -2 & 4 & -2 & -1 & 4 & -3 \\ 1 & -2 & 0 & -3 & 4 & a \end{array} \right].$$

接下来，我们对第 2 行、第 3 行和第 4 行分别进行以下变换：

- 第 2 行减去 4 倍的第 1 行；
- 第 3 行加上 2 倍的第 1 行；
- 第 4 行减去第 1 行。

得到：

$$\left[\begin{array}{ccccc|c} 1 & -2 & 1 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & -3 & 2 \\ 0 & 0 & 0 & -3 & 6 & -3 \\ 0 & 0 & -1 & -2 & 3 & a \end{array} \right].$$

接着，我们对第 4 行进行以下变换：

- 第 4 行减去第 2 行；
- 第 4 行减去第 3 行。

得到：

$$\left[\begin{array}{ccccc|c} 1 & -2 & 1 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & -3 & 2 \\ 0 & 0 & 0 & -3 & 6 & -3 \\ 0 & 0 & 0 & 0 & 0 & a+1 \end{array} \right].$$

最后，我们将第 2 行和第 3 行分别乘以 -1 和 $-\frac{1}{3}$ ，得到：

$$\left[\begin{array}{ccccc|c} 1 & -2 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 3 & -2 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & a+1 \end{array} \right].$$

这个（增广）矩阵处于一个方便的形式，即**行阶梯形式（REF）**。将这个紧凑的表示形式重新转换为包含变量的显式形式，我们得到：



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

只有当 $a = -1$ 时，这个方程组才有解。一个特解是：

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

通解（包含所有可能的解）是：

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

在接下来的内容中，我们将详细说明一种构造性方法来获得线性方程组的特解和通解。

注（主元和阶梯结构）： 按照上述方法将方程组化简完成后，某一行从左边开始的第一个非零数字称为主元，它总是严格位于其上方行的主元的右边。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）： 如果一个矩阵满足以下条件，则称其处于行阶梯形式：

- 所有只包含零的行位于矩阵的底部；相应地，所有至少包含一个非零元素的行位于只包含零的行的上方。
- 只考虑非零行，从左边开始的第一个非零数字（也称为主元或领先系数）总是严格位于其上方行的主元的右边。

在其他文献中，有时要求主元是1。

注（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，其他变量称为自由变量。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（获得特解）： 行阶梯形式使我们的生活更简单，当我们需要确定一个特解时。为此，我们用主元列表示方程组的右侧，使得 $b = \sum_{i=1}^P \lambda_i p_i$ ，其中 $p_i, i = 1, \dots, P$ ，是主元列。通过从最右边的主元列开始，向左工作，可以最简单地确定 λ_i 。在前面的例子中，我们将尝试找到 $\lambda_1, \lambda_2, \lambda_3$ ，使得

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

从这里，我们直接找到 $\lambda_3 = 1, \lambda_2 = -1, \lambda_1 = 2$ 。当我们把所有内容放在一起时，我们不能忘记非主元列（译者注：非主元列是一列的所有元素都不是任意一行的主元的列），我们隐式地将系数设置为0。因此，我们得到特解 $x = [2, 0, -1, 1, 0]^\top$ 。◆

注（简化行阶梯形式）： 如果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规范形式），则：

- 它处于行阶梯形式。
- 每个主元是**1**。
- 主元是其列中唯一的非零元素。

简化行阶梯形式将在第2.3.3节中发挥重要作用，因为它允许我们以直接的方式确定线性方程组的通解。高斯消元法是一种执行初等变换以将线性方程组带入简化行阶梯形式的算法。◆

例2.7（简化行阶梯形式） 验证以下矩阵处于简化行阶梯形式（主元以粗体表示）：

$$A = \begin{bmatrix} \mathbf{1} & 3 & 0 & 0 & 3 \\ 0 & 0 & \mathbf{1} & 0 & 9 \\ 0 & 0 & 0 & \mathbf{1} & -4 \end{bmatrix}. \quad (2.49)$$

找到 $Ax = \mathbf{0}$ 的解的关键思想是查看非主元列，我们需要将它们表示为（线性）组合的主元列。简化行阶梯形式使这相对直接，我们用它们左边的主元列的和与倍数来表示非主元列：第二列是第一列的3倍（我们可以忽略第二列右边的主元列）。因此，为了得到尽可能多的**0**，我们做下面的操作

1. 第二列减去三倍第一列：将第二列的**3**变成**0**
2. 第五列减去九倍第三列，减去四倍第四列

这样我们就得到了一个简化行阶梯形式的矩阵。总之，我们仍然在解一个齐次方程组 $Ax = 0$ 其中 $x \in \mathbb{R}^5$ 。通过第一步，我们知道原方程的解集一定包含 $[3, -1, 0, 0, 0]^\top$ ；通过第二步，我们知道原方程的解集一定还包含 $[3, 0, 9, -4, -1]^\top$ 。做完这两步后矩阵已经化成了简化行阶梯形，因此原方程的解集为：

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

2.3.3 负一技巧

在接下来的内容中，我们介绍一个实用的技巧，用于读出齐次线性方程组 $Ax = 0$ 的解，其中 $A \in \mathbb{R}^{k \times n}$, $x \in \mathbb{R}^n$ 。首先，我们假设 A 处于简化行阶梯形式，且没有只包含零的行，即

$$A = \begin{bmatrix} 0 & \cdots & 0 & \mathbf{1} & * & \cdots & * & 0 & * & \cdots & * & 0 & * & \cdots & * \\ \vdots & & \vdots & 0 & 0 & \cdots & 0 & \mathbf{1} & * & \cdots & * & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & \vdots & \cdots & \vdots & 0 & \vdots & \cdots & \vdots & \vdots & \vdots & & \vdots \\ \vdots & & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & 0 & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \mathbf{1} & * & \cdots & * \end{bmatrix} \quad (2.51)$$

alt text

其中 $*$ 可以是任意实数，条件是每行的第一个非零元素必须是 $\mathbf{1}$ ，且相应列中的所有其他元素必须是 0 。包含主元（以粗体标记）的列 j_1, \dots, j_k 是标准单位向量 $e_1, \dots, e_k \in \mathbb{R}^k$ 。我们通过添加 $n - k$ 行的形式

$$[0 \ \cdots \ 0 \ -1 \ 0 \ \cdots \ 0] \quad (2.52)$$

将这个矩阵扩展为 $n \times n$ 矩阵 \tilde{A} ，使得 \tilde{A} 的对角线上包含 1 或 -1 。然后，包含对角线上的 -1 的 \tilde{A} 的列是齐次方程组 $Ax = 0$ 的解。更准确地说，这些列构成了 $Ax = 0$ 的解空间的一个基，我们稍后将称其为核或零空间（见第2.7.3节）。

例2.8（负一技巧） 让我们重新审视已经处于简化REF的矩阵(2.49)：



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

我们通过在对角线上缺少主元的位置添加形式为(2.52)的行，将这个矩阵扩展为 5×5 矩阵：

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

从这个形式中，我们可以直接读出 $Ax = 0$ 的解，通过取 \tilde{A} 中对角线上包含-1的列：

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

这与我们在(2.50)中通过“洞察”得到的解完全一致。

计算逆矩阵

为了计算 A^{-1} ，其中 $A \in \mathbb{R}^{n \times n}$ ，我们需要找到一个矩阵 X ，使得 $AX = I_n$ 。然后， $X = A^{-1}$ 。我们可以将此写成一组同时线性方程 $AX = I_n$ ，其中我们解 $X = [x_1 | \dots | x_n]$ 。我们使用增广矩阵（译者注：就是把线性方程组中的所有数都写成一个矩阵，等号左边的系数矩阵放在左边，等号右边的数字放在右边）表示法来紧凑地表示这组方程组：

$$[A | I_n] \Rightarrow \dots \Rightarrow [I_n | A^{-1}]. \quad (2.56)$$

这意味着，如果我们将增广方程组带入简化行阶梯形式，我们可以在方程组的右侧直接读出逆矩阵。因此，确定矩阵的逆等同于解线性方程组。

例2.9（通过高斯消元法计算逆矩阵） 为了确定



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

的逆矩阵，我们写下增广矩阵

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

并使用高斯消元法将其带入简化行阶梯形式：

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

因此，所需的逆矩阵是其右侧：

$$A^{-1} = \begin{bmatrix} -1 & 2 & -2 & 2 \\ 1 & -1 & 2 & -2 \\ 1 & -1 & 1 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}. \quad (2.58)$$

我们可以通过执行乘法 AA^{-1} 并观察我们是否恢复 I_4 来验证(2.58)确实是逆矩阵。

2.3.4 解线性方程组的算法

在接下来的内容中，我们将简要讨论解决形式为 $Ax = b$ 的线性方程组的方法。我们假设存在解。如果没有解，我们需要诉诸于近似解，这在本章中没有涵盖。解决近似问题的一种方法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 A^{-1} ，使得解 $Ax = b$ 为 $x = A^{-1}b$ 。然而，这只在 A 是方阵且可逆的情况下才可能，然而这并不常见。否则在较弱的条件下（即 A 需要具有线性独立的列），我们可以使用变换

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

并使用 Moore-Penrose 伪逆 $(A^\top A)^{-1}A^\top$ 来确定解 $Ax = b$ ，这对应于最小范数最小二乘解。计算矩阵-矩阵乘积和 $A^\top A$ 的逆所需的计算量较大。此外，由于数值



精度的原因，通常不推荐计算逆或伪逆。因此，在接下来的内容中，我们将简要讨论解线性方程组的其他方法。高斯消元法在计算行列式（第4.1节）、检查一组向量是否线性独立（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）以及确定线性空间的基（第2.6.1节）中发挥重要作用。高斯消元法是一种直观且构造性的方法，用于解决具有数千个变量的线性方程组。然而，对于具有数百万变量的系统，这种方法是不切实际的，因为所需的算术运算次数与联立方程的数量呈立方关系。在实践中，具有许多线性方程的系统通常通过间接方法解决，例如固定迭代方法，如 Richardson 方法、Jacobi 方法、Gauss-Seidel 方法和逐次超松弛迭代法，或 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。我们推荐参考 Stoer 和 Burlirsch (2002)、Strang (2003) 和 Liesen 和 Mehrmann (2015) 的书籍以获取更多详细信息。设 x^* 是 $Ax = b$ 的解。这些迭代方法的关键思想是设置一个迭代形式

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

对于合适的 C 和 d ，以减少每一步的残差误差 $\|x^{(k+1)} - x^*\|$ 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。

< 上一章节

下一章节 >

2.2 矩阵

2.4 线性空间



2.4 线性空间

到目前为止，我们已经研究了线性方程组及其解法（第2.3节）。我们看到线性方程组可以用矩阵-向量表示法紧凑地表示为 $Ax = b$ 。接下来，我们将更深入地研究线性空间，即向量所在的结构化空间。在本章开头，我们非正式地将向量定义为可以相加并能被标量乘的数学对象，并且结果仍然是相同类型的对象。现在，我们准备正式化这一概念，并从群的概念开始，群是集合和在这些元素上定义的操作，它保持了集合的某种结构。

2.4.1 群

群在计算机科学中扮演着重要角色。除了为集合上的操作提供基本框架外，它们还广泛应用于密码学、编码理论和图形学。

定义2.7（群）

考虑一个集合 \mathcal{G} 和一个在 \mathcal{G} 上定义的二元运算 $\otimes : \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}$ 。那么 $G := (\mathcal{G}, \otimes)$ 被称为一个**群**，如果满足以下条件：

1. 关于 \otimes 的封闭性： $\forall x, y \in \mathcal{G} : x \otimes y \in \mathcal{G}$
2. 结合律： $\forall x, y, z \in \mathcal{G} : (x \otimes y) \otimes z = x \otimes (y \otimes z)$
3. 单位元： $\exists e \in \mathcal{G} \forall x \in \mathcal{G} : x \otimes e = x$ 且 $e \otimes x = x$
4. 逆元： $\forall x \in \mathcal{G} \exists y \in \mathcal{G} : x \otimes y = e$ 且 $y \otimes x = e$ ，其中 e 是单位元。

我们通常用 x^{-1} 表示 x 的逆元。

如果此外 $\forall x, y \in \mathcal{G} : x \otimes y = y \otimes x$ ，则 $G = (\mathcal{G}, \otimes)$ 是一个**Abel 群**（交换群）。

注释 逆元是相对于操作 \otimes 定义的，并不一定就是所谓的 $\frac{1}{x}$ 。



例2.10（群）

让我们来看一些具有相关操作的集合示例，并检查它们是否是群：

- $(\mathbb{Z}, +)$ ，即整数集 \mathbb{Z} 关于整数加法 $+$ 构成一个 **Abel 群**。
- $(\mathbb{N}_0, +)$ ，即含有 0 的自然数集关于自然数的加法 $+$ 不是一个群：尽管 $(\mathbb{N}_0, +)$ 具有单位元 $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ （即 0 ），但缺少逆元：例如我们找不到 1 的逆元（ -1 并不在自然数集中）。
- (\mathbb{Z}, \cdot) ，即整数集关于整数乘法不是一个群：尽管 (\mathbb{Z}, \cdot) 包含单位元（即 1 ），但除了 $z = \pm 1$ 外，其他整数缺少逆元。
- (\mathbb{R}, \cdot) ，即实数集关于实数乘法不是一个群，因为 0 没有逆元。
- $(\mathbb{R} \setminus \{0\}, \cdot)$ ，即去掉 0 的实数集关于实数乘法成为一个 **Abel 群**。
- $(\mathbb{R}^n, +)$ 、 $(\mathbb{Z}^n, +)$ ，其中 $n \in \mathbb{N}$ ，如果“ $+$ ”定义为对应分量相加，即

$$(x_1, \dots, x_n) + (y_1, \dots, y_n) = (x_1 + y_1, \dots, x_n + y_n) \quad (2.61)$$

则它们是 **Abel 群**。此时， $(x_1, \dots, x_n)^{-1} := (-x_1, \dots, -x_n)$ 是逆元，而 $e = (0, \dots, 0)$ 是单位元。

- $(\mathbb{R}^{m \times n}, +)$ 是 **Abel 群**（按分量加法定义，如(2.61)）。
- 让我们仔细分析 $n \times n$ 矩阵构成的集合和上面的矩阵乘法运算，也就是 $(\mathbb{R}^{n \times n}, \cdot)$ ：矩阵乘法的封闭性和结合性直接由定义得出。
 - 单位元：单位矩阵 I_n 是 $(\mathbb{R}^{n \times n}, \cdot)$ 中的单位元。
 - 逆元：如果逆存在（即 A 是可逆的），则 A^{-1} 是 $A \in \mathbb{R}^{n \times n}$ 的逆元。在这种情况下， $(\mathbb{R}^{n \times n}, \cdot)$ 是一个群，称为**一般线性群**（General Linear Group）。

（译者注：严格来说，我们在此滥用了符号 $+$ 和 \cdot 。但因为此例中出现的都是我们熟知的数学对象，读者可以轻易地推断出它是哪一种加法和乘法。）

定义2.8（一般线性群）

可逆矩阵 $A \in \mathbb{R}^{n \times n}$ 的集合关于矩阵乘法定义为(2.13)，称为**一般线性群**，记作 $GL(n, \mathbb{R})$ 。然而，由于矩阵乘法不是交换的，因此该群不是 **Abel 群**。



2.4.2 线性空间

在讨论群时，我们研究了集合 \mathcal{G} 和 \mathcal{G} 上的内操作。接下来，我们将考虑包含内操作“+”和外操作“.”（标量乘法）的集合。我们可以将内操作视为一种加法形式，而外操作则为一种缩放形式。注意，内操作和外操作与内积和外积无关。

定义2.9 (线性空间)

一个实值线性空间 V 是这些资料 $(V, \mathbb{R}, +, \cdot)$, V 上具有两个运算:

$$+: \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{V} \quad \text{加法} \quad (2.62)$$

$$\cdot: \mathbb{R} \times \mathcal{V} \rightarrow \mathcal{V} \quad \text{纯量乘法} \quad (2.63)$$

其中:

1. $(\mathcal{V}, +)$ 是 Abel 群。

2. 分配律:

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

3. 标量乘法的结合律: $\forall \lambda, \psi \in \mathbb{R}, x \in V : \lambda \cdot (\psi \cdot x) = (\lambda\psi) \cdot x$

4. 标量乘法单位元: $\forall x \in V : 1 \cdot x = x$

集合 V 中的元素 x 称为**向量**。集合 V 中的单位元称为**零向量**，记作 **0**（在 Euclid 空间中可以写为 $[0, \dots, 0]^\top$ ），而运算“+”称为**向量加法**。集合 \mathbb{R} 中的元素 λ 称为**标量**，而运算“.”称为**标量乘法**。注意，标量乘法与标量积不同，我们将在第3.2节中讨论。（译者注：在较早的资料中，常常将上面的数学对象为线性空间，而将有限维线性空间（Euclid 空间）称为向量空间。如今的资料中“向量空间”和“线性空间”的含义相同。）

注释



我们在此未定义“向量乘法” \mathbf{ab} , 其中 $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ 。理论上, 我们可以定义逐元素乘法(即 Hadamard 积), 使得 $\mathbf{c} = \mathbf{ab}$ 且 $c_j = a_j b_j$ 。这种“数组乘法”在许多编程语言中很常见, 但在标准矩阵乘法规则下, 它在数学上意义有限。通过将向量视为 $n \times 1$ 矩阵(我们通常这样做), 我们可以使用矩阵乘法定义(2.13)。然而, 向量的维度不匹配。只有以下向量乘法是定义的:

- $\mathbf{ab}^\top \in \mathbb{R}^{n \times n}$ (外积)
- $\mathbf{a}^\top \mathbf{b} \in \mathbb{R}$ (内积/标量积/点积)

例2.11 (线性空间)

让我们来看一些重要的例子:

- $V = \mathbb{R}^n$, 其中 $n \in \mathbb{N}$, 是线性空间, 其操作定义如下:
 - 加法: $\mathbf{x} + \mathbf{y} = (x_1, \dots, x_n) + (y_1, \dots, y_n) = (x_1 + y_1, \dots, x_n + y_n)$, 对于所有 $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$
 - 标量乘法: $\lambda \mathbf{x} = \lambda(x_1, \dots, x_n) = (\lambda x_1, \dots, \lambda x_n)$, 对于所有 $\lambda \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^n$
- $V = \mathbb{R}^{m \times n}$, 其中 $m, n \in \mathbb{N}$, 是线性空间, 其操作定义如下:
 - 加法: $\mathbf{A} + \mathbf{B} = \begin{bmatrix} a_{11} + b_{11} & \dots & a_{1n} + b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & \dots & a_{mn} + b_{mn} \end{bmatrix}$, 按元素定义, 对于所有 $\mathbf{A}, \mathbf{B} \in V$
 - 标量乘法: $\lambda \mathbf{A} = \begin{bmatrix} \lambda a_{11} & \dots & \lambda a_{1n} \\ \vdots & \ddots & \vdots \\ \lambda a_{m1} & \dots & \lambda a_{mn} \end{bmatrix}$, 这和第2.2节中的定义相同。
- $V = \mathbb{C}$, 具有复数的标准加法定义。

注释



在接下来的内容中，我们将用 V 表示线性空间 $(V, \mathbb{R}, +, \cdot)$ ，当“+”和“·”是标准向量加法和标量乘法时。此外，我们将用 $x \in V$ 表示 V 中的向量，以简化符号。

注释

线性空间 \mathbb{R}^n 、 $\mathbb{R}^{n \times 1}$ 和 $\mathbb{R}^{1 \times n}$ 仅在书写向量的方式上有所不同。在接下来的内容中，我们不会区分 \mathbb{R}^n 和 $\mathbb{R}^{n \times 1}$ ，这允许我们将 n -元组写为列向量：

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

这种表示法简化了线性空间操作的符号。然而，我们确实会区分 $\mathbb{R}^{n \times 1}$ 和 $\mathbb{R}^{1 \times n}$ （即行向量），以避免与矩阵乘法混淆。默认情况下，我们用 \mathbf{x} 表示列向量，而行向量则用 \mathbf{x}^\top 表示，即 \mathbf{x} 的转置。

2.4.3 向量子空间

接下来，我们引入向量子空间的概念。直观上，它们是包含在原始线性空间内的集合，具有这样的性质：当我们对子空间内的元素进行线性空间操作时，结果总是落在子空间中。在这个意义上，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章将展示如何使用向量子空间进行降维。

定义2.10（向量子空间）

设 $V = (V, +, \cdot)$ 是一个线性空间，且 $U \subseteq V$, $U \neq \emptyset$ 。那么 $U = (U, +, \cdot)$ 称为 V 的向量子空间（或线性子空间），如果 U 是一个线性空间，且线性空间操作“+”和“·”限制在 $U \times U$ 和 $\mathbb{R} \times U$ 上。我们用 $U \subseteq V$ 表示 V 的子空间 U 。

如果 $U \subseteq V$ 且 V 是一个线性空间，那么 U 自然地从 V 继承了许多性质，因为这些性质对所有 $x \in V$ 都成立，特别是对所有 $x \in U \subseteq V$ 也成立。这包括Abel群的性质、分配律、结合律和单位元。为了确定 $(U, +, \cdot)$ 是否是 V 的子空间，我们仍需要证明：

1. $U \neq \emptyset$, 特别是: $\mathbf{0} \in U$
2. U 的封闭性:
 - 关于外操作: $\forall \lambda \in \mathbb{R}, \forall x \in U : \lambda x \in U$
 - 关于内操作: $\forall x, y \in U : x + y \in U$

例2.12 (向量子空间)：让我们来看一些例子：

- 对于每一个线性空间 V , 平凡子空间是 V 本身和 $\{\mathbf{0}\}$ 。
- 只有图2.6中的D是 \mathbb{R}^2 的子空间（具有通常的内/外操作）。A和C不满足封闭性；B不包含 $\mathbf{0}$ 。
- 齐次线性方程组 $Ax = \mathbf{0}$ 的解集是 \mathbb{R}^n 的一个子空间。
- 非齐次线性方程组 $Ax = b$ ($b \neq \mathbf{0}$) 的解不是 \mathbb{R}^n 的一个子空间。
- 任意多个子空间的交集仍然是一个子空间。

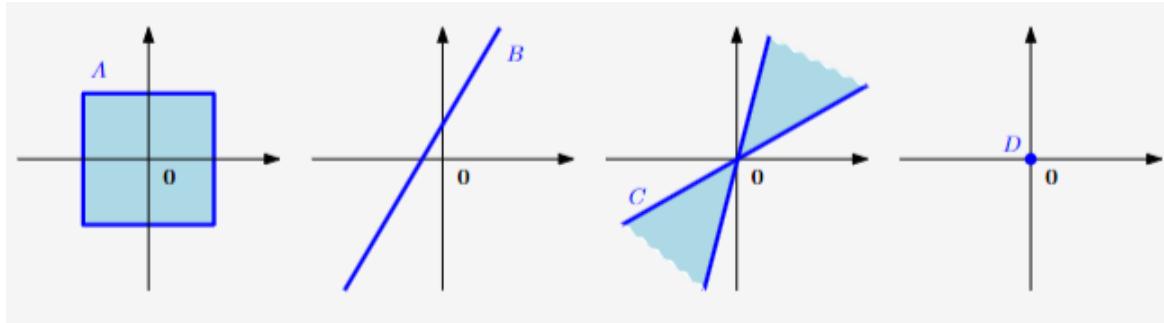


图2.6：并非 \mathbb{R}^2 的所有子集都是子空间。**A**和**C**不满足封闭性；**B**不包含 $\mathbf{0}$ 。只有**D**是子空间

注释：每一个子空间 $U \subseteq (\mathbb{R}^n, +, \cdot)$ 都是齐次线性方程组 $Ax = \mathbf{0}$ 的解空间，其中 $x \in \mathbb{R}^n$ 。♦

< 上一章节

下一章节 >

2.3 解线性方程组

2.5 线性无关





2.5 线性无关

接下来，我们将深入研究线性空间中的向量。特别是，我们可以将向量相加并用标量乘法进行缩放。线性空间的封闭性质保证了这样操作后得到的仍然是线性空间中的另一个向量。有可能找到一组向量，用它们可以表示线性空间中的每一个向量，通过将它们相加并进行缩放。这组向量称为**基**，我们将在第2.6节中讨论。在我们到达那里之前，我们需要引入线性组合和线性无关的概念。

2.5.1 线性组合

定义2.11（线性组合）：考虑一个线性空间 V 和有限个向量 $\mathbf{x}_1, \dots, \mathbf{x}_k \in V$ 。那么，每一个形如

$$\mathbf{v} = \lambda_1 \mathbf{x}_1 + \cdots + \lambda_k \mathbf{x}_k = \sum_{i=1}^k \lambda_i \mathbf{x}_i \in V \quad (2.65)$$

的向量 $\mathbf{v} \in V$ ，其中 $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ ，称为向量 $\mathbf{x}_1, \dots, \mathbf{x}_k$ 的**线性组合**。

零向量 $\mathbf{0}$ 总是可以表示为 k 个向量 $\mathbf{x}_1, \dots, \mathbf{x}_k$ 的线性组合，因为 $\mathbf{0} = \sum_{i=1}^k 0 \cdot \mathbf{x}_i$ 总是成立的。接下来，我们感兴趣的是向量集合的非平凡线性组合，即线性组合中的系数 λ_i 不全为零的情况。

2.5.2 线性无关

定义2.12（线性无关）：设 V 是一个线性空间，且 $\mathbf{x}_1, \dots, \mathbf{x}_k \in V$ 。如果存在一个非平凡的线性组合，使得

$$\mathbf{0} = \sum_{i=1}^k \lambda_i \mathbf{x}_i$$

其中至少有一个 $\lambda_i \neq 0$ ，则称向量 $\mathbf{x}_1, \dots, \mathbf{x}_k$ 是**线性相关的**。如果只有平凡解，即 $\lambda_1 = \cdots = \lambda_k = 0$ ，则称向量 $\mathbf{x}_1, \dots, \mathbf{x}_k$ 是**线性无关的**。



线性无关是线性代数中最重要的概念之一。直观上，一组线性无关的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，我们就会失去某些信息。在接下来的部分中，我们将更正式地形式化这种直觉。

例2.13（线性相关的向量）：一个地理学的例子可能有助于澄清线性无关的概念。一个在肯尼亚内罗毕的人描述卢旺达基加利的位置时可能会说：“你可以先向西北方向走506公里到乌干达的坎帕拉，然后向西南方向走374公里。”这足以描述基加利的位置，因为地理坐标系可以被视为一个二维线性空间（忽略海拔和地球的曲率）。这个人可能会补充说：“它大约在西边751公里处。”尽管这个说法是正确的，但鉴于前面的信息，它是不必要的（见图2.7的示意图）。在这个例子中，“向西北方向506公里”的向量（蓝色）和“向西南方向374公里”的向量（紫色）是线性无关的。这意味着西南方向的向量不能用西北方向的向量表示，反之亦然。然而，“向西751公里”的向量（黑色）是另外两个向量的线性组合，这使得这组向量是线性相关的。同样地，给定“向西751公里”和“向西南方向374公里”，可以线性组合得到“向西北方向506公里”。

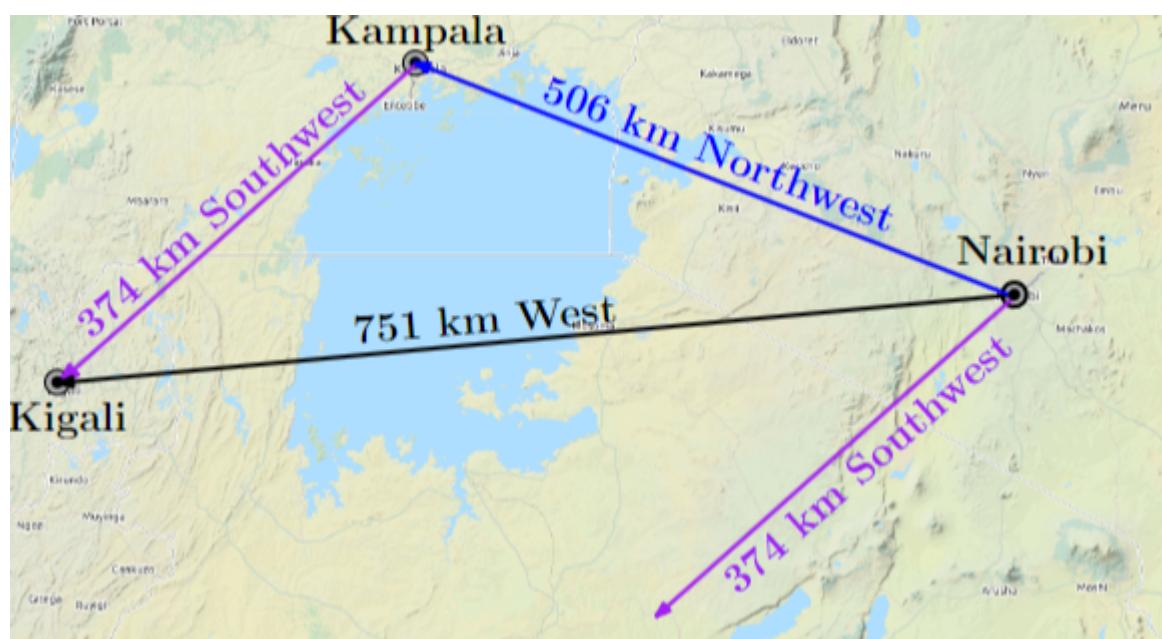


图2.7 一个近似的地理学的例子：二维空间（平面）中线性无关的向量

注释：以下性质有助于判断向量是否线性无关：

- k 个向量要么线性相关，要么线性无关。没有第三种可能。
- 如果向量 $\mathbf{x}_1, \dots, \mathbf{x}_k$ 中至少有一个是零向量，则它们是线性相关的。
- 如果两个向量相同，它们也是线性相关的。



- 对于向量集合 $\{\mathbf{x}_1, \dots, \mathbf{x}_k : \mathbf{x}_i \neq \mathbf{0}, i = 1, \dots, k\}$, 其中 $k \geq 2$, 它们是线性相关的当且仅当 (至少) 其中一个向量是其他向量的线性组合。特别是, 如果一个向量是另一个向量的倍数, 即 $\mathbf{x}_i = \lambda \mathbf{x}_j$, $\lambda \in \mathbb{R}$, 则集合 $\{\mathbf{x}_1, \dots, \mathbf{x}_k : \mathbf{x}_i \neq \mathbf{0}, i = 1, \dots, k\}$ 是线性相关的。

注释: 判断向量 $\mathbf{x}_1, \dots, \mathbf{x}_k \in V$ 是否线性无关的一个实用方法是使用高斯消元法: 将所有向量作为矩阵 A 的列向量, 并进行高斯消元法, 直到矩阵处于行阶梯形式 (简化行阶梯形式在这里是不必要的) :

- 主元列表示的向量是线性无关的。
- 非主元列可以表示为它们左边主元列的线性组合。例如, 行阶梯形式 $\begin{bmatrix} 1 & 3 & 0 & 0 & 2 \end{bmatrix} \tag{2.66}$ 表明第一列和第三列是主元列。第二列是非主元列, 因为它等于第一列的三倍。所有列向量线性无关当且仅当所有列都是主元列。如果至少有一个非主元列, 则列向量 (因此对应的向量) 是线性相关的。

例2.14: 考虑 \mathbb{R}^4 中的向量:

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

为了检查它们是否线性相关, 我们采用一般方法, 解方程

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

我们将向量 \mathbf{x}_i , $i = 1, 2, 3$, 作为矩阵的列, 并应用初等行变换, 直到确定主元列:

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



这里，矩阵的每一列都是主元列。因此，不存在非平凡解，我们要求 $\lambda_1 = 0$, $\lambda_2 = 0$, $\lambda_3 = 0$ 才能解这个方程组。因此，向量 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 是线性无关的。

注释：考虑一个线性空间 V 和其中的 k 个线性无关的向量 $\mathbf{b}_1, \dots, \mathbf{b}_k$ ，以及 m 个线性组合：

$$\mathbf{x}_1 = \sum_{i=1}^k \lambda_{i1} \mathbf{b}_i, \quad \dots, \quad \mathbf{x}_m = \sum_{i=1}^k \lambda_{im} \mathbf{b}_i. \quad (2.70)$$

定义 $B = [\mathbf{b}_1, \dots, \mathbf{b}_k]$ 为矩阵，其列向量是线性无关的向量 $\mathbf{b}_1, \dots, \mathbf{b}_k$ ，则可以更紧凑地表示为：

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

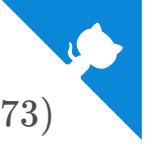
为了测试 $\mathbf{x}_1, \dots, \mathbf{x}_m$ 是否线性无关，我们采用一般方法，测试 $\sum_{j=1}^m \psi_j \mathbf{x}_j = \mathbf{0}$ 。根据(2.71)，我们有：

$$\sum_{j=1}^m \psi_j \mathbf{x}_j = \sum_{j=1}^m \psi_j B\boldsymbol{\lambda}_j = B \sum_{j=1}^m \psi_j \boldsymbol{\lambda}_j. \quad (2.72)$$

这意味着 $\{\mathbf{x}_1, \dots, \mathbf{x}_m\}$ 线性无关当且仅当列向量 $\{\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_m\}$ 线性无关。

注释：在线性空间 V 中， m 个线性组合的 k 个向量 $\mathbf{x}_1, \dots, \mathbf{x}_k$ 是线性相关的，如果 $m > k$ 。

例2.15：考虑一组线性无关的向量 $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4 \in \mathbb{R}^n$ 和



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

向量 $\mathbf{x}_1, \dots, \mathbf{x}_4 \in \mathbb{R}^n$ 是否线性无关？为了回答这个问题，我们需要检查列向量

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

是否线性无关。对应的线性方程组的系数矩阵为

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

其简化行阶梯形式为

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

我们看到对应的线性方程组是非平凡可解的：最后一列不是主元列，且 $\mathbf{x}_4 = -7\mathbf{x}_1 - 15\mathbf{x}_2 - 18\mathbf{x}_3$ 。因此， $\mathbf{x}_1, \dots, \mathbf{x}_4$ 是线性相关的，因为 \mathbf{x}_4 可以表示为 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 的线性组合。

< 上一章节

下一章节 >

2.4 线性空间

2.6 向量组的基与秩



2.6 向量组的基与秩

2.6.1 生成集和基

定义2.13 (生成集和张成空间) :

考虑一个线性空间 $V = (V, +, \cdot)$ 和一组向量 $A = \{\mathbf{x}_1, \dots, \mathbf{x}_k\} \subseteq V$ 。如果 V 中的每一个向量 \mathbf{v} 都可以表示为 $\mathbf{x}_1, \dots, \mathbf{x}_k$ 的线性组合，则称 A 是 V 的一个生成集。所有 A 中向量的线性组合的集合称为 A 的张成空间，记作 $\text{span}[A]$ 或 $\text{span}[\mathbf{x}_1, \dots, \mathbf{x}_k]$ 。如果 A 张成了线性空间 V ，我们写作 $V = \text{span}[A]$ 或 $V = \text{span}[\mathbf{x}_1, \dots, \mathbf{x}_k]$ 。

生成集是能够张成向量（子）空间的向量集合，即每一个向量都可以表示为生成集中向量的线性组合。接下来，我们将更具体地描述最小的生成集，它张成了一个向量（子）空间。

定义2.14 (基) : 考虑一个线性空间 $V = (V, +, \cdot)$ 和 $A \subseteq V$ 。如果一个生成集 A 是最小的，即不存在更小的集合 $\tilde{A} \subset A \subseteq V$ 能够张成 V ，则称 A 是 V 的一个基。每一个线性无关的生成集都是最小的，因此称为 V 的一个基。

设 $V = (V, +, \cdot)$ 是一个线性空间， $B \subseteq V$, $B \neq \emptyset$ 。那么，以下陈述是等价的：

- B 是 V 的一个基。
- B 是一个最小的生成集。
- B 是 V 中一个最大的线性无关向量集合，即在 B 中添加任何其他向量都将使其线性相关。
- V 中的每一个向量 \mathbf{x} 都可以表示为 B 中向量的线性组合，并且每一个线性组合都是唯一的，即

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

其中 $\mathbf{b}_1, \dots, \mathbf{b}_k \in \mathcal{B}$, $\lambda_i, \psi_i \in \mathbb{R}$, 我们立即有 $\lambda_i = \psi_i, i = 1, \dots, k$ 。

例2.16: 在 \mathbb{R}^3 中, 标准基 (或称为规范基) 是

$$B = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \quad (2.78)$$

\mathbb{R}^3 中不同的基包括:

$$B_1 = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}, \quad B_2 = \left\{ \begin{bmatrix} 0.5 \\ 0.8 \\ 0.4 \end{bmatrix}, \begin{bmatrix} 1.8 \\ 0.3 \\ 0.3 \end{bmatrix}, \begin{bmatrix} -2.2 \\ -1.3 \\ 3.5 \end{bmatrix} \right\} \quad (2.79)$$

集合

$$A = \left\{ \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ -1 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \\ -4 \end{bmatrix} \right\} \quad (2.80)$$

是线性无关的, 但不是 \mathbb{R}^4 的生成集 (也不是基), 因为例如向量 $[1, 0, 0, 0]^\top$ 无法通过 A 中的向量的线性组合来表示。

注释: 每一个线性空间 V 都有一个基 B 。前面的例子表明, 一个线性空间 V 可以有多个基, 即基不是唯一的。然而, 所有基都包含相同数量的元素, 即基向量 (译者注: 基中的向量称为基向量)。◆

我们只考虑有限维线性空间 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 的一个子空间, 则 $\dim(U) \leq \dim(V)$, 且 $\dim(U) = \dim(V)$ 当且仅当 $U = V$ 。直观上, 线性空间的维数可以被看作是该线性空间中独立方向的数量。线性空间的维数对应于其基向量的数量。



注释: 线性空间的维数并不一定是向量中的元素数量。例如, 线性空间

$$V = \text{span} \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$

是一维的, 尽管基向量包含两个元素。◆

注释: 一个子空间 $U = \text{span}[\mathbf{x}_1, \dots, \mathbf{x}_m] \subseteq \mathbb{R}^n$ 的基可以通过以下步骤找到:

1. 将张成向量作为矩阵 A 的列。
2. 确定 A 的行阶梯形式。
3. 与主元列对应的张成向量构成 U 的一个基。

例2.17 (确定基) : 对于一个向量子空间 $U \subseteq \mathbb{R}^5$, 由向量

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 2 \\ -1 \\ -1 \\ -1 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 2 \\ -1 \\ 1 \\ 2 \\ -2 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} 3 \\ -4 \\ 3 \\ 5 \\ -3 \end{bmatrix}, \quad \mathbf{x}_4 = \begin{bmatrix} -1 \\ 8 \\ -5 \\ -6 \\ 1 \end{bmatrix} \in \mathbb{R}^{(2,81)}$$

我们感兴趣的是找出 U 的一个基。为此, 我们需要检查 $\mathbf{x}_1, \dots, \mathbf{x}_4$ 是否线性无关。因此, 我们需要解方程

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

这导致一个齐次线性方程组, 其系数矩阵为

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

通过基本的线性方程组变换规则, 我们得到行阶梯形式:



$$\left[\begin{array}{cccc} 1 & 2 & 3 & -1 \\ 2 & -1 & -4 & 8 \\ -1 & 1 & 3 & -5 \\ -1 & 2 & 5 & -6 \\ -1 & -2 & -3 & 1 \end{array} \right] \Rightarrow \cdots \Rightarrow \left[\begin{array}{cccc} 1 & 2 & 3 & -1 \\ 0 & 1 & 2 & -2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right]. \quad (2.84)$$

由于主元列表明了哪些向量是线性无关的，我们从行阶梯形式中看到 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4$ 是线性无关的（因为方程组 $\lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 + \lambda_4 \mathbf{x}_4 = \mathbf{0}$ 只有平凡解 $\lambda_1 = \lambda_2 = \lambda_4 = 0$ ）。因此， $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4\}$ 是 U 的一个基。

2.6.2 秩

矩阵 $A \in \mathbb{R}^{m \times n}$ 的线性无关列的数量等于线性无关行的数量，称为 A 的秩，记作 $\text{rank}(A)$ 。

注释：秩具有一些重要的性质：

- $\text{rank}(A) = \text{rank}(A^\top)$ ，即列秩（线性无关列的数量）等于行秩（线性无关行的数量）。
- A 的列张成一个子空间 $U \subseteq \mathbb{R}^m$ ，其维数为 $\text{rank}(A)$ 。我们稍后将称这个子空间为像或值域。可以通过对 A 应用高斯消元法来找到 U 的一个基，以确定主元列。
- A 的行张成一个子空间 $W \subseteq \mathbb{R}^n$ ，其维数为 $\text{rank}(A)$ 。可以通过对 A^\top 应用高斯消元法来找到 W 的一个基。
- 对于所有 $A \in \mathbb{R}^{n \times n}$ ， A 是可逆的当且仅当 $\text{rank}(A) = n$ 。
- 对于所有 $A \in \mathbb{R}^{m \times n}$ 和所有 $\mathbf{b} \in \mathbb{R}^m$ ，线性方程组 $A\mathbf{x} = \mathbf{b}$ 有解当且仅当 $\text{rank}(A) = \text{rank}([A \mid \mathbf{b}])$ ，其中 $[A \mid \mathbf{b}]$ 表示增广矩阵。
- 对于 $A \in \mathbb{R}^{m \times n}$ ，齐次方程组 $A\mathbf{x} = \mathbf{0}$ 的解空间的维数为 $n - \text{rank}(A)$ 。我们稍后将称这个子空间为核或零空间。核的维数为 $n - \text{rank}(A)$ 。
- 一个矩阵 $A \in \mathbb{R}^{m \times n}$ 具有满秩，如果其秩等于对于相同维度的矩阵可能的最大秩。这意味着满秩矩阵的秩是行数和列数中较小的那个，即 $\text{rank}(A) = \min(m, n)$ 。如果一个矩阵的秩不等于满秩，则称该矩阵是秩亏的。



例2.18 (秩) :

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

A 有两行线性无关的行/列，因此 $\text{rk}(A) = 2$ 。

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

我们使用高斯消元法来确定秩：

$$\begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix} \Rightarrow \dots \Rightarrow \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 0 \end{bmatrix}. \quad (2.84)$$

这里，线性无关的行和列的数量为2，因此 $\text{rank}(A) = 2$ 。

< 上一章节

2.5 线性无关

下一章节 >

2.7 线性映射



2.7 线性映射

接下来，我们将研究线性空间之间的映射，这些映射保持线性空间的结构，从而可以定义坐标的概念。在本章开头，我们提到向量是可以相加并被标量乘的数学对象，结果仍然是向量。我们希望这种性质在应用映射时得以保持：考虑两个实线性空间 V, W ，一个映射 $\Phi : V \rightarrow W$ 保持线性空间的结构，如果

$$\Phi(\mathbf{x} + \mathbf{y}) = \Phi(\mathbf{x}) + \Phi(\mathbf{y}) \quad (2.85)$$

$$\Phi(\lambda \mathbf{x}) = \lambda \Phi(\mathbf{x}) \quad (2.86)$$

对所有 $\mathbf{x}, \mathbf{y} \in V$ 和 $\lambda \in \mathbb{R}$ 成立。我们可以将这些条件总结为以下定义：

定义2.15（线性映射）：对于线性空间 V, W ，一个映射 $\Phi : V \rightarrow W$ 称为**线性映射**（或**线性空间同态/线性变换**），如果

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

事实证明，我们可以用矩阵来表示线性映射（第2.7.1节）。回想一下，我们也可以将一组向量收集为矩阵的列。在处理矩阵时，我们需要牢记矩阵所代表的内容：一个线性映射还是一组向量。我们将在第4章中更多地了解线性映射。在继续之前，我们将简要介绍特殊的映射。

定义2.16（单射、满射、双射）：考虑一个映射 $\Phi : V \rightarrow W$ ，其中 V, W 可以是任意集合。那么 Φ 称为：

- **单射** (*Injective*)：如果 $\forall \mathbf{x}, \mathbf{y} \in V : \Phi(\mathbf{x}) = \Phi(\mathbf{y}) \Rightarrow \mathbf{x} = \mathbf{y}$ 。
- **满射** (*Surjective*)：如果 $\Phi(V) = W$ 。
- **双射** (*Bijective*)：如果它是单射且满射。

如果 Φ 是满射，那么 W 中的每一个元素都可以通过 Φ 从 V 中的某个元素“到达”。如果 Φ 是双射，那么存在一个映射 $\Psi : W \rightarrow V$ ，使得 $\Psi \circ \Phi(\mathbf{x}) = \mathbf{x}$ 。这个映射 Ψ 被称为 Φ 的**逆映射**，通常记作 Φ^{-1} 。根据这些定义，我们引入以下线性映射之间的特殊情形：

- **同构** (*Isomorphism*)： $\Phi : V \rightarrow W$ 线性且双射。



- **自同态** (Endomorphism) : $\Phi : V \rightarrow V$ 线性。线性空间 V 的所有自同态形成一个集合, 记为 $\text{End}(V)$
- **自同构** (Automorphism) : $\Phi : V \rightarrow V$ 线性且双射。线性空间 V 的所有自同构形成一个集合, 记为 $\text{Aut}(V)$
- **恒等映射** (Identity Automorphism) : $\text{id}_V : V \rightarrow V, \mathbf{x} \mapsto \mathbf{x}$ 是 V 中的恒等映射。

例2.19 (同态) : 映射 $\Phi : \mathbb{R}^2 \rightarrow \mathbb{C}$, $\Phi(\mathbf{x}) = x_1 + ix_2$ 是一个同态:

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

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 中的元素逐元加法转换为复数集中的相应加法。注意, 我们只证明了线性, 而不是双射。

定理2.17 (Axler, 2015, 定理3.59) : 有限维线性空间 V 和 W 是同构的, 当且仅当 $\dim(V) = \dim(W)$ 。

定理2.17表明, 如果两个线性空间的维数相同, 则它们之间存在一个线性双射映射。直观上, 这意味着维数相同的线性空间在某种意义上是相同的, 因为它们可以相互转换而不会有任何损失。定理2.17还为我们将 $\mathbb{R}^{m \times n}$ ($m \times n$ 矩阵的线性空间) 和 \mathbb{R}^{mn} (长度为 mn 的线性空间) 视为相同提供了依据, 因为它们的维数都是 mn , 并且存在一个线性双射映射, 可以将一个转换为另一个。

注释: 考虑线性空间 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, 那么 $\Phi^{-1} : W \rightarrow V$ 也是一个同构。
- 如果 $\Phi : V \rightarrow W$ 和 $\Psi : V \rightarrow W$ 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 维线性空间都与 \mathbb{R}^n 同构（定理2.17）。我们考虑一个 n 维线性空间 V 的一个基 $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ 。接下来，有序基的顺序将很重要。因此，我们写作

$$B = (\mathbf{b}_1, \dots, \mathbf{b}_n) \quad (2.89)$$

并称这个 n -元组为 V 的一个**有序基**。

注（一些记号）：现在我们有多个看起来相似易混淆的记号。因此我们在此重申：

- $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ 是一个有序基；
- $B = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ 是一个（无序）基；
- $B = [\mathbf{b}_1, \dots, \mathbf{b}_n]$ 是一个矩阵，其列向量是向量 $\mathbf{b}_1, \dots, \mathbf{b}_n$ 。◆

定义2.18（坐标）：考虑一个线性空间 V 和 V 的一个有序基 $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ 。对于任意 $\mathbf{x} \in V$ ，我们得到一个唯一的表示（线性组合）

$$\mathbf{x} = \alpha_1 \mathbf{b}_1 + \cdots + \alpha_n \mathbf{b}_n \quad (2.90)$$

的 \mathbf{x} 关于 B 。那么 $\alpha_1, \dots, \alpha_n$ 称为 \mathbf{x} 关于 B 的**坐标**，而向量

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

称为 \mathbf{x} 关于有序基 B 的**坐标向量或坐标表示**。

基有效地定义了一个坐标系。我们熟悉二维中的笛卡尔坐标系，它由规范基向量 $\mathbf{e}_1, \mathbf{e}_2$ 张成。在这个坐标系中，一个向量 $\mathbf{x} \in \mathbb{R}^2$ 有一个表示，它告诉我们如何线性组合 \mathbf{e}_1 和 \mathbf{e}_2 来得到 \mathbf{x} 。然而， \mathbb{R}^2 的任何基都定义了一个有效的坐标系，同一个向量 \mathbf{x} 在 $(\mathbf{b}_1, \mathbf{b}_2)$ 基中可能有不同的坐标表示。在图2.8中， \mathbf{x} 关于标准基 $(\mathbf{e}_1, \mathbf{e}_2)$ 的坐标是 $[2, 2]^\top$ 。然而，关于基 $(\mathbf{b}_1, \mathbf{b}_2)$ ，同一个向量 \mathbf{x} 被表示为 $[1.09, 0.72]^\top$ ，即 $\mathbf{x} = 1.09\mathbf{b}_1 + 0.72\mathbf{b}_2$ 。在接下来的部分中，我们将发现如何得到这种表示。

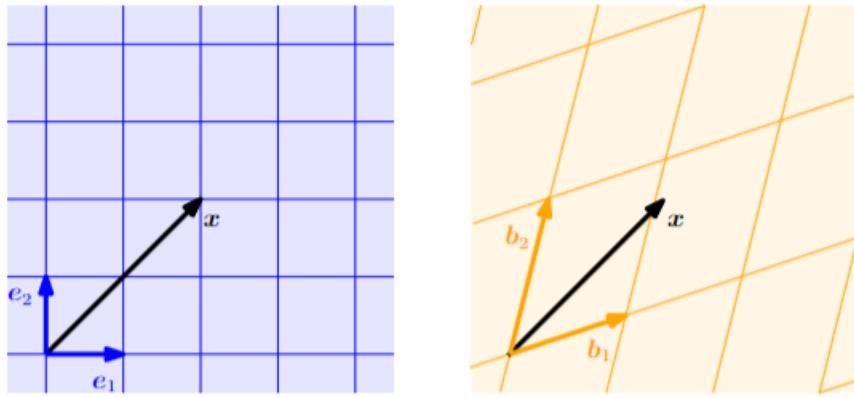


图2.8 相同的向量在基不同的两个线性空间中的坐标表示不同

例2.20: 考虑一个几何向量 $\mathbf{x} \in \mathbb{R}^2$, 其关于 \mathbb{R}^2 的标准基 $(\mathbf{e}_1, \mathbf{e}_2)$ 的坐标为 $[2, 3]^\top$ 。这意味着我们可以写作 $\mathbf{x} = 2\mathbf{e}_1 + 3\mathbf{e}_2$ 。然而, 我们不需要选择标准基来表示这个向量。如果我们使用基向量 $\mathbf{b}_1 = [1, -1]^\top, \mathbf{b}_2 = [1, 1]^\top$, 我们将得到坐标 $\frac{1}{2}[-1, 5]^\top$, 以表示关于 $(\mathbf{b}_1, \mathbf{b}_2)$ 的同一个向量 (见图2.9)。

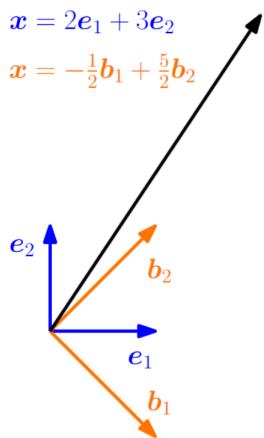


图2.9 同一向量的不同坐标表示取决于基的选取

注释: 对于一个 n 维线性空间 V 和 V 的一个有序基 B , 映射 $\Phi : \mathbb{R}^n \rightarrow V$, $\Phi(\mathbf{e}_i) = \mathbf{b}_i, i = 1, \dots, n$, 是线性的 (并且由于定理2.17是一个同构), 其中 $(\mathbf{e}_1, \dots, \mathbf{e}_n)$ 是 \mathbb{R}^n 的标准基。♦

现在, 我们已经准备好明确地建立矩阵和有限维线性空间之间的线性映射之间的联系。



定义2.19 (变换矩阵) : 考虑线性空间 V, W , 分别有对应的 (有序) 基 $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ 和 $C = (\mathbf{c}_1, \dots, \mathbf{c}_m)$ 。此外, 考虑一个线性映射 $\Phi : V \rightarrow W$ 。对于 $j \in \{1, \dots, n\}$,

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

是 $\Phi(\mathbf{b}_j)$ 关于 C 的唯一表示。那么, 我们称矩阵 A_Φ , 其元素由

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

给出, 为 Φ 的**变换矩阵** (关于有序基 B 和 C)。

$\Phi(\mathbf{b}_j)$ 关于有序基 C 的坐标是变换矩阵 A_Φ 的第 j 列。考虑 (有限维) 线性空间 V, W , 分别有有序基 B, C , 以及一个线性映射 $\Phi : V \rightarrow W$ 和其变换矩阵 A_Φ 。如果 $\hat{\mathbf{x}}$ 是 $\mathbf{x} \in V$ 关于 B 的坐标向量, 而 $\hat{\mathbf{y}}$ 是 $\mathbf{y} = \Phi(\mathbf{x}) \in W$ 关于 C 的坐标向量, 那么

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

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) : 考虑一个同态 $\Phi : V \rightarrow W$ 和 V 的有序基 $B = (\mathbf{b}_1, \dots, \mathbf{b}_3)$ 以及 W 的有序基 $C = (\mathbf{c}_1, \dots, \mathbf{c}_4)$ 。给定

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

那么关于 B 和 C 的变换矩阵 A_Φ 满足 $\Phi(\mathbf{b}_k) = \sum_{i=1}^4 \alpha_{ik}\mathbf{c}_i$, $k = 1, \dots, 3$, 并且由

$$A_\Phi = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3] = \begin{bmatrix} 1 & 2 & 0 \\ -1 & 1 & 3 \\ 3 & 7 & 1 \\ -1 & 2 & 4 \end{bmatrix}, \quad (2.96)$$

给出, 其中 $\boldsymbol{\alpha}_j$, $j = 1, 2, 3$, 是 $\Phi(\mathbf{b}_j)$ 关于 C 的坐标向量。

例2.22 (向量的线性变换) : 考虑 \mathbb{R}^2 中的一组向量, 如图2.10(a)所示, 每个向量用一个点表示, 对应于 (x_1, x_2) -坐标。这些向量排列成一个正方形。当我们使用矩阵

A_1 在(2.97)中对每个向量进行线性变换时，我们得到图2.10(b)中的旋转正方形。如果我们将应用由矩阵 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 -坐标被拉伸了2倍。图2.10(d)显示了使用 A_3 对原始正方形进行线性变换后的结果，这是一个反射、旋转和拉伸的组合。

$$A_1 = \begin{bmatrix} \cos\left(\frac{\pi}{4}\right) & -\sin\left(\frac{\pi}{4}\right) \\ \sin\left(\frac{\pi}{4}\right) & \cos\left(\frac{\pi}{4}\right) \end{bmatrix}, \quad A_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \quad A_3 = \frac{1}{2} \begin{bmatrix} 3 & -1 \\ 1 & 1 \end{bmatrix}$$

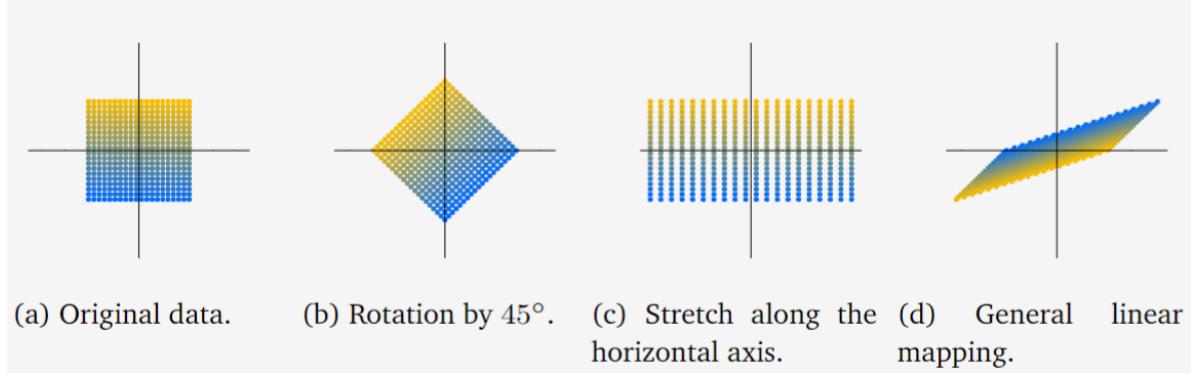


图2.10 线性映射的例子。 (a) 以点表示的均匀分布的原始向量。 (b) 逆时针旋转 45° (c) 沿着水平方向两侧拉长 (d) 一般的线性变换

2.7.2 基变换

接下来，我们将更仔细地研究当我们在 V 和 W 中改变基时，线性映射 $\Phi : V \rightarrow W$ 的变换矩阵如何变化。考虑 V 的两个有序基

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

以及 W 的两个有序基

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

此外， $A_\Phi \in \mathbb{R}^{m \times n}$ 是线性映射 $\Phi : V \rightarrow W$ 关于基 B 和 C 的变换矩阵，而 $\tilde{A}_\Phi \in \mathbb{R}^{m \times n}$ 是关于 \tilde{B} 和 \tilde{C} 的相应变换矩阵。在接下来的内容中，如果我们选择从 B, C 到 \tilde{B}, \tilde{C} 进行基变换，我们将研究 A 和 \tilde{A} 之间的关系（我们是否可以将 A_Φ 转换为 \tilde{A}_Φ ）。

注释： 我们实际上得到了不同坐标表示的恒等映射 id_V 。在图2.9的上下文中，这意味着将关于 $(\mathbf{e}_1, \mathbf{e}_2)$ 的坐标映射到关于 $(\mathbf{b}_1, \mathbf{b}_2)$ 的坐标，而不改变

向量 \mathbf{x} 。通过改变基并相应地改变向量的表示，关于新基的变换矩阵可以具有特别简单的形式，这使得计算变得直接。◆

例2.23 (基变换)：考虑一个变换矩阵

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (2.100)$$

关于 \mathbb{R}^2 的规范基。如果我们定义一个新的基

$$B = \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right) \quad (2.101)$$

那么我们得到一个基 B 下的矩阵

$$\tilde{A} = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.102)$$

相对于 A ，它是对角矩阵更容易处理。

在接下来的内容中，我们将研究将一个基的坐标向量映射到另一个基的坐标向量的映射。我们将首先陈述我们的主要结果，然后进行解释。

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

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

在 V 中，

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

在 W 中，以及关于 B 和 C 的变换矩阵 A_Φ ，关于基 \tilde{B} 和 \tilde{C} 的相应变换矩阵 \tilde{A}_Φ 由

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

给出。这里， $S \in \mathbb{R}^{n \times n}$ 是恒等映射 id_V 的变换矩阵，它将关于 \tilde{B} 的坐标映射到关于 B 的坐标，而 $T \in \mathbb{R}^{m \times m}$ 是恒等映射 id_W 的变换矩阵，它将关于 \tilde{C} 的坐标映射到关于 C 的坐标。

证明：根据 Drumm 和 Weil (2001)，我们可以将 V 的新基 \tilde{B} 的向量表示为 B 的基向量的线性组合，使得

$$\tilde{\mathbf{b}}_j = \sum_{i=1}^n s_{ij} \mathbf{b}_i, \quad j = 1, \dots, n. \quad (2.106)$$

类似地，我们将 W 的新基 \tilde{C} 的向量表示为 C 的基向量的线性组合，得到

$$\tilde{\mathbf{c}}_k = \sum_{l=1}^m t_{lk} \mathbf{c}_l, \quad k = 1, \dots, m. \quad (2.107)$$

我们定义 $S = ((s_{ij})) \in \mathbb{R}^{n \times n}$ 为变换矩阵，它将关于 \tilde{B} 的坐标映射到关于 B 的坐标，而 $T = ((t_{lk})) \in \mathbb{R}^{m \times m}$ 为变换矩阵，它将关于 \tilde{C} 的坐标映射到关于 C 的坐标。特别地， S 的第 j 列是 $\tilde{\mathbf{b}}_j$ 关于 B 的坐标表示，而 T 的第 k 列是 $\tilde{\mathbf{c}}_k$ 关于 C 的坐标表示。注意， S 和 T 都是可逆的。我们将从两个角度研究 $\Phi(\tilde{\mathbf{b}}_j)$ 。首先，应用映射 Φ ，我们得到

$$\Phi(\tilde{\mathbf{b}}_j) = \sum_{k=1}^m \underbrace{\tilde{k}_j}_{\in W} \tilde{\mathbf{c}}_k \xrightarrow{(2.107)} \sum_{k=1}^m \tilde{k}_j \sum_{l=1}^m t_{lk} \mathbf{c}_l = \sum_{l=1}^m \left(\sum_{k=1}^m t_{lk} \tilde{k}_j \right) \mathbf{c}_l, \quad (2.108)$$

其中我们首先将 W 中的新基向量 $\tilde{\mathbf{c}}_k$ 表示为 C 中基向量 \mathbf{c}_l 的线性组合，然后交换了求和的顺序。另一方面，将 $\tilde{\mathbf{b}}_j \in V$ 表示为 B 中基向量 $\mathbf{b}_i \in V$ 的线性组合，我们得到

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

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

其中我们利用了 Φ 的线性。比较 (2.108) 和 (2.109b)，我们得到

$$\sum_{k=1}^m t_{lk} \tilde{k}_j = \sum_{i=1}^n a_{li} s_{ij}, \quad \forall j = 1, \dots, n, \quad \forall l = 1, \dots, m,$$

因此，

$$T \tilde{A}_\Phi = A_\Phi S \in \mathbb{R}^{m \times n}, \quad (2.111)$$

从而，

$$\tilde{A}_\Phi = T^{-1} A_\Phi S,$$

(2.112)

这就证明了定理2.20。

定理2.20告诉我们，当我们在 V 中从 B 到 \tilde{B} 进行基变换，以及在 W 中从 C 到 \tilde{C} 进行基变换时，线性映射 $\Phi : V \rightarrow W$ 的变换矩阵 A_Φ 被替换为等价矩阵 \tilde{A}_Φ ，满足

$$\tilde{A}_\Phi = T^{-1} A_\Phi S. \quad (2.113)$$

图2.11说明了这种关系：考虑一个同态 $\Phi : V \rightarrow W$ 和 V 的有序基 B, \tilde{B} 以及 W 的有序基 C, \tilde{C} 。映射 Φ_{CB} 是 Φ 的一个实例，它将 B 的基向量映射为 C 的基向量的线性组合。假设我们知道关于有序基 B, C 的变换矩阵 A_Φ 。当我们从 B 到 \tilde{B} 在 V 中进行基变换，以及从 C 到 \tilde{C} 在 W 中进行基变换时，我们可以通过以下方式确定关于基 \tilde{B}, \tilde{C} 的映射 $\Phi_{\tilde{C}\tilde{B}}$ ：

$$\Phi_{\tilde{C}\tilde{B}} = \Xi_{\tilde{C}\tilde{C}} \circ \Phi_{CB} \circ \Psi_{B\tilde{B}} = \Xi_{\tilde{C}\tilde{C}}^{-1} \circ \Phi_{CB} \circ \Psi_{B\tilde{B}}. \quad (2.114)$$

具体来说，我们使用 $\Psi_{B\tilde{B}} = \text{id}_V$ 和 $\Xi_{\tilde{C}\tilde{C}} = \text{id}_W$ ，即 V 和 W 中的恒等映射。

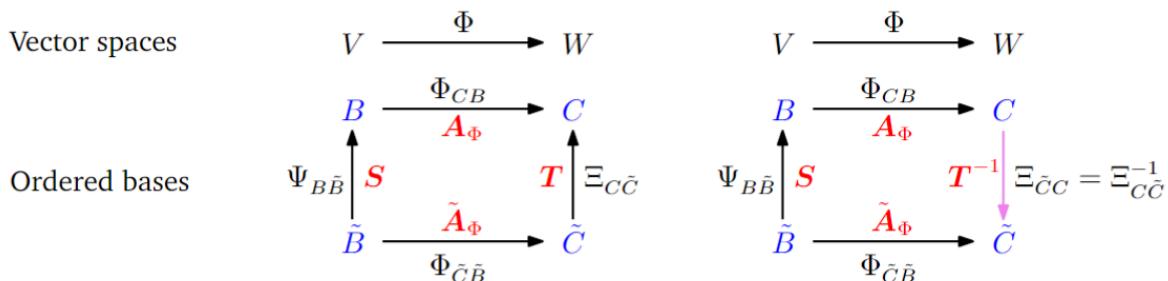


图2.11 线性变换和基变换之间的关系

定义2.21 (等价) : 如果存在可逆矩阵 $S \in \mathbb{R}^{n \times n}$ 和 $T \in \mathbb{R}^{m \times m}$ ，使得

$$\tilde{A} = T^{-1} AS,$$

则称矩阵 $A, \tilde{A} \in \mathbb{R}^{m \times n}$ 是等价的。

定义2.22 (相似) : 如果存在一个可逆矩阵 $S \in \mathbb{R}^{n \times n}$ ，使得

$$\tilde{A} = S^{-1} AS,$$

则称矩阵 $A, \tilde{A} \in \mathbb{R}^{n \times n}$ 是相似的。

注释: 相似矩阵总是等价的。然而, 等价矩阵不一定是相似的。◆

注释: 考虑线性空间 V, W, X 。从注释中我们知道, 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。如果 A_Φ 和 A_Ψ 是相应映射的变换矩阵, 那么整体变换矩阵是 $A_{\Psi \circ \Phi} = A_\Psi A_\Phi$ 。◆

从这个注释来看, 我们可以从线性映射的组合的角度来看待基变换:

- A_Φ 是线性映射 $\Phi_{CB} : V \rightarrow W$ 关于基 B, C 的变换矩阵。
- \tilde{A}_Φ 是线性映射 $\Phi_{\tilde{C}\tilde{B}} : V \rightarrow W$ 关于基 \tilde{B}, \tilde{C} 的变换矩阵。
- S 是线性映射 $\Psi_{B\tilde{B}} : V \rightarrow V$ (自同构) 的变换矩阵, 它用 B 表示 \tilde{B} 。通常, $\Psi = \text{id}_V$ 是 V 中的恒等映射。
- T 是线性映射 $\Xi_{C\tilde{C}} : W \rightarrow W$ (自同构) 的变换矩阵, 它用 C 表示 \tilde{C} 。通常, $\Xi = \text{id}_W$ 是 W 中的恒等映射。

如果非正式地用基来表示这些变换, 那么 $A_\Phi : B \rightarrow C$, $\tilde{A}_\Phi : \tilde{B} \rightarrow \tilde{C}$, $S : \tilde{B} \rightarrow B$, $T : \tilde{C} \rightarrow C$, 以及 $T^{-1} : C \rightarrow \tilde{C}$, 并且

$$\tilde{B} \rightarrow \tilde{C} = \tilde{B} \rightarrow B \rightarrow C \rightarrow \tilde{C} \quad (2.115)$$

$$\tilde{A}_\Phi = T^{-1} A_\Phi S. \quad (2.116)$$

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 $x \mapsto Sx \mapsto A_\Phi(Sx) \mapsto T^{-1}(A_\Phi(Sx)) = \tilde{A}_\Phi x$ 。

例2.24 (基变换)

考虑一个线性映射 $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^4$, 其变换矩阵为

$$A_\Phi = \begin{bmatrix} 1 & 2 & 0 \\ -1 & 1 & 3 \\ 3 & 7 & 1 \\ -1 & 2 & 4 \end{bmatrix} \quad (2.117)$$

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

我们希望找到关于新基

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

的变换矩阵 \tilde{A}_Φ 。那么，

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

其中 S 的第 i 列是 $\tilde{\mathbf{b}}_i$ 关于 B 的基向量的坐标表示。由于 B 是标准基，坐标表示很容易找到。对于一般的基 B ，我们需要解一个线性方程组来找到 λ_i ，使得

$$\sum_{i=1}^3 \lambda_i \mathbf{b}_i = \tilde{\mathbf{b}}_j, \quad j = 1, \dots, 3.$$

类似地， T 的第 j 列是 $\tilde{\mathbf{c}}_j$ 关于 C 的基向量的坐标表示。因此，我们得到

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

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

在第4章中，我们将利用基变换的概念，找到一个基，使得一个自同态矩阵具有特别简单的（对角）形式。在第10章中，我们将研究一个数据压缩问题，并找到一个方便的基，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子空间。接下来，我们将更仔细地描述它们。

定义2.23（像和核）：对于 $\Phi : V \rightarrow W$ ，我们定义

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

为 Φ 的**核 (kernel)或零空间 (null space)**，以及

$$\text{Im}(\Phi) := \Phi(V) = \{\mathbf{w} \in W : \exists \mathbf{v} \in V, \Phi(\mathbf{v}) = \mathbf{w}\} \quad (2.123)$$

为 Φ 的 **像 (image)** 或 **值域 (range, co-domain)**。

我们还称 V 和 W 分别为 Φ 的**定义域**和**值域**。直观上，核是 V 中所有被 Φ 映射到 W 中的零向量 $\mathbf{0}_W$ 的向量集合。像则是 W 中所有可以通过 Φ 从 V 中的某个向量“到达”的向量集合。图2.12给出了一个说明。

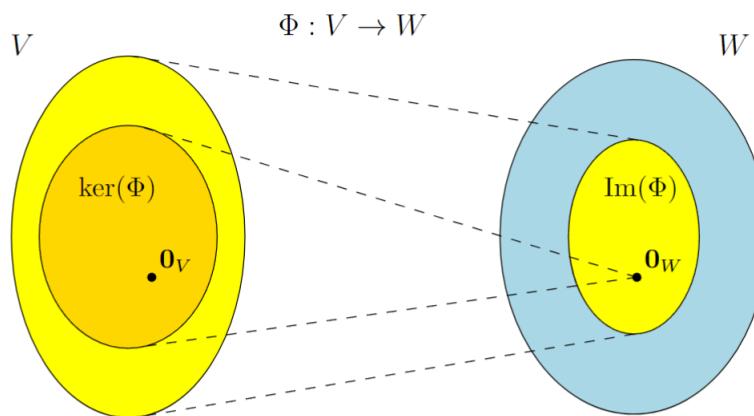


图2.12 线性映射的像和核

注释：考虑线性映射 $\Phi : V \rightarrow W$ ，其中 V, W 是线性空间。

- $\Phi(\mathbf{0}_V) = \mathbf{0}_W$ ，因此 $\mathbf{0}_V \in \ker(\Phi)$ 总是成立。特别地，零空间永远不会是空的。

- $\text{Im}(\Phi) \subseteq W$ 是 W 的一个子空间，而 $\ker(\Phi) \subseteq V$ 是 V 的一个子空间。

注释（零空间和列空间）：考虑 $A \in \mathbb{R}^{m \times n}$ 和线性映射 $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $\mathbf{x} \mapsto A\mathbf{x}$ 。对于 $A = [\mathbf{a}_1, \dots, \mathbf{a}_n]$, 其中 \mathbf{a}_i 是 A 的列，我们得到

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

$$= \text{span}[\mathbf{a}_1, \dots, \mathbf{a}_n] \subseteq \mathbb{R}^m, \quad (2.124b)$$

即像就是 A 的列的张成空间，也称为**列空间**。因此，像（列空间）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 $A\mathbf{x} = \mathbf{0}$ 的一般解，它捕捉了所有可能的线性组合，这些线性组合在 \mathbb{R}^n 中产生 $\mathbf{0} \in \mathbb{R}^m$ 。零空间是 \mathbb{R}^n 的一个子空间，其中 n 是矩阵的“宽度”。零空间关注列之间的关系，我们可以用它来确定如何/是否可以用其他列来表示一列。

例2.25（线性映射的像和核）：映射

$$\begin{aligned} \Phi : \mathbb{R}^4 &\rightarrow \mathbb{R}^2, \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \mapsto \begin{bmatrix} 1 & 2 & -1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 + 2x_2 \\ x_1 + x_4 \end{bmatrix} \\ &= x_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + x_2 \begin{bmatrix} 2 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} -1 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned} \quad (2.125b)$$

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

$$\text{Im}(\Phi) = \text{span} \left[\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right]. \quad (2.126)$$

为了计算 Φ 的核（零空间），我们需要解 $A\mathbf{x} = \mathbf{0}$ ，即我们需要解一个齐次方程组。为此，我们使用高斯消元法将 A 转换为简化行阶梯形式：

$$\begin{bmatrix} 1 & 2 & -1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \cdots \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix}. \quad (2.127)$$

这个矩阵处于简化行阶梯形式，我们可以使用“负一技巧”来计算零空间的一个基（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3列 \mathbf{a}_3 等于 $-\frac{1}{2}$ 倍的第2列 \mathbf{a}_2 。因此, $\mathbf{0} = \mathbf{a}_3 + \frac{1}{2}\mathbf{a}_2$ 。同样地, 我们看到 $\mathbf{a}_4 = \mathbf{a}_1 - \frac{1}{2}\mathbf{a}_2$, 因此 $\mathbf{0} = \mathbf{a}_1 - \frac{1}{2}\mathbf{a}_2 - \mathbf{a}_4$ 。总体上, 这给出了零空间 (核) 为

$$\ker(\Phi) = \text{span} \left[\begin{bmatrix} 0 \\ \frac{1}{2} \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ \frac{1}{2} \\ 0 \\ 1 \end{bmatrix} \right]. \quad (2.128)$$

秩-零度定理: 对于线性空间 V, W 和线性映射 $\Phi : V \rightarrow W$, 我们有

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

秩-零度定理也被称为线性映射的基本定理 (Axler, 2015, 定理3.22)。以下是一些直接的推论:

- 如果 $\dim(\text{Im}(\Phi)) < \dim(V)$, 那么 $\ker(\Phi)$ 是非平凡的, 即核包含的不仅仅是 $\mathbf{0}_V$, 且 $\dim(\ker(\Phi)) \geq 1$ 。
- 如果 A_Φ 是关于有序基的 Φ 的变换矩阵且 $\dim(\text{Im}(\Phi)) < \dim(V)$, 那么线性方程组 $A_\Phi \mathbf{x} = \mathbf{0}$ 有无穷多个解。
- 如果 $\dim(V) = \dim(W)$, 那么以下三个条件是等价的:
 - Φ 是单射。
 - Φ 是满射。
 - Φ 是双射, 因为 $\text{Im}(\Phi) \subseteq W$ 。

< 上一章节

下一章节 >

2.6 向量组的基与秩

2.8 仿射空间



2.8 仿射空间

2.8 仿射空间

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

注释：在机器学习文献中，线性和仿射之间的区别有时并不清楚，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。♦

2.8.1 仿射子空间

定义2.25（仿射子空间）：设 V 是一个线性空间， $\mathbf{x}_0 \in V$ 且 $U \subseteq V$ 是一个子空间。那么子集

$$L = \mathbf{x}_0 + U := \{\mathbf{x}_0 + \mathbf{u} : \mathbf{u} \in U\} \quad (2.130a)$$

$$= \{\mathbf{v} \in V : \exists \mathbf{u} \in U, \mathbf{v} = \mathbf{x}_0 + \mathbf{u}\} \subseteq V \quad (2.130b)$$

称为 V 的一个仿射子空间或线性流形。 U 称为方向或仿射子空间的方向空间，而 \mathbf{x}_0 称为支点。在第12章中，我们称这样的子空间为超平面。

注意，仿射子空间的定义排除了 $\mathbf{0}$ ，如果 $\mathbf{x}_0 \notin U$ 。因此，如果 $\mathbf{x}_0 \notin U$ ，仿射子空间不是 V 的一个（线性）子空间。仿射子空间的例子是 \mathbb{R}^3 中的点、线和平面，它们不一定通过原点。

注释：考虑 V 的两个仿射子空间 $L = \mathbf{x}_0 + U$ 和 $\tilde{L} = \tilde{\mathbf{x}}_0 + \tilde{U}$ 。那么 $L \subseteq \tilde{L}$ 当且仅当 $U \subseteq \tilde{U}$ 且 $\mathbf{x}_0 - \tilde{\mathbf{x}}_0 \in \tilde{U}$ 。仿射子空间通常由参数描述：考虑 V 的一个 k 维仿射空间 $L = \mathbf{x}_0 + U$ 。如果 $(\mathbf{b}_1, \dots, \mathbf{b}_k)$ 是 U 的一个有序基，那么 L 中的每一个元素 \mathbf{x} 都可以唯一地表示为

$$\mathbf{x} = \mathbf{x}_0 + \lambda_1 \mathbf{b}_1 + \cdots + \lambda_k \mathbf{b}_k,$$

(2.13)

其中 $\lambda_1, \dots, \lambda_k \in \mathbb{R}$ 。这个表示称为 L 的参数方程，具有方向向量 $\mathbf{b}_1, \dots, \mathbf{b}_k$ 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线，可以写为 $\mathbf{y} = \mathbf{x}_0 + \lambda \mathbf{b}_1$ ，其中 $\lambda \in \mathbb{R}$ 且 $U = \text{span}[\mathbf{b}_1] \subseteq \mathbb{R}^n$ 是 \mathbb{R}^n 的一个一维子空间。这意味着一条线由一个支点 \mathbf{x}_0 和一个定义方向的向量 \mathbf{b}_1 定义。图2.13给出了一个说明。

\mathbb{R}^n 中的二维仿射子空间称为平面。平面的参数方程为 $\mathbf{y} = \mathbf{x}_0 + \lambda_1 \mathbf{b}_1 + \lambda_2 \mathbf{b}_2$ ，其中 $\lambda_1, \lambda_2 \in \mathbb{R}$ 且 $U = \text{span}[\mathbf{b}_1, \mathbf{b}_2] \subseteq \mathbb{R}^n$ 。这意味着一个平面由一个支点 \mathbf{x}_0 和两个线性无关的向量 $\mathbf{b}_1, \mathbf{b}_2$ 定义，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，相应的参数方程为 $\mathbf{y} = \mathbf{x}_0 + \sum_{i=1}^{n-1} \lambda_i \mathbf{b}_i$ ，其中 $\mathbf{b}_1, \dots, \mathbf{b}_{n-1}$ 形成一个 $(n-1)$ -维子空间 U 的基。这意味着一个超平面由一个支点 \mathbf{x}_0 和 $(n-1)$ 个线性无关的向量 $\mathbf{b}_1, \dots, \mathbf{b}_{n-1}$ 定义，它们张成方向空间。在 \mathbb{R}^2 中，一条线也是一个超平面。在 \mathbb{R}^3 中，一个平面也是一个超平面。

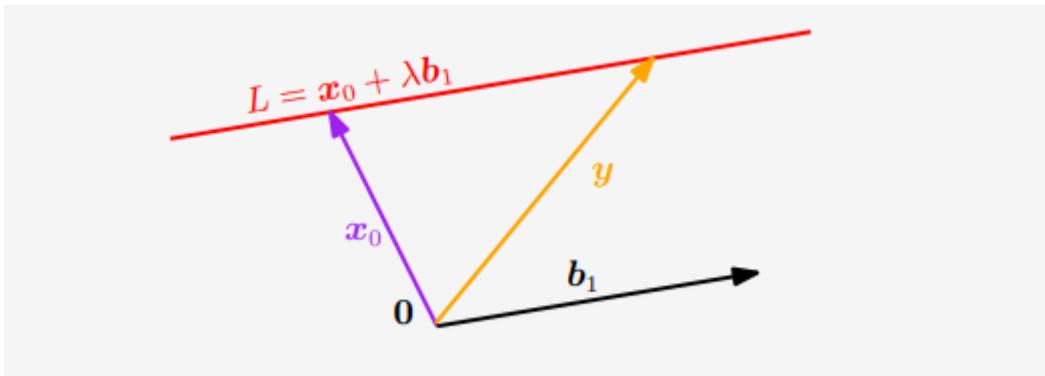


图2.13 线是仿射子空间。向量 \mathbf{y} 在线 $\mathbf{x}_0 + \lambda \mathbf{b}_1$ 上位于具有支点 \mathbf{x}_0 和方向 \mathbf{b}_1 的仿射子空间 L 中。

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 $\mathbf{x} \in \mathbb{R}^m$ ，线性方程组 $A\lambda = \mathbf{x}$ 的解要么是空集，要么是 \mathbb{R}^n 的一个 $n - \text{rk}(A)$ 维仿射子空间。特别地，线性方程 $\lambda_1 \mathbf{b}_1 + \cdots + \lambda_n \mathbf{b}_n = \mathbf{x}$ 的解（其中 $(\lambda_1, \dots, \lambda_n) \neq (0, \dots, 0)$ ）是 \mathbb{R}^n 中的一个超平面。在 \mathbb{R}^n 中，每一个 k -维仿射子空间是齐次线性方程组 $A\mathbf{x} = \mathbf{b}$ 的解，其中 $A \in \mathbb{R}^{m \times n}$ ， $\mathbf{b} \in \mathbb{R}^m$ 且 $\text{rk}(A) = n - k$ 。回想一下，对于齐次方程组 $A\mathbf{x} = \mathbf{0}$ ，解是一个向量子空间，我们也可以将其视为一个特殊的仿射空间，其支点 $\mathbf{x}_0 = \mathbf{0}$ 。◆



2.8.2 仿射映射

和我们在 2.7 节中讨论的线性空间之间的线性映射类似，我们也可以定义两个仿射空间之间的仿射映射。这样一来我们在线性空间中得到的许多性质可以在仿射空间中得以保持。

定义 2.26 (仿射空间) 对两个线性空间 V 和 W ，线性映射 $\Phi : V \rightarrow W$ ，以及 $a \in W$ ，就可以定义 V 到 W 的仿射映射

$$\phi : V \rightarrow W \quad (2.132)$$

$$x \mapsto a + \Phi(x). \quad (2.133)$$

其中的向量 a 称为 ϕ 的位移向量。

- 每个仿射映射 $\phi : V \rightarrow W$ 都可以唯一地写为一个线性映射 $\Phi : V \rightarrow W$ 和一个位移 $\tau : W \rightarrow W$ 的复合，也即 $\phi = \tau \circ \Phi$ 。
- 两个仿射映射的复合还是仿射映射
- 仿射映射保持几何结构（如维数和平行的性质）

< 上一章节

下一章节 >

2.7 线性映射

2.9 拓展阅读



2.9 拓展阅读

学习线性代数的资源有很多，包括Strang (2003)、Golan (2007)、Axler (2015)、Liesen and Mehrmann (2015) 的教科书。我们在本章的介绍中也提到了一些在线资源。我们在这里只讨论了高斯消去法，但有许多其他方法来求解线性方程组，我们参考 Stoer and Burlish (2002)，Golub and Van Loan (2012) 以及 Horn and Johnson (2013) 的数值线性代数教科书进行深入讨论。

在本书中，我们解读了线性代数中的一些话题（例如向量、矩阵、线性无关、基等）以及线性空间中测度的一些话题。在第三章中，我们将会介绍内积的概念，这导出了范数的概念。这些概念让我们能够定义角度、长度与距离，我们将使用这些概念来进行正交投影。投影是很多机器学习算法的关键，例如线性回归与主成分分析，我们会分别在第9章与第10章中讨论它们。

< 上一章节

2.8 仿射空间

下一章节 >

2.10 番外篇：多重线性代数与张量



番外篇 多重线性代数与张量

本文主要参照 Youtube 博主 [@eigenchris](#) 的张量简介视频。

- 原链接: <https://www.youtube.com/watch?v=8ptMTLzV4-I>
- b站搬运链接: <https://www.bilibili.com/video/BV1cr4y147eW?>

0 张量就是多维数组吗？

0.1 张量的三个理解

0.1.1 数组定义：张量是多维数组

Tensors are multi-dimensional arrays 张量是多维数组

常用名称	张量阶数
数	0
向量	1
矩阵	2
三维及以上的数组	3+

- 这是一个错误的定义，数组只是张量的表示，而不是张量本身



0.1.2 坐标定义：张量是坐标变换下的不变量

Tensors are objects that are **invariant** under a change of coordinates, and has components that change in a special, predictable way under a change of coordinates

张量是在坐标变换下**保持不变**的数学对象，其分量在坐标变换时以一种特殊且可预测的方式变化

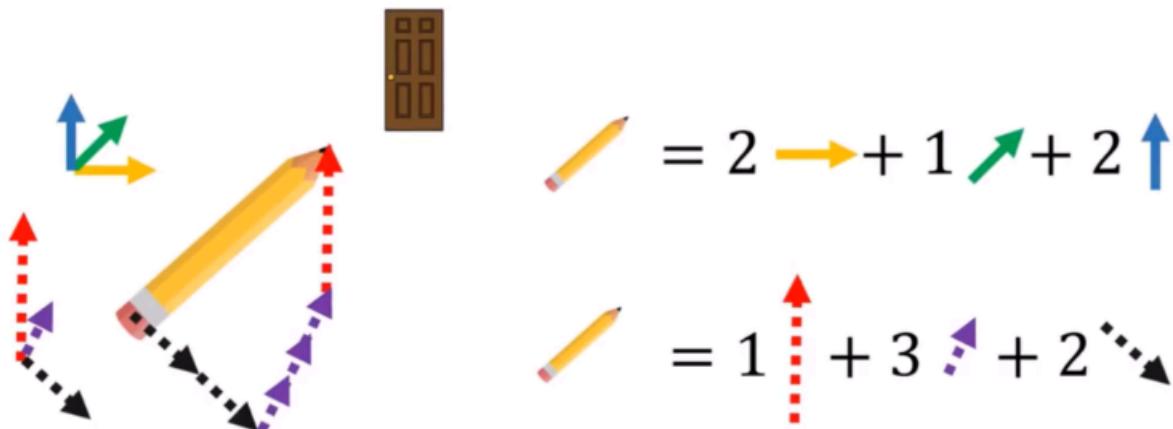


图1 坐标是变化的，笔是不变的

- 拿起你手中的笔，将其指向你房间的门或其他地方。不论你将你手中的笔指向哪里，它的形状、大小等性质都不会改变。
- 想象你现在建立了一个三维坐标系（虽然这听起来很蠢），这支笔可以写成基向量的线性组合。当你建立另一个坐标系时，这支笔在这个坐标系下的坐标会发生变化。而笔本身并没有改变。
- 不同坐标系下笔对应的坐标是变化的，但笔是不变的——向量在不同坐标系下的表示是变化的，而向量本身是不变的

0.1.3 张量的抽象定义



Tensor is a collection of vectors and covectors combined together using the **tensor product**

张量是一组使用张量积组合在一起的向量和余向量

读者在此可能会遇见问题：这个定义确实简洁，我听过“向量”但我不知道什么是“余向量”，也不知道什么是“张量积”。此定义我们先按下不表，到最后想必读者在明白一切后会明白这个定义的绝妙之处。

1 重回线性代数

1.1 基变换

我们举二维Euclid空间（平面） \mathbb{R}^2 作为例子。我们取平面上的两个基： $\{\mathbf{e}_1, \mathbf{e}_2\}$ 和 $\{\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2\}$ 。我们在此做一个约定，从 $\{\mathbf{e}_1, \mathbf{e}_2\}$ 变换到 $\{\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2\}$ 我们称为**正向变换**（forward transformation），反之就称为**逆向变换**（backward transformation）。按照线性空间性质，后者的每个基向量一定可以写成前面的基向量的线性组合：

$$\begin{aligned}\tilde{\mathbf{e}}_1 &= f_{1,1}\mathbf{e}_1 + f_{2,1}\mathbf{e}_2, \\ \tilde{\mathbf{e}}_2 &= f_{1,2}\mathbf{e}_1 + f_{2,2}\mathbf{e}_2;\end{aligned}\tag{1}$$

因此正向变换矩阵就可以写为

$$\mathbf{F} = \begin{bmatrix} f_{1,1} & f_{1,2} \\ f_{2,1} & f_{2,2} \end{bmatrix}.\tag{2}$$

反过来， $\{\mathbf{e}_1, \mathbf{e}_2\}$ 也可以由 $\{\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2\}$ 的线性组合表示出来

$$\begin{aligned}\mathbf{e}_1 &= b_{1,1}\tilde{\mathbf{e}}_1 + b_{2,1}\tilde{\mathbf{e}}_2, \\ \mathbf{e}_2 &= b_{1,2}\tilde{\mathbf{e}}_1 + b_{2,2}\tilde{\mathbf{e}}_2;\end{aligned}\tag{3}$$

因此得到后向变换矩阵：



$$\mathbf{B} = \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix}.$$

现在我们可以做一个小小的变换：将(1)代入(3)。对于 \mathbf{e}_1 ，我们有

$$\begin{aligned} \mathbf{e}_1 &= b_{1,1}\tilde{\mathbf{e}}_1 + b_{2,1}\tilde{\mathbf{e}}_2 && \text{按照(3)的写法} \\ &= b_{1,1}(f_{1,1}\mathbf{e}_1 + f_{2,1}\mathbf{e}_2) + b_{2,1}(f_{2,1}\mathbf{e}_1 + f_{2,2}\mathbf{e}_2) && \text{将(1)代入(3) } 5 \\ &= (f_{1,1}b_{1,1} + f_{1,2}b_{2,1})\mathbf{e}_1 + (f_{2,1}b_{1,1} + f_{2,2}b_{2,1})\mathbf{e}_2 && \text{将(1)代入(3)} \end{aligned}$$

此时注意到系数对应相等，就得到了

$$\begin{aligned} f_{1,1}b_{1,1} + f_{1,2}b_{2,1} &= 1, \\ f_{2,1}b_{1,1} + f_{2,2}b_{2,1} &= 0. \end{aligned} \tag{6}$$

这样的情形对于 \mathbf{e}_2 也是类似的。因此我们可以得到正向变换矩阵是反向变换矩阵的逆：

$$F = B^{-1}. \tag{7}$$

但为了作区分，下文中依然将使用 F 和 B 的记号。

再熟悉二维情形后，我们扩展至 n 维的情形。 \mathbb{R}^n 中有两个基 $\{\mathbf{e}_i\}_{i=1}^n$ 和 $\{\tilde{\mathbf{e}}_i\}_{i=1}^n$ ，它们可以互相表出：

$$\mathbf{e}_i = \sum_{j=1}^n b_{j,i} \tilde{\mathbf{e}}_j \quad (8a)$$

$$\tilde{\mathbf{e}}_j = \sum_{k=1}^n f_{k,j} \mathbf{e}_k \quad (8b)$$

然后我们再做和(5)相同的事情，即将(8b)代入(8a)：

$$\begin{aligned} \mathbf{e}_i &= \sum_{j=1}^n b_{j,i} \tilde{\mathbf{e}}_j \\ &= \sum_{j=1}^n b_{j,i} \left(\sum_{k=1}^n f_{k,j} \mathbf{e}_k \right) \\ &= \sum_{j=1}^n \left(\sum_{k=1}^n f_{k,j} b_{j,i} \right) \mathbf{e}_k \end{aligned} \tag{9}$$

因此我们有

$$\sum_{j=1}^n \left(\sum_{k=1}^n f_{k,j} b_{j,i} \right) = \delta_{j,k} = \begin{cases} 1, & j = k; \\ 0, & j \neq k. \end{cases} \quad (10)$$

这也再次证明了正向变换和反向变换互为逆变换。(10) 中的 $\delta_{j,k}$ 被称为 Kronecker-delta 记号。

1.2 线性空间

对于向量的理解，也有三种

1. 向量是一个数为元素的列表
2. 向量是像箭头一样的东西
3. 向量是线性空间中的元素

那么线性空间是什么？我想有过线性代数背景的读者都会知道。我们在此快速引入一些更加抽象的概念，然后自然地过渡到线性空间。它们在将来的内容中也会派上用场。

1.2.1 群、环、域

群 (group) 指的是一组资料 (G, \cdot) 其中 G 是一个非空集合， $\cdot : G \times G \rightarrow G$ 是 G 上的一个二元运算（通常称为乘法）。它们满足下面的条件

1. (结合律) 对于任意 $a, b, c \in G$ ，都有 $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
2. (中性元) 存在 1_G ，使得对任意 $g \in G$ 都有 $1_G \cdot g = g \cdot 1_G = g$
3. (逆元) 对于任意的 $g \in G$ ，总存在 $g^{-1} \in G$ ，使得 $g \cdot g^{-1} = g^{-1} \cdot g = 1_G$

满足交换律的群又称 **Abel 群**，我们一般把 Abel 群上的二元运算写成加法，把其上的单位元写成 **0**。

注 在 2.4 节介绍群时，我们为方便读者理解，用的是“单位元”。实际上当我们处理加法群的时候单位元显得并不合适，特别是在接下来的环和域中，会

同时存在两个运算，加法和乘法。因此我们将群中的**identity**称为中性元，以作区分。

环 (ring) 指的是一组资料 $(R, +, \times)$ ，其中 R 是一个非空集合，其上有加法、乘法两个运算，满足下面的条件：

1. (**Abel 群**) $(R, +)$ 是一个 Abel 群，其上有中性元 0_R
2. (**乘法**) R 上的乘法具有结合律，并有乘法单位元 1_R
3. (**分配律**) 环上的加法和乘法满足分配律 如果环上的乘法也满足交换律，则称该环为**交换环**。

如果交换环 R 上除了中性元 0_R 其他的任意元素 r 都存在逆（像实数、复数一样），则称其为一个 **域 (field)** 。

一些例子

- 整数集合 \mathbb{Z} 关于加法形成一个群
- 二维空间中的所有关于原点的旋转变换关于映射的复合构成一个群。其对应的所有矩阵常被记为 $SO(2)$
- 所有实系数多项式 $\mathbb{R}[x]$ 关于多项式的加法和乘法构成一个环
- 所有相同形状的方阵关于矩阵加法和乘法构成一个环，它不是交换环
- 有理数集 \mathbb{Q} 、实数集 \mathbb{R} 和复数集 \mathbb{C} 关于我们熟悉的乘法和加法构成域，这也是我们常说的**数域**

1.2.2 模和线性空间

假设 R 是一个交换环， $M = (M, +)$ 是一个交换群，其中性元是 0_M 。左 R -模包含一个这样的乘法： $\cdot : R \times M \rightarrow M$ ，满足下面的条件

1. 对任意 $r_1, r_2 \in R$, $m \in M$, 都有 $r_1 \cdot (r_2 \cdot m) = (r_1 r_2) \cdot m$
2. 乘法分配律成立，即

$$(r_1 + r_2) \cdot m = r_1 \cdot m + r_2 \cdot m, \quad r \cdot (m_1 + m_2) = r \cdot m_1 + r \cdot m_2$$



$$3. \mathbf{1}_R \cdot m = m$$

$$4. \mathbf{0}_R \cdot m = \mathbf{0}_m$$

看起来是不是越来越像我们在线性代数课程中学到的那八条公理了？事实上，如果将上面的 R 换成一个域 k ，我们就称 M 是一个 k -线性空间，并称 M 里面的元素是向量（vector）， R 里面的元素是标量（scalar）。我们将线性空间中的域降级为模，一些好的结论依然成立。这里需要声明一个符号滥用：线性空间的标量乘法（数乘）和域中的乘法是不一样的，但为表述简便，在不引起歧义的情况下，我们将省略线性空间中的所以乘法（就像上一节中一样）。

一些例子

- Euclid 空间。 \mathbb{R}^n 中的元素关于加法和 \mathbb{R} -数乘形成一个线性空间
- 连续函数空间。区间 $[a, b]$ 上的所有连续实值函数关于函数的加法和 \mathbb{R} 乘法构成一个线性空间（这是我们无法将其画成向量了）
- 线性映射。线性空间 V 和 W 之间的所有线性映射 $\text{Hom}(V, W)$ 关于线性映射的加法和数乘也构成线性空间

1.3 坐标变换

现在我们将视线转回 \mathbb{R}^n 中的向量。我们有两个基 $\{\mathbf{e}_i\}_{i=1}^n$ 和 $\{\tilde{\mathbf{e}}_i\}_{i=1}^n$ ，对于 \mathbb{R}^n 上的任一向量 \mathbf{v} ，我们都可以将其用这两个基表示

$$\mathbf{v} = v_1 \mathbf{e}_1 + v_2 \mathbf{e}_2 + \cdots + v_n \mathbf{e}_n \quad (11a)$$

$$\mathbf{v} = \tilde{v}_1 \tilde{\mathbf{e}}_1 + \tilde{v}_2 \tilde{\mathbf{e}}_2 + \cdots + \tilde{v}_n \tilde{\mathbf{e}}_n \quad (11b)$$

回忆基向量之间的变换，我们有

$$\begin{aligned} \mathbf{e}_i &= \sum_{j=1}^n b_{j,i} \tilde{\mathbf{e}}_j \quad (1) \\ \tilde{\mathbf{e}}_j &= \sum_{k=1}^n f_{k,j} \mathbf{e}_k \quad (2) \end{aligned}$$

将其代入 (11a)



$$\begin{aligned}
 \mathbf{v} &= v_1 \mathbf{e}_1 + v_2 \mathbf{e}_2 + \cdots + v_n \mathbf{e}_n \\
 &= \sum_{i=1}^n v_i \mathbf{e}_i \\
 &= \sum_{i=1}^n v_i \left(\sum_{j=1}^n b_{j,i} \tilde{\mathbf{e}}_j \right) \\
 &= \sum_{j=1}^n \left(\sum_{i=1}^n b_{j,i} \cdot v_i \right) \tilde{\mathbf{e}}_j
 \end{aligned} \tag{12}$$

我们也可以相应地对 (11b) 做类似的替换，然后相互对照系数，最后得到下面的结果，并将其与基变换的法则进行比较：

$\tilde{\mathbf{v}}_i = \sum_{j=1}^n b_{i,j} \mathbf{v}_j \quad (3)$ $\mathbf{v}_i = \sum_{j=1}^n f_{i,j} \tilde{\mathbf{v}}_j \quad (4)$	$\mathbf{e}_i = \sum_{j=1}^n b_{j,i} \tilde{\mathbf{e}}_j \quad (5)$ $\tilde{\mathbf{e}}_j = \sum_{k=1}^n f_{k,j} \mathbf{e}_k \quad (6)$
--	--

Contravariant

Covariant

读者不难发现，基变换和坐标变换的模式是反着来的。我们将坐标变换的性质称为 **反变的 (contra-variant)**，而将基变换的性质称为 **协变的 (co-variant)**。在后文中，我们对像向量坐标这样的反变的对象，我们将下标挪到上面。对于这个初看起来有些令人匪夷所思的性质，我们可以这样记忆：对于一个二维向量，如果将其伸长一倍，其坐标值也会增加一倍；如果将两个基向量伸长一倍，原向量的坐标值就会变为之前的二分之一。我们也可以考虑旋转：如果我们将两个基向量都逆时针旋转某角度，那么固定两个基向量的视角，看起来就像原来的向量顺时针旋转了相同的角度。

1.4 余向量及其分量的变换

我们在线性代数中常接触的两类向量被称为行向量和列向量。**千万不要简单的认为余向量就是将列向量转置得到的行向量，这句话只有当你使用标准正交基时才是对的。**事实上，我们可以将余向量看做吃进一个向量的线性映射： $\alpha : V \rightarrow \mathbb{R}$ 。这在我们熟知的行列向量的上下文中是十分自然的：



$$[2 \ 1] \begin{pmatrix} 3 \\ -4 \end{pmatrix} = [2 \ 1] \begin{bmatrix} 3 \\ -4 \end{bmatrix} = 2 \cdot 3 + 1 \cdot (-4) = 2.$$

我们说余向量是线性映射，具体说的是满足下面的条件

$$\alpha(n\mathbf{v} + m\mathbf{w}) = n\alpha(\mathbf{v}) + m\alpha(\mathbf{w}).$$

这条性质被称为**线性性 (linearity)**。对作用于二维Euclid空间的余向量，我们可以像画等高线一样将其画出来：

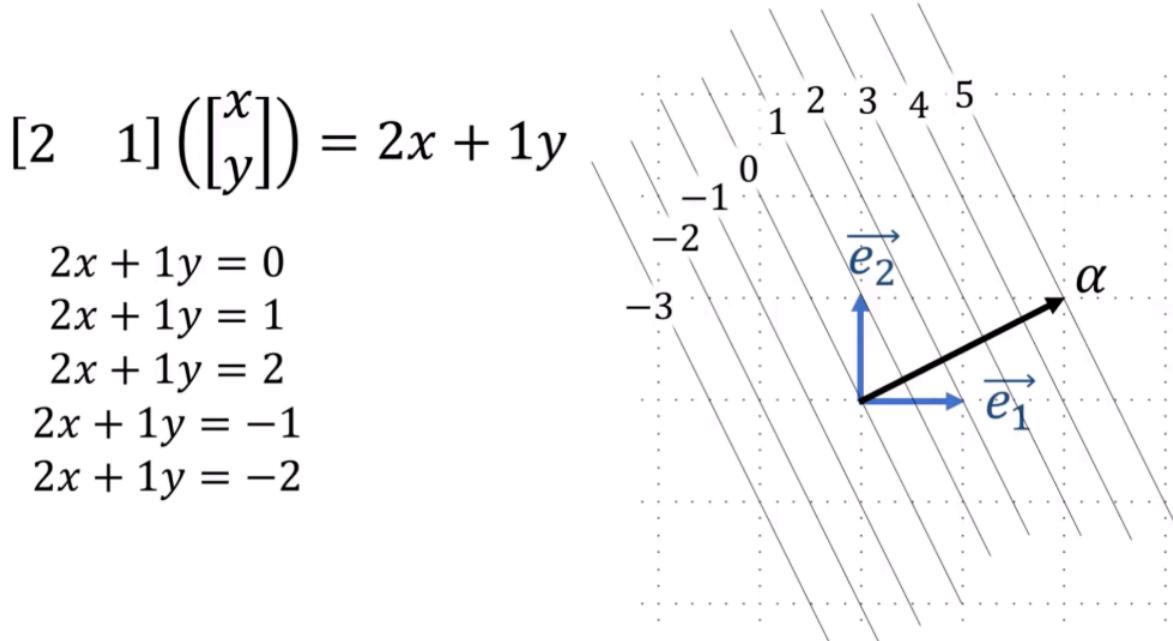


图2 余向量作用在向量上

想知道这个余向量作用在任意向量上的值是多少，只要看其跨越了多少等高线即可（不严谨的说法）。像一般的函数一样，余向量也可以加减，数乘。事实上对于作用于线性空间 V 的余向量，它们的家位于一个被称为是**对偶线性空间**的地方，它被称为 V^* （也有书将其写为 V^\vee ），对偶线性空间的几条基本性质和线性空间类似，这里不再赘述。

现在我们考虑余向量的变换。我们已经知道，线性空间的元素可以写成基向量的线性组合，那么构成余向量的基本部件是什么？以 n 维线性空间为例子，并引入这样的余向量：

$$\epsilon^i \in V^*, \epsilon^i(\mathbf{e}_j) = \delta_{i,j} \quad (13)$$

这样就可以提取出余向量的各个“分量”了。我们接着拿一个任意的向量来实验：



$$\alpha(\mathbf{v}) = \alpha \left(\sum_{i=1}^n v^i \mathbf{e}_i \right) \quad \text{展开成基向量的线性组合} \quad (7)$$

$$= \sum_{i=1}^n v^i \alpha(\mathbf{e}_i) \quad \text{余向量的线性性} \quad (8)$$

$$= \sum_{i=1}^n \epsilon^i(v) \alpha(\mathbf{e}_i) \quad (13) \text{中的定义} \quad (9)$$

$$= \sum_{i=1}^n \alpha_i \epsilon^i(v) \quad \text{将 } \alpha(\mathbf{e}_i) \text{ 写为 } \alpha_i \quad (10)$$

这就得到了余向量的一个展开，上文中的 ϵ^i 称为 **对偶基 (dual basis)**。读者可能发现，对偶基的标号是上标，而余向量的标号是下标。我们接下来就来证明余向量的基变换是反变的，坐标变换是协变的。假设对偶空间 V^* 中有两个对偶基 $\{\epsilon^j\}_{j=1}^n$ 和 $\{\tilde{\epsilon}^i\}_{i=1}^n$ ，对于每个对偶基 $\tilde{\epsilon}^i$ ，我们可以将其写成前一个对偶基中基向量的线性组合：

$$\tilde{\epsilon}^i = \sum_{j=1}^n Q_{i,j} \epsilon^j$$

我们将 $\tilde{\epsilon}^i$ 作用在 V 的一个基向量 $\tilde{\mathbf{e}}_k$ 上：

$$\tilde{\epsilon}^i(\tilde{\mathbf{e}}_k) = \delta_{i,k} = \sum_{j=1}^n Q_{i,j} \epsilon^j(\tilde{\mathbf{e}}_k) \quad (11)$$

$$= \sum_{j=1}^n Q_{i,j} \epsilon^j \left(\sum_{l=1}^n f_{l,k} \mathbf{e}_l \right) \quad (12)$$

$$= \sum_{l=1}^n \left[\sum_{j=1}^n Q_{i,j} f_{l,k} \epsilon^j(\mathbf{e}_l) \right] \quad (13)$$

$$= \sum_{l=1}^n \left[\sum_{j=1}^n Q_{i,j} f_{l,k} \delta_{j,l} \right] \quad (14)$$

$$= \sum_{j=1}^n Q_{i,j} f_{j,k} \quad (15)$$

这就说明了由 $\{\epsilon^j\}_{j=1}^n$ 变换到 $\{\tilde{\epsilon}^i\}_{i=1}^n$ 的矩阵是 F 的逆矩阵，也就是 B 。对偶地，读者也可以自行验证反过来变换的矩阵 P 是 B 的逆，也就是 F 。这说明了对偶基是反变的。我们把向量拿来作比较，如下所示



$$\mathbf{e}_i = \sum_{j=1}^n b_{j,i} \tilde{\mathbf{e}}_j \quad (16)$$

$$\tilde{\mathbf{e}}_j = \sum_{k=1}^n f_{k,j} \mathbf{e}_k \quad (17)$$

Covariant

$$\epsilon^i = \sum_{j=1}^n f_{i,j} \tilde{\epsilon}^j \quad (18)$$

$$\tilde{\epsilon}^j = \sum_{k=1}^n b_{j,k} \epsilon^k \quad (19)$$

Contravariant

类似地，也可以证明余向量的坐标是协变的，这里不再赘述。

1.5 线性映射和矩阵

矩阵是线性映射在某个基下的表示。几何直观上看，如果把一个带有网格线的平面经过线性映射，得到的结果一定符合下面三个要求，如下图所示

1. 网格线映射过后依然是直的
2. 间隔相等的网格线映射过后间隔还是相等（间隔的距离可能变化）
3. 原点映射过后还是原点



图3 线性映射的集合直观。第一个图是映射前的网格，后面三个图分别代表由三个不同映射作用后的网格

抽象地讲，线性映射是线性的（废话！）其遵守的线性性我们在余向量时已有提及，这里不再赘述，而是展示通常的矩阵和矩阵乘法是怎么来的。假设有一个线性映射 $L : V \rightarrow V$ ，其中 V 的维数是 2。由于线性空间的元素总能写成基向量的线性组合，又由线性映射的性质，我们只需知道线性映射把基向量打到了哪里，就知道任意向量经过线性映射后的结果。假设 L 作用在基向量上的结果如下，我们就像这样将其写为矩阵：

$$L(\mathbf{e}_1) = L_1^1 \mathbf{e}_1 + L_1^2 \mathbf{e}_2 \quad (20)$$

$$L(\mathbf{e}_2) = L_2^1 \mathbf{e}_1 + L_2^2 \mathbf{e}_2 \quad (21)$$

$$L = \begin{bmatrix} L_1^1 & L_1^2 \\ L_2^1 & L_2^2 \end{bmatrix} \quad (14)$$

我们现在随便将一个向量 \mathbf{v} 送进去看看：

$$\begin{aligned}
L(\mathbf{v}) &= L(v^1 \mathbf{e}_1 + v^2 \mathbf{e}_2) & (22) \\
&= v^1 L(\mathbf{e}_1) + v^2 L(\mathbf{e}_2) & \text{线性性} & (23) \\
&= v^1 (L_1^1 \mathbf{e}_1 + L_1^2 \mathbf{e}_2) + v^2 (L_2^1 \mathbf{e}_1 + L_2^2 \mathbf{e}_2) & \text{代入 (14)} & (24) \\
&= (v^1 L_1^1 + v^2 L_2^1) \mathbf{e}_1 + (v^1 L_1^2 + v^2 L_2^2) \mathbf{e}_2 & (25)
\end{aligned}$$

这就定义了矩阵对列向量的乘法。矩阵对矩阵的乘法也是类似的，其本质是线性映射的复合。读者可以通过线性映射的性质自行推导矩阵乘法公式。

1.6 线性映射在不同基下表示的变换

从上一节读者可以看到，线性映射的矩阵与其采用的基是紧密相关的。我们现在看看如果换成一个其他的基，线性映射的矩阵会有什么变化。考虑 n 维线性空间到它自己的线性映射 L ，我们将基 $\{\mathbf{e}_j\}_{j=1}^n$ 下的矩阵写为 L ，在基 $\{\tilde{\mathbf{e}}_j\}_{j=1}^n$ 下的矩阵写为 \tilde{L} 。我们将 L 作用在基向量 $\tilde{\mathbf{e}}_i$ 上，有

$$\sum_{q=1}^n \tilde{L}_i^q \tilde{\mathbf{e}}_q = L(\tilde{\mathbf{e}}_i) = L\left(\sum_{j=1}^n F_i^j \mathbf{e}_j\right) \quad (26)$$

$$= \sum_{j=1}^n F_i^j L(\mathbf{e}_j) \quad (27)$$

$$= \sum_{j=1}^n F_i^j \left(\sum_{k=1}^n L_j^k \mathbf{e}_k \right) \quad (28)$$

$$= \sum_{j=1}^n F_i^j \left[\sum_{k=1}^n L_j^k \left(\sum_{l=1}^n B_k^l \tilde{\mathbf{e}}_l \right) \right] \quad (29)$$

$$= \sum_{l=1}^n \left[\sum_{j=1}^n \sum_{k=1}^n B_k^l L_j^k F_i^j \right] \tilde{\mathbf{e}}_l \quad (30)$$

其中的 F_i^j , B_k^l 是之前提到的正向变换矩阵和逆向变换矩阵。可见 $\tilde{L} = BLF$ 。从线性映射的角度也好理解。对于 $\{\tilde{\mathbf{e}}_j\}_{j=1}^n$ 下的向量，先用 F 将其坐标转换为基 $\{\mathbf{e}_j\}_{j=1}^n$ 下的坐标，然后在这个基下做线性映射 L ，然后再将 $\{\mathbf{e}_j\}_{j=1}^n$ 下的像用 B 转换回 $\{\tilde{\mathbf{e}}_j\}_{j=1}^n$ 下的坐标，如下图所示（注意这里的作用顺序是 F , L , B , 而不是反过来！）。

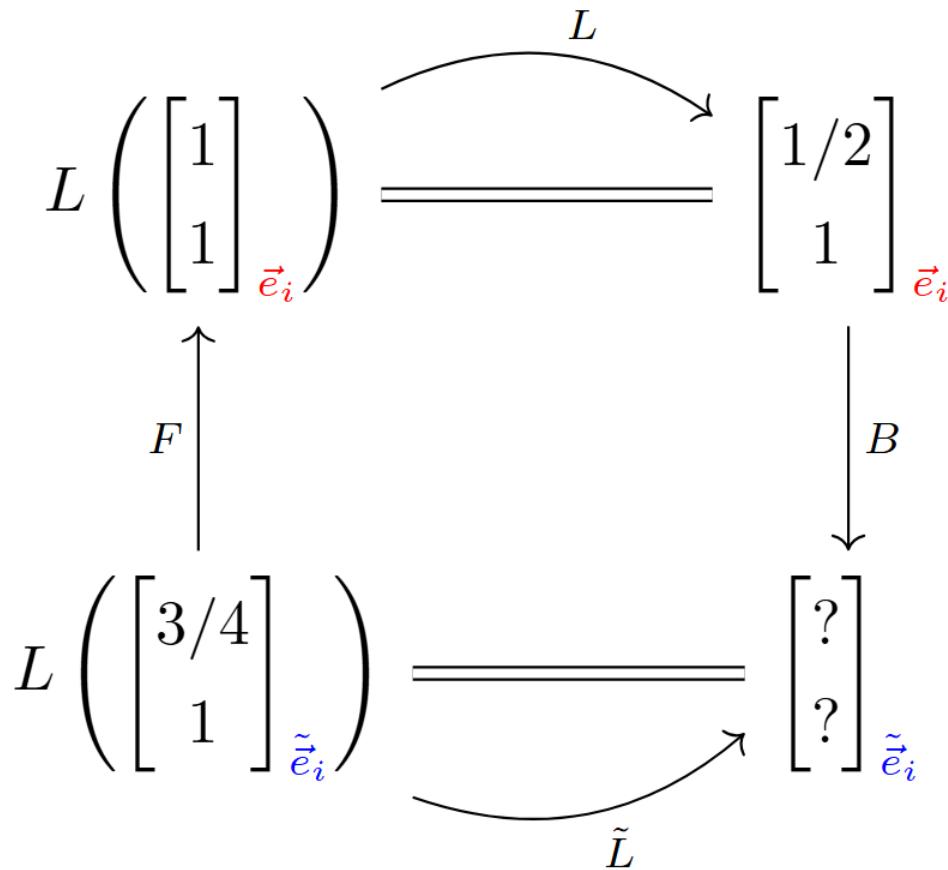


图4 线性映射在不同基下表示的转换

相应地，我们也可以得到 $L = F\tilde{L}B$ （只要将上图的两个竖直箭头反向即可）。

讲了这么多，这和张量有什么关系呢？余向量的基变换是反变的，我们将其称为一种 $(1, 0)$ -张量；向量是协变的，我们将其称为一种 $(0, 1)$ -张量。而线性变换在基变换下既用到了正向变换又用到了反向变换，我们将其称为一种 $(1, 1)$ -张量。

2 张量一瞥

2.1 度量张量

2.1.1 向量的长度

我们在中学时学过勾股定理，并用它来衡量一个向量的长度。但其限制是直角三角形，在线性空间基不是标准正交基时不再成立。事实上，向量的长度 $\|v\|$ 由下式给



$$\|v\|^2 = \langle v, v \rangle$$

其中 $\langle \cdot, \cdot \rangle$ 是内积。对于一般的向量，我们可以写出它的平方长度：

$$\|v\| = (v^1)^2 \langle e_1, e_1 \rangle + 2v^1 v^2 \langle e_1, e_2 \rangle + (v^2)^2 \langle e_2, e_2 \rangle$$

也可以写成矩阵形式：

$$\|v\| = [v^1 \quad v^2] \underbrace{\begin{bmatrix} \langle e_1, e_1 \rangle & \langle e_1, e_2 \rangle \\ \langle e_2, e_1 \rangle & \langle e_2, e_2 \rangle \end{bmatrix}}_{g_{e_i}} \begin{bmatrix} v^1 \\ v^2 \end{bmatrix}$$

其中 g_{e_i} 就成为在基 e_i 下的度量张量。这是我们可以立即看出勾股定理在直角三角形时恰好成立的原因是不同的标准正交基向量的内积为零，因为此时度量张量是单位矩阵 I ，而这恰好给出了勾股定理。当基变化时，度量张量会怎么变化呢？请看：将 g_{e_i} 简记为 g ，把新的基 $\{\tilde{e}_j\}$ 下的度量张量记作 \tilde{g} ，则其分量 $\tilde{g}_{i,j}$ 是两个新的基向量的内积

$$\tilde{g}_{i,j} = \langle \tilde{e}_i, \tilde{e}_j \rangle \tag{31}$$

$$= \left\langle \sum_k^n F_i^k e_k, \sum_{l=1}^n F_j^l e_l \right\rangle \tag{32}$$

$$= \sum_{k=1}^n \sum_{l=1}^n F_i^k F_j^l \langle e_k, e_l \rangle \tag{33}$$

$$= \sum_{k=1}^n \sum_{l=1}^n F_i^k F_j^l g_{k,l} \tag{34}$$

因此我们可以看到基发生变化后，度量张量要做两次正向变换得到新的基下的度量张量，也即两个协变变换。因此我们称其为一种 $(0, 2)$ -张量。

此时大家应该理解了张量的第二个定义：张量是在坐标变换变换下**保持不变**的数学对象，其分量在坐标变换时以一种特殊且可预测的方式变化。一般的张量的变换可以写成

$$\tilde{T}_{x,y,z,\dots}^{a,b,c,\dots} = (B_i^a B_j^b B_k^c \dots) T_{r,s,t,\dots}^{i,j,k,\dots} (F_x^r F_y^s F_z^t \dots)$$

其中 T 的下标是协变分量，其变换要使用正向变换 F ，而上标是反变分量，其变换要使用逆向变换 B 。如果一个张量有 m 个反变分量， n 个协变分量，我们就称其为

一个 (m, n) -张量。



2.2 双线性型

读者可能已注意到，上文中的内积或度量张量单独对输入的两个向量都是线性的。这样的函数有一个名字：**双线性型（bilinear form）**。线性空间 V 上的双线性型 $\mathcal{B} : V \times V \rightarrow \mathbb{R}$ 满足固定一个分量，对另一个分量线性。具体而言就是

1. $a\mathcal{B}(v, w) = \mathcal{B}(av, w) = \mathcal{B}(v, aw)$
2. $\mathcal{B}(v + u, w) = \mathcal{B}(v, w) + \mathcal{B}(u, w)$
3. $\mathcal{B}(v, w + u) = \mathcal{B}(v, w) + \mathcal{B}(v, u)$

上文中提到的余向量也被称为**线性型（linear form）**。度量张量是一个特殊的双线性型，交换度量张量的两个输入，输出的值不变。一般的双线性型没有这样的良好性质。

2.4 线性映射由向量和余向量构成

这看起来很离谱。但不要着急，我们做下面的事情：对于矩阵 A 我们总可以拆成这样的线性组合：

$$A = \sum_{i=1}^n \sum_{j=1}^m A_i^j E_{i,j}.$$

其中 $A_{i,j}$ 是矩阵在 i 行 j 列处的分量， $E_{i,j}$ 是除了 i 行 j 列其他地方都为零的矩阵。由基本的矩阵乘法知识我们有 $E_{i,j} = e_i e_j^\top$ 。所以上式写成余向量的形式就是

$$A = \sum_{i=1}^n \sum_{j=1}^m A_i^j (e_i \epsilon^j) = \sum_{j=1}^m \left(\sum_{i=1}^n A_i^j e_i \right) \epsilon^j$$

而最右边那一项说的就是我们通过线性映射作用在基上的结果构造矩阵的过程。回到主题，我们可以看见 $\{e_i \epsilon_j\}$ 构成了线性映射空间的一个基。这样有什么好处呢？好处在于我们之前费很大功夫得出的线性映射的矩阵在基变换下的变化在用上这样的表示之后无比显然：

$$L = \sum_{k,l=1}^n L_l^k e_k \epsilon^l = \sum_{k=1}^n L_l^k \left(\sum_{i=1}^n B_k^i \tilde{e}_i \right) \left(\sum_{j=1}^n F_j^l \tilde{\epsilon}^j \right) = \sum_{i,j,k,l} \textcolor{red}{L_l^k B_k^i F_j^l} \tilde{e}_i \tilde{\epsilon}^j \quad (35)$$

接下来我们将不将它们写成 $e_i \epsilon^j$ ，而将其写成 $\textcolor{red}{e_i \otimes \epsilon^j}$ 。在用于矩阵这样的对象时（行列向量当然也是矩阵） \otimes 被称为 **Kronecker积**。它把左边的矩阵搬到右边的每个元素的位置，然后数乘那个元素：

$$\begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \otimes [\alpha_1 \quad \alpha_2] = \left[\alpha_1 \cdot \begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \quad \alpha_2 \cdot \begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \right] = \begin{bmatrix} [\alpha_1 v^1] & [\alpha_1 v^1] \\ [\alpha_1 v^2] & [\alpha_1 v^2] \end{bmatrix}$$

这看起来和将 Kronecker 积改成矩阵乘法得到的结果没什么区别，但这一思想对我们十分重要。

2.5 双线性型由余向量构成

双线性型吃进两个向量，吐出一个数。它也可以写成余向量的组合：

$$\mathcal{B} = \sum_{i,j} \mathcal{B}_{i,j} \epsilon^i \epsilon^j = \sum_{i,j} \mathcal{B}_{i,j} (\epsilon^i \otimes \epsilon^j)$$

在使用这样的表示后双线性型在基变换下的变化也是显然的。我们也可以按照类似上一节的方法将行向量的 Kronecker 积结果写出来：

$$[\alpha_1 \quad \alpha_2] \otimes [\beta_1 \quad \beta_2] = [[\alpha_1 \beta_1 \quad \alpha_2 \beta_1] \quad [\alpha_1 \beta_2 \quad \alpha_2 \beta_2]]$$

嗯……这看起来不太对劲，在我们的意识中度量张量似乎是个矩阵，为什么得到了这样的东西？回想度量张量的矩阵形式，它按道理应该吃两个向量，结果确实一个行向量和一个列向量。它之所以为一个矩阵是因为我们滥用了记号，让一个向量变成行向量（余向量），迫使度量张量看起来像个矩阵。如果我们就使用列向量的表示，那么上面那个怪怪的玩意儿其实更加合理：

$$\begin{aligned} [[\alpha_1 \beta_1 \quad \alpha_2 \beta_1] \quad [\alpha_1 \beta_2 \quad \alpha_2 \beta_2]] \begin{bmatrix} v^1 \\ v^2 \end{bmatrix} \begin{bmatrix} w^1 \\ w^2 \end{bmatrix} &= \left(v^1 [\alpha_1 \beta_1 \quad \alpha_2 \beta_1] + v^2 [\alpha_1 \beta_2 \quad \alpha_2 \beta_2] \right) \begin{bmatrix} w^1 \\ w^2 \end{bmatrix} \\ &= [v^1 \alpha_1 \beta_1 + v^2 \alpha_1 \beta_2 \quad v^1 \alpha_2 \beta_1 + v^2 \alpha_2 \beta_2] \begin{bmatrix} w^1 \\ w^2 \end{bmatrix} \end{aligned}$$

张量积和 Kronecker 共用一个符号，但含义不同。张量积作用于张量、线性空间和模，而 Kronecker 积作用于数组（如矩阵）。张量积具有多重线性，即



- $(\alpha v_1) \otimes v_2 \otimes v_3 \otimes \cdots = v_1 \otimes (\alpha v_2) \otimes v_3 \otimes \cdots = \cdots$
- $(v_1 + w) \otimes v_2 \otimes v_3 \otimes \cdots = v_1 \otimes v_2 \otimes v_3 \otimes \cdots + w \otimes v_2 \otimes v_3 \otimes \cdots$ (对其他位置的对象也一样) 当然张量积的多重线性性 Kronecker 也具备。

一般的张量也可以写成向量和余向量之间的张量积的线性组合。换句话说，形如 $a \otimes b$ 这样的对象是张量的基本组成部件。有了它们就可以轻易地得到任意形式的向量在基变换下的变化规律了。对于连个张量缩并，我们需要指出对那些分量缩并。例如 $Q_{j,k}^i$ 与 $D^{a,b}$ ，有多种缩并形式，如 $Q_{j,k}^i D^{j,k}$ 和 $Q_{j,k}^i D^{j,l}$ ，其中两个张量对应相同的上下标就是缩并求和掉的维度，在 `pytorch` 中可以使用 `einsum` 函数完成张量的缩并。

使用 Kronecker 积可以方便地将张量表示为数组，但容易使人迷失在数字的阵列中，而逐渐忘记该张量的性质。最后，线性空间的张量积这样一来也不难理解。两个线性空间 V 和 W 的张量积 $V \otimes W$ 里面包含了形如 $v \otimes w$ 的元素。其中 v 和 w 分别是 V 和 W 中的元素。

< 上一章节

下一章节 >

2.9 拓展阅读

习题



本教程由 [Datawhale 开源社区](#) 编译，与对应的英文原版均开源免费

练习 2.1

我们考虑 $(\mathbb{R} \setminus \{-1\}, \star)$, 其中

$$a \star b := ab + a + b, \quad a, b \in \mathbb{R} \setminus \{-1\}$$

- a. 证明 $(\mathbb{R} \setminus \{-1\}, \star)$ 是一个 Abel 群。 b. 在 Abel 群 $(\mathbb{R} \setminus \{-1\}, \star)$ 中解方程

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

练习 2.2

设 n 是 $\mathbb{N} \setminus \{0\}$ 中的元素。设 k, x 是 \mathbb{Z} 中的元素。我们定义整数 k 的同余类 \bar{k} 为集合

$$\begin{aligned}\bar{k} &= \{x \in \mathbb{Z} \mid x - k = 0 \pmod{n}\} \\ &= \{x \in \mathbb{Z} \mid \exists a \in \mathbb{Z}, \text{使得 } x - k = n \cdot a\}\end{aligned}$$

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。Euclid 除法表明这个集合是一个包含 n 个元素的有限集:

$$\mathbb{Z}_n = \{\bar{0}, \bar{1}, \dots, \bar{n-1}\}$$

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 Abel 群吗? b. 现在我们为所有 a 和 b 在 \mathbb{Z}_n 中定义另一个运算 \otimes :

$$a \otimes b = a \times b$$

其中 $a \times b$ 表示 \mathbb{Z} 中的通常乘法。设 $n = 5$ 。绘制 $\mathbb{Z}_5 \setminus \{0\}$ 中元素在 \otimes 下的乘法表，即计算所有 a 和 b 在 $\mathbb{Z}_5 \setminus \{0\}$ 中的乘积 $a \otimes b$ 。由此，证明 $\mathbb{Z}_5 \setminus \{0\}$ 在 \otimes 下是封闭的，并且存在单位元。列出 $\mathbb{Z}_5 \setminus \{0\}$ 中所有元素在 \otimes 下的逆元。得出结论： $(\mathbb{Z}_5 \setminus \{0\}, \otimes)$ 是一个 Abel 群。c. 证明 $(\mathbb{Z}_8 \setminus \{0\}, \otimes)$ 不是一个群。d. 回忆 Bezout 定理指出，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 $(\mathbb{Z}_n \setminus \{0\}, \otimes)$ 是一个群当且仅当 $n \in \mathbb{N} \setminus \{0\}$ 是质数。

练习 2.3

考虑以下定义的 3×3 矩阵集合 G :

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

我们将 \cdot 定义为标准矩阵乘法。 (G, \cdot) 是一个群吗？如果是，它是 Abel 群吗？请证明你的答案。

练习 2.4

计算以下矩阵乘积（如果可能的话）：

a.

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

b.

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

c.



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

d.

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

e.

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

练习 2.5

求解以下非齐次线性方程组 $Ax = b$ 的所有解, 其中 A 和 b 定义如下:

a.

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

b.

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

练习 2.6

使用高斯消元法，求解非齐次方程组 $Ax = b$ 的所有解，其中

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



练习 2.7

求解方程组 $Ax = 12x$ 的所有解 $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3$ ，其中

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

并且满足 $\sum_{i=1}^3 x_i = 1$ 。

练习 2.8

如果可能的话，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.

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



练习 2.9

以下哪些集合是 \mathbb{R}^3 的子空间?
a. $A = \{(\lambda, \lambda + \mu^3, \lambda - \mu^3) \mid \lambda, \mu \in \mathbb{R}\}$ b.
 $B = \{(\lambda^2, -\lambda^2, 0) \mid \lambda \in \mathbb{R}\}$ c. 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. $D = \{(\xi_1, \xi_2, \xi_3) \in \mathbb{R}^3 \mid \xi_2 \in \mathbb{Z}\}$

练习 2.10

以下向量集合是否线性无关?

a.

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

b.

$$x_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

练习 2.11

将

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

表示为以下向量的线性组合:



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

练习 2.12

考虑 \mathbb{R}^4 的两个子空间：

$$U_1 = \text{span} \left(\begin{bmatrix} 1 \\ 1 \\ -3 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ -1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} \right)$$

$$U_2 = \text{span} \left(\begin{bmatrix} -1 \\ -2 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ -2 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -3 \\ 6 \\ -2 \\ -1 \end{bmatrix} \right)$$

确定 $U_1 \cap U_2$ 的一个基。

练习 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

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

a. 确定 U_1 和 U_2 的维数。 b. 确定 U_1 和 U_2 的基。 c. 确定 $U_1 \cap U_2$ 的一个基。

练习 2.14

考虑两个子空间 U_1 和 U_2 , 其中 U_1 由 A_1 的列向量生成, U_2 由 A_2 的列向量生成, 其中

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

- a. 确定 U_1 和 U_2 的维数。 b. 确定 U_1 和 U_2 的基。 c. 确定 $U_1 \cap U_2$ 的一个基。

练习 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. 不借助基向量, 计算 $F \cap G$ 。 c. 找出 F 和 G 的基, 使用这些基向量计算 $F \cap G$, 并验证与上一问题的结果是否一致。

练习 2.16

以下映射是否为线性映射?

- a. 设 $a, b \in \mathbb{R}$ 。

$$\Phi : L^1([a, b]) \rightarrow \mathbb{R}$$

$$f \mapsto \Phi(f) = \int_a^b f(x) dx$$

其中 $L^1([a, b])$ 表示在 $[a, b]$ 上的可积函数集合。

b.

$$\Phi : C^1 \rightarrow C^0$$

$$f \mapsto \Phi(f) = f'$$

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



c.

$$\Phi : \mathbb{R} \rightarrow \mathbb{R}$$

$$x \mapsto \Phi(x) = \cos(x)$$

d.

$$\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$$

$$x \mapsto \begin{bmatrix} 1 & 2 & 3 \\ 1 & 4 & 3 \end{bmatrix} x$$

e. 设 $\theta \in [0, 2\pi)$,

$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$x \mapsto \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} x$$

练习 2.17

考虑线性映射

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

$$\Phi \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right) = \begin{bmatrix} 3x_1 + 2x_2 + x_3 \\ x_1 + x_2 + x_3 \\ x_1 - 3x_2 \\ 2x_1 + 3x_2 + x_3 \end{bmatrix}$$

求变换矩阵 A_Φ 。确定 $\text{rk}(A_\Phi)$ 。计算 Φ 的核和像。 $\dim(\ker(\Phi))$ 和 $\dim(\text{Im}(\Phi))$ 分别是多少？

练习 2.18

设 E 是一个线性空间。设 f 和 g 是 E 上的两个自同构，使得 $f \circ g = \text{id}_E$ （即 $f \circ g$ 是恒等映射 id_E ）。证明 $\ker(f) = \ker(g \circ f)$, $\text{Im}(g) = \text{Im}(g \circ f)$, 并且



$\ker(f) \cap \text{Im}(g) = \{0_E\}$ 。

练习 2.19

考虑一个内态射 $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, 其变换矩阵 (相对于 \mathbb{R}^3 的标准基) 为

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

a. 确定 $\ker(\Phi)$ 和 $\text{Im}(\Phi)$ 。b. 确定相对于基

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

的变换矩阵 \tilde{A}_Φ , 即执行向新基 B 的基变换。

练习 2.20

考虑 b_1, b_2, b'_1, b'_2 是 \mathbb{R}^2 中的四个向量, 它们在 \mathbb{R}^2 的标准基下的表示为

$$b_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad b_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad b'_1 = \begin{bmatrix} 2 \\ -2 \end{bmatrix}, \quad b'_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

我们定义两个有序基 $B = (b_1, b_2)$ 和 $B' = (b'_1, b'_2)$ 。a. 证明 B 和 B' 是 \mathbb{R}^2 的两个基, 并绘制这些基向量。b. 计算从 B' 到 B 的基变换矩阵 P^1 。c. 考虑 \mathbb{R}^3 中的三个向量 c_1, c_2, c_3 , 它们在 \mathbb{R}^3 的标准基下的定义为

$$c_1 = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}, \quad c_2 = \begin{bmatrix} 0 \\ -1 \\ 2 \end{bmatrix}, \quad c_3 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

我们定义 $C = (c_1, c_2, c_3)$ 。(i) 证明 C 是 \mathbb{R}^3 的一个基, 例如通过使用行列式 (见第 4.1 节)。(ii) 设 $C' = (c'_1, c'_2, c'_3)$ 是 \mathbb{R}^3 的标准基。确定从 C 到 C' 的基变换矩阵 P^2 。d. 考虑一个同态 $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, 使得

$$\Phi(b_1 + b_2) = c_2 + c_3$$

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

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 相对于有序基 B 和 C 的变换矩阵 A_Φ 。
e. 确定 A' , 即 Φ 相对于基 B' 和 C' 的变换矩阵。
f. 考虑向量 $x \in \mathbb{R}^2$, 其在 B' 中的坐标为 $[2, 3]^\top$ 。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 中的坐标。
(iii) 然后, 将 $\Phi(x)$ 用 c'_1, c'_2, c'_3 表示。
(iv) 使用 x 在 B' 中的表示和矩阵 A' 直接找到这个结果。

< 上一章节

2.10 番外篇：多重线性代数与张量



翻译：何瑞杰

第三章 解析几何 (70)

在第二章中，我们从一般但抽象的角度研究了向量、线性空间和线性映射。在本章中，我们将从几何直觉的视角考虑这些概念。例如，我们将考虑（Euclid 空间中的）两个向量的几何表示，计算它们的长度、它们之间的距离和夹角。我们需要将线性空间装配上诱导出其几何特征的内积以完成上面所说的事情。内积及其诱导的范数和度量与我们直觉中的“相似度”和“距离”相对应；我们将在第十二章中使用它们构建支持向量机模型。随后我们将使用上面定义的向量的长度和向量间的夹角讨论正交投影。它将在第九章中的极大似然估计和第十章中的主成分分析中占中心地位。

图 3.1 给出了本章的概念地图。

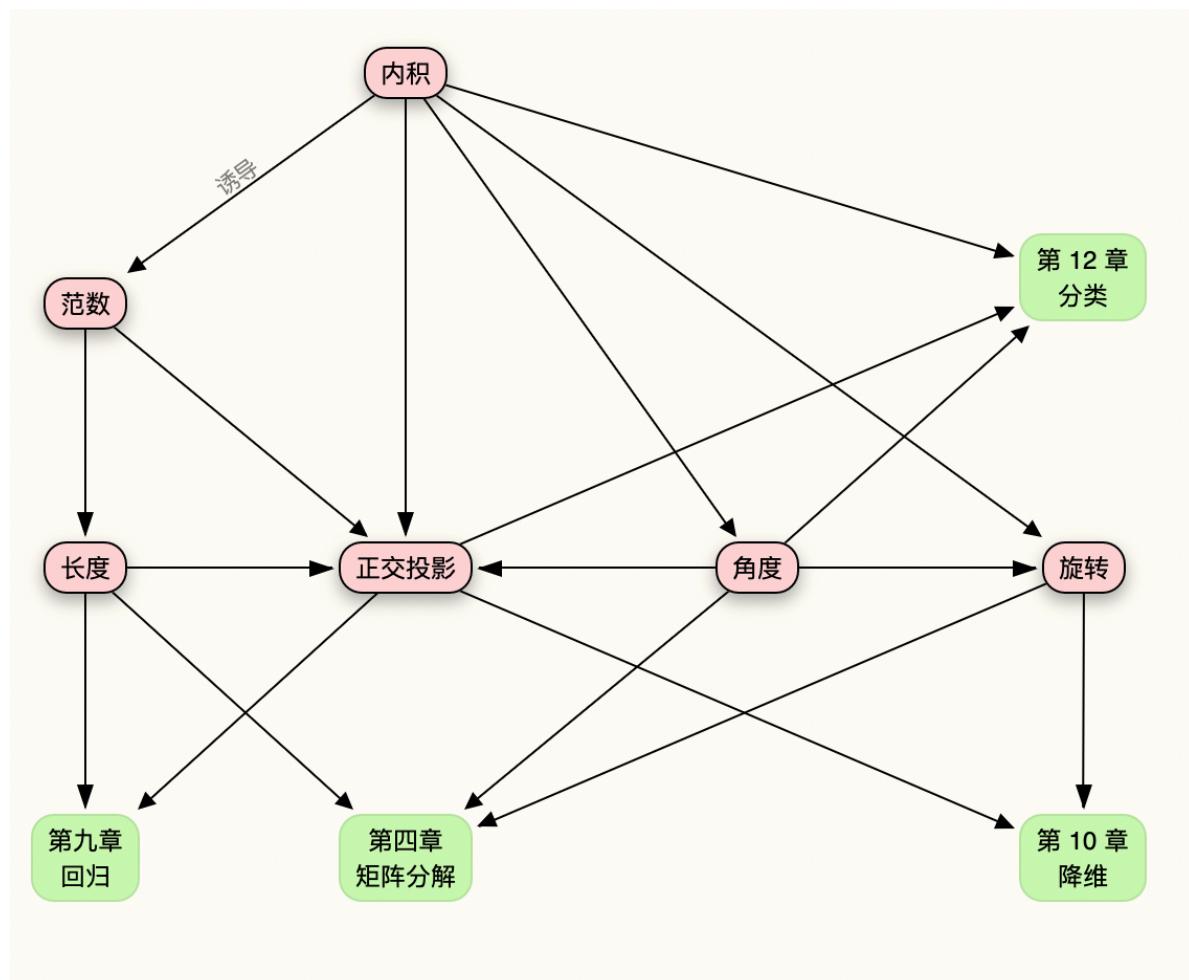


图 3.1: 本章的概念地图



< 上一章节

第二章 线性代数

下一章节 >

第四章 矩阵分解



3.1 范数

当我们考虑几何意义下的向量，也就是原点出发的有向线段时，其长度显然是原点到有向线段终点之间的直线距离。下面我们将使用范数的概念讨论向量的长度。

定义 3.1 (范数) 一个范数是线性空间 V 上的一个函数：

$$\|\cdot\| : V \rightarrow \mathbb{R} \quad (3.1)$$

$$x \mapsto \|x\|, \quad (3.2)$$

它给出每个线性空间中每个向量 x 的实值长度 $\|x\| \in \mathbb{R}$ ，使得任意的 $x, y \in V$ 以及 $\lambda \in \mathbb{R}$ ，满足下面的条件：

- (绝对一次齐次) $\|\lambda x\| = |\lambda| \|x\|$,
- (三角不等式) $\|x + y\| \leq \|x\| + \|y\|$,
- (半正定) $\|x\| \geq 0$, 当且仅当 $x = 0$ 时取等

如图 3.2 所示，在几何中，三角不等式是说任意三角形的两边之和一定大于等于第三边。

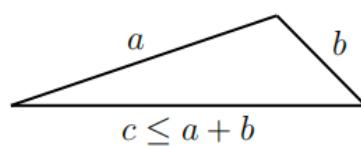


图 3.2：三角不等式的几何表示

虽然定义 3.1 考虑的是所有线性空间（2.4 节），但在本书中，我们仅考虑有限维线性空间 \mathbb{R}^n 。

最后别忘了，我们使用下标 i 表示 \mathbb{R}^n 中的向量 x 的第 i 个分量。

示例 3.1 (曼哈顿范数) \mathbb{R}^n 上的曼哈顿范数（又叫 l_1 范数）的定义如下：

$$\|x\|_1 := \sum_{i=1}^n |x_i|, \quad (3.3)$$

其中 $|\cdot|$ 是绝对值函数。图 3.3 的左侧显示了平面 \mathbb{R}^2 上所有满足 $\|x\| = 1$ 的点集。

示例 3.2 (Euclid 范数) 向量 $x \in \mathbb{R}^n$ 的* Euclid 范数* (又叫 l_2 范数) 定义如下:

$$\|x\|_2 := \sqrt{\sum_{i=1}^n x_i^2} = \sqrt{x^\top x}, \quad (3.4)$$

它计算向量 x 从原点出发到终点的 Euclid 距离 (译者注: 也就是我们通常意义下的距离)。图 3.3 的右侧显示了 \mathbb{R}^2 平面上所有满足 $\|x\|_2 = 1$ 的点集。

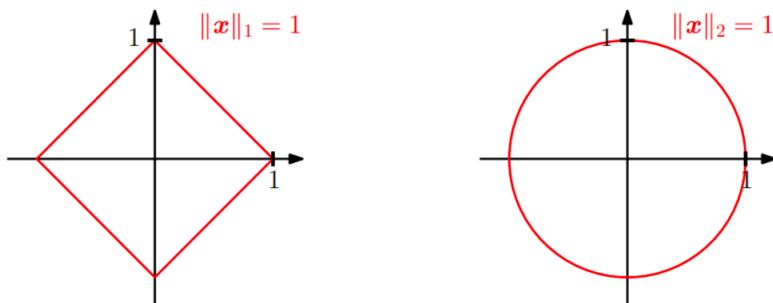


图 3.3: 平面上满足向量在不同范数的度量下值为1的情况: 左侧为曼哈顿范数, 右侧为 Euclid 范数

注: 在本书中, 若不指明, 范数一般是指 Euclid 范数 (式 3.4)。

[下一章节 >](#)

3.2 内积



3.2 内积

内积的引入是后面若干几何直觉上的概念，如向量长度、向量间夹角的铺垫。

引入内积的一个主要目的是确认两个向量是否正交。

3.2.1 点积

我们已经熟悉一些特殊形式的点积，如标量积或 \mathbb{R}^n 中的点积，由下面的式子给出：

$$x^\top y = \sum_{i=1}^n x_i y_i. \quad (3.5)$$

在本书中，我们称这样的内积形式为点积。需要注意的是，我们将介绍的内积是更加一般的概念，只要满足一些条件即可。

3.2.2 一般的点积

回忆在 2.7 节中提到的线性映射：我们可以利用其性质对加法和标量乘法进行重排。一个 V 上的双线性映射 Ω 接受两个参数，并对其中的任意一个参数保持线性（译者注：即双重线性）。任取 $x, y, z \in V$, $\lambda, \psi \in \mathbb{R}$, 我们有

$$\Omega(\lambda x + \psi y, z) = \lambda \cdot \Omega(x, z) + \psi \cdot \Omega(y, z), \quad (3.6)$$

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

在式中，(式 3.6) 表示函数对第一个变量线性；(式 3.7) 表示函数对第二个变量线性（见式 2.87）。

定义 3.2 设 V 为线性空间，双线性映射 $\Omega : V \times V \rightarrow \mathbb{R}$ 将两个 V 中的向量映射到一个实数，则

- 若对所有 $x, y \in V$, 都有 $\Omega(x, y) = \Omega(y, x)$, 也即两个变量可以调换顺序，则称 Ω 为对称的

- 若对所有 $x \in V$, 都有

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

则称 Ω 为正定的。

定义 3.3 设 V 为线性空间, 双线性映射 $\Omega : V \times V \rightarrow \mathbb{R}$ 将两个 V 中的向量映射到一个实数, 则

- 对称且正定的双线性映射 Ω 叫做 V 上的一个内积, 并简写 $\Omega(x, y)$ 为 $\langle x, y \rangle$ 。
- 二元组 $(V, \langle \cdot, \cdot \rangle)$ 称为内积空间或装配有内积的 (实) 线性空间。特别地, 如果内积采用 (式 3.5) 中定义的点积, 则称 $(V, \langle \cdot, \cdot \rangle)$ 为 Euclid 线性空间 (译者注: 简称欧氏空间)

本书中我们称这些空间为内积空间。

示例 3.3 (不是点积的内积) 考虑 $V = \mathbb{R}^2$, 定义下面的内积:

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

可以验证这是一个与点积不同的内积, 证明留作练习。

3.2.3 对称和正定矩阵

对称和正定矩阵在机器学习中十分重要, 它们是由内积定义的。在 4.3 节中, 我们在讨论矩阵分解时将会回到这个概念。在 12.4 节中, 对称和半正定矩阵还在核的定义中起到关键作用。假设 n 维线性空间 V 装配有内积 $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{R}$ (参见定义 3.3) 并取 V 中的一个基 (已排序) $B = (b_1, \dots, b_n)$, 在 2.6.1 节中我们知道任意 $x, y \in V$, 可以找到 $\lambda_i, \psi_i \in \mathbb{R}, i = 1, \dots, n$, 使得两个向量可以写成基 B 中向量



的线性组合，即 $x = \sum_{i=1}^n \psi_i b_i \in V$, $y = \sum_{j=1}^n \lambda_j b_j \in V$ 。由内积的双线性性，对所有的 $x, y \in V$, 有

$$\langle x, y \rangle = \left\langle \sum_{i=1}^n \psi_i b_i, \sum_{j=1}^n \lambda_j b_j \right\rangle = \sum_{i=1}^n \sum_{j=1}^n \psi_i \lambda_j \langle b_i, b_j \rangle = \hat{x}^\top A \hat{y},$$

其中 $A_{i,j} := \langle b_i, b_j \rangle$ (译者注：这就是线性空间 V 中的一个度量矩阵)， \hat{x} 和 \hat{y} 为原向量在基 B 下的坐标。这意味着内积 $\langle \cdot, \cdot \rangle$ 被矩阵 A 唯一确定，且由于内积具有对称性，不难看出 A 是对称矩阵。进一步地，根据内积的正定性，我们可以得出下面的结论：

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

定义 3.4 (对称正定矩阵) 一个 n 级对称矩阵 $A \in \mathbb{R}^{n \times n}$ 若满足 (式 3.11)，则叫做对称正定矩阵 (或仅称为正定矩阵)。如果只满足将 (式 3.11) 中的不等号改成 \geq 的条件，则称为对称半正定矩阵

示例 3.4 (对称正定矩阵) 考虑下面两个矩阵

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

其中 A_1 是对称且正定的，因为它不仅对称 (译者注：这显而易见)，而且对于任意 $x \in \mathbb{R}^2 - \{0\}$ 都有，

$$\begin{aligned} x^\top A_1 x &= [x_1 \ x_2] \begin{bmatrix} 9 & 6 \\ 6 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= 9x_1^2 + 12x_1x_2 + 5x_2^2 \\ &= (3x_1 + 2x_2)^2 + x_2^2 > 0. \end{aligned} \quad (3.13)$$

相反地， A_2 不是正定矩阵。如果取 $x = [2, -3]^\top$ ，可以验证二次型 $x^\top A x$ 是负数。

假设 $A \in \mathbb{R}^{n \times n}$ 是一个对称正定矩阵，则它可以定义一个在基 B 下的内积：

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

其中 $x, y \in V$ 。

定理 3.5 考虑一个有限维实线性空间 V 及它的一个基（有序） B ，双线性函数 $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{R}$ 是其上的一个内积当且仅当有一个对称正定矩阵 $A \in \mathbb{R}^{n \times n}$ ，与之对应，即

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

下面再列举两个对称正定矩阵的性质

- 矩阵 A 的零空间（核）只包含零向量，因为若 x 不为零，则有 $x^\top A x > 0$ ，于是 $Ax \neq 0$ 。
- 矩阵 A 的所有对角元 (a_{ii} , $i = 1, \dots, n$) 都是正数，因为 $a_{ii} = e_i^\top A e_i > 0$ ，其中 e 是 B 中第 i 个基向量。

< 上一章节

下一章节 >

3.1 范数

3.3 向量长度和距离



3.3 向量长度和距离 (75)

在 3.1 节中，我们已经讨论过计算向量长度需要用到的范数。内积和范数这两个概念紧密相连，因为任意的内积可以自然地诱导出一个范数

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

我们就可以使用内积计算向量的长度了。

注释 不是所有范数都是由内积诱导出来的

曼哈顿范数（式 3.3）就是一个例子。下面我们聚焦于内积诱导的范数进行讨论，并引出相关的几何直观概念，如长度、距离和夹角。

注 (柯西-施瓦兹不等式)：内积空间 $(V, \langle \cdot, \cdot \rangle)$ 中由内积 $\langle \cdot, \cdot \rangle$ 诱导的范数 $\|\cdot\|$ 满足柯西-施瓦兹不等式：

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

示例 3.5 (使用内积计算向量长度) 在几何中，我们常关心向量的长度。现在我们可以使用内积计算它们。例如取 $x = [1, 1]^\top \in \mathbb{R}^2$ ，并令其上的内积为点积，则可以得到其长度

$$\|x\| = \sqrt{x^\top x} = \sqrt{1^2 + 1^2} = \sqrt{2}. \quad (3.18)$$

现在我们考虑另一个矩阵决定的内积：

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



如果我们根据这个内积的定义进行计算范数，当 x_1 和 x_2 同号时，结果会小于内积内积诱导出的范数的值，反之则会大于它。我们可以使用 $x = [1, 1]^\top$ 进行实验，并发现它“看上去”比使用点积诱导出的范数的度量下要短：

$$\langle x, x \rangle = x_1^2 - x_1 x_2 + x_2^2 = 1 - 1 + 1 = 1 \implies \|x\| = \sqrt{1} \notin \mathbb{N}$$

定义 3.6 (距离和度量) 考虑一个内积空间 $(V, \langle \cdot, \cdot \rangle)$ ，任取向量 $x, y \in V$ ，称

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

为向量 x 和 y 之间的距离。如果我们选用点积作为 V 上的内积，则得出的距离称为* Euclid 距离（也称欧氏距离）。这样的映射

$$d : V \times V \rightarrow \mathbb{R} \quad (3.22)$$

$$(x, y) \mapsto d(x, y) \quad (3.23)$$

称为度量*。

注释 和向量长度类似，确定向量之间的距离不一定需要内积，使用范数足矣。如果我们有由内积诱导的范数，向量间的距离因选择的内积的不同而不同。

一个度量 d 满足下面三条性质：

1. (正定性) 对任意的 $x, y \in V$, $d(x, y) \geq 0$, 当且仅当 $x = y$ 时取等,
2. (对称性) 对任意的 $x, y \in V$, $d(x, y) = d(y, x)$,
3. (三角不等式) 对任意的 $x, y, z \in V$, $d(x, y) + d(y, z) \geq d(x, z)$ 。



注释 第一次看到度量的定义时，读者会发现它和内积十分相似。但如果细致比对定义 3.3 和定义 3.6，我们会发现二者的“方向”截然相反。如果两向量 $x, y \in V$ 的内积较大，则它们之间的度量较小，反之亦然。

< 上一章节

下一章节 >

3.2 内积

3.4 向量夹角和正交



3.4 向量夹角和正交 (76)

在对向量的长度和两向量之间的距离进行定义的基础上，内积还可以通过定义两向量之间的夹角 ω 以刻画线性空间中的几何特征。我们使用Cauchy-Schwarz不等式

(3.17) 定义内积空间中两个向量 x 和 y 之间的夹角 ω ，这和我们在 \mathbb{R}^2 和 \mathbb{R}^3 中的结论相同。假设两个向量均布为零，我们有

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

如图3.4所示，在 $[0, \pi]$ 中有唯一的 ω 满足下面的等式：

$$\cos \omega = \frac{\langle x, y \rangle}{\|x\|\|y\|}. \quad (3.25)$$

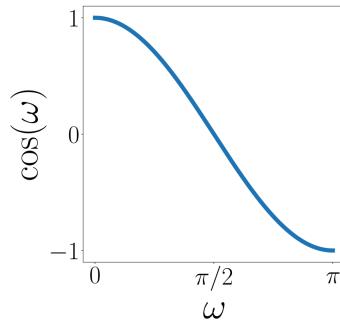


图3.4 定义域为 $[0, \pi]$ 时的余弦函数图像，此时角度值和余弦函数值在 $[-1, 1]$ 内一一对应

而 ω 就是 x 和 y 之间的夹角。直观意义上，两向量之间的夹角给出了其方向的相似程度，例如两向量 x 和 $y = 4x$ (x 经过常数缩放后的版本) 的夹角为零，因此它们的方向相同。

示例 3.6 (向量之间的夹角) 如图3.5所示，计算向量 $x = [1, 1]^\top \in \mathbb{R}^2$ 和 $y = [1, 2]^\top \in \mathbb{R}^2$ 的夹角。我们令向量的内积为点积，有

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

于是两个向量的夹角余弦值为 $\arccos\left(\frac{3}{\sqrt{10}}\right) \approx 0.32 \text{ rad}$, 大约为 18° 。

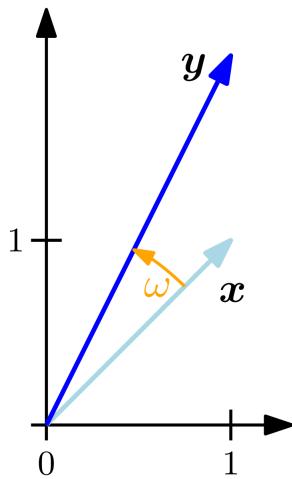


图3.5 使用向量 x 和 y 之间的内积计算它们之间的夹角 omega 。

内积的一个关键用途是判断向量之间是否正交。

定义3.7 (向量的正交) 两个向量 x 和 y 正交 (orthogonal) 当且仅当它们的内积为零, 即 $\langle x, y \rangle = 0$, 我们记为 $x \perp y$ 。进一步地, 如果 $\|x\| = \|y\| = 1$, 也即两个向量是单位向量, 则称它们 单位正交 (orthonormal)。
◦ 特殊地, 零向量 $\mathbf{0}$ 与任意向量都正交。

注: 正交性将垂直这一概念推广至通常点积之外的双线性型范畴。在我们的讨论中, 可以从几何的角度认为在某一内积下正交的两个向量的夹角为直角。

示例3.7 (单位正交向量)

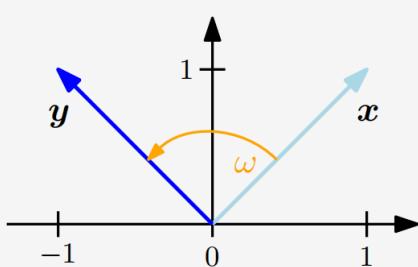


图3.6 使用不同的内积定义计算得到的两向量\$x\$和\$y\$之间的夹角不同

如图3.6所示，考虑向量 $x = [1, 1]^\top, y = [-1, 1]^\top \in \mathbb{R}^2$ ，考虑它们在不同内积定义下的夹角大小。如果使用通常的点积作为内积，则它们之间的夹角为 90° ，也即 $x \perp y$ 。但如果使用下面的内积定义则会得到不同的结果：

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

可以计算得到在这个内积之下两向量之间的夹角为

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

于是 x 和 y 并不正交。因此在一个内积下正交的两个向量在另一个内积下不一定正交。

定义3.8 (正交矩阵) 方阵 $A \in \mathbb{R}^{n \times n}$ 为正交矩阵当且仅当满足下面的条件：

$$AA^\top = I = A^\top A, \quad (3.29)$$

进而有

$$A^{-1} = A^\top, \quad (3.30)$$

这是说，正交矩阵的逆是它的转置。

注：一般我们将这些矩阵称为“orthogonal matrix”，严格意义上它们应该叫做“orthonormal matrix”。因为“orthonormal matrix”对应的变换在线性空间内保持向量的长度和向量之间的夹角。译者注：中文中不做区分，统一称为“正交矩阵”。正交矩阵对应的变换在 \mathbb{R}^2 和 \mathbb{R}^3 中属于刚体变换。

正交矩阵作用在向量 x 上不改变它的长度。当范数为向量点积作为内积诱导的范数时，我们有

$$\|Ax\|^2 = (Ax)^\top (Ax) = x^\top A^\top Ax = x^\top Ix = x^\top x = \|x\|^2. \quad (3.31)$$

进一步地，两个向量 x 和 y 之间使用内积度量的夹角在同时被正交矩阵作用后依然保持不变。假设内积依然为点积， Ax 和 Ay 之间的夹角余弦值为

$$\cos \omega = \frac{(Ax)^\top (Ay)}{\|Ax\| \|Ay\|} = \frac{x^\top A^\top Ay}{\sqrt{x^\top A^\top A x y^\top A^\top A y}} = \frac{x^\top y}{\|x\| \|y\|}, \quad (3.32)$$

以上内容表示，正交矩阵对应的线性变换同时保持长度和夹角。事实上，这些正交矩阵定义了一系列的旋转和翻转。在章节3.9中我们会进一步讨论它们。

< 上一章节

下一章节 >

3.3 向量长度和距离

3.5 正交基



3.5 正交基

在2.6.1节中，我们讨论了基向量的性质，我们发现在 n 维空间中，我们需要 n 个基向量（也就是 n 个线性无关的向量）。在3.3和3.4两节中，我们使用内积计算向量的长度和向量之间的夹角。在本节中，我们将讨论基向量互相垂直且长度为1这一特殊情况，我们称其为**正交基**。

我们不妨使用更加严谨的语言介绍它们：

定义 3.9 (正交基) 考虑一个 n 维线性空间 V 和它上面的一个基 $\{b_1, \dots, b_n\}$ ，如果

$$\langle b_i, b_j \rangle = 0, \quad i \neq j \tag{3.33}$$

$$\langle b_i, b_i \rangle = 1 \tag{3.34}$$

对于所有的 $i, j = 1, \dots, n$ 都成立，那么 $\{b_1, \dots, b_n\}$ 就叫做**标准正交基 (orthonormal basis, ONB)**，注意所有的向量的长度均为1。假如这个基只满足(3.33)，则它就叫做**正交基 (orthogonal basis)**。

让我们回忆一下，在2.6.1节中我们使用Gauss消元法寻找一个向量组张成空间的基的过程。假设我们有一个未标准化 (unnormalized) 且非正交的向量组 $\{\tilde{b}_1, \dots, \tilde{b}_n\}$ ，我们将其堆叠成一个矩阵 $\tilde{B} = [\tilde{b}_1, \dots, \tilde{b}_n]$ ，然后在增广矩阵 $[\tilde{B} \tilde{B}^\top | \tilde{B}]$ 上应用Gauss消元法，就可以得到一个标准正交基。像这样迭代地构造正交基 $\{b_1, \dots, b_n\}$ 的方法叫做**Gram-Schmidt正交化过程**。

示例 3.8 (正交基) Euclid空间 \mathbb{R}^n 上的标准基是标准正交基，其中内积为两个向量的点积。特别地，在 \mathbb{R}^2 中，两个向量

$$b_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad b_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{3.35}$$

组成一个正交基，因为 $b_1^\top b_2 = 0$ 且 $\|b_1\| = \|b_2\| = 1$ 。



我们将在第十章和第十二章介绍支持向量机和主成分分析时深入讲解标准正交基这一概念。

< 上一章节

下一章节 >

3.4 向量夹角和正交

3.6 正交补



3.6 正交补

我们在定义了正交这一概念之后可以来看看互相正交的线性空间了。这样的线性空间在第十章讨论线性降维的几何视角时十分重要。

考虑一个 D 维的线性空间 V 和一个 M 维的子空间 $U \subset V$, U 的正交补 (**orthogonal complement**) U^\perp 是一个 $(D - M)$ 维的子空间, 其中的任何向量都与 U 中的任何向量垂直。进一步我们有 $U \cap U^\perp = \{0\}$, 于是 V 中的任何向量 x 可以被唯一分解为下面的形式:

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

其中 (b_1, \dots, b_M) 是 U 的一个基, $(b_1^\perp, \dots, b_{D-M}^\perp)$ 是 U^\perp 的一个基。

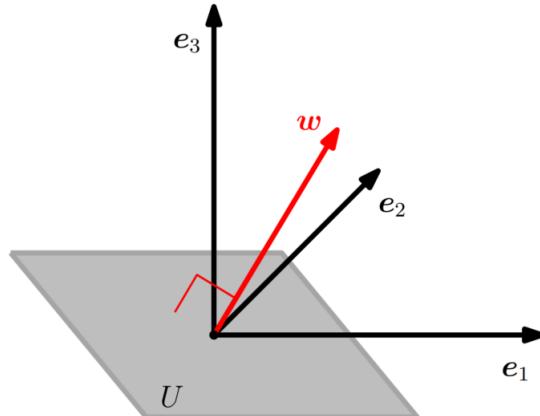


图3.7 三维线性空间的平面可被与其垂直单位向量唯一确定, 后者是其正交补空间的基

因此, 三维线性空间中的平面 U 的正交补也可以用来描述平面本身 (平面是二维的)。具体来说, 三维空间的一个向量 w 如果满足 $\|w\| = 1$, 是某个平面 U 的正交补空间 U^\perp 的一个基, 如图3.7所示。在图中, 所有与 w 垂直的向量一定在平面 U 中, 故 w 也被称作平面 U 的法向量 (**normal vector**)

一般地, 正交补空间可被用来刻画 n 维线性空间和仿射空间 (**affine space**, 译者注: 也称线性流形) 中的超平面 (**hyperplanes**)

< 上一章节

下一章节 >

3.5 正交基

3.7 函数的内积





3.7 函数的内积

到现在我们了解了内积的各种性质，并利用它们计算有限维向量的长度、夹角和距离。在本节中，我们将看到另一种向量之间的内积：函数的内积。

到此为止我们讨论的所有内积都定义在具有有限个分量的向量之上。我们可以将向量 $x \in \mathbb{R}^n$ 视作有 n 个取值的函数，这样内积的概念可以推广至具有无限个分量（可数无穷）以及连续（不可数无穷）的向量之上。在这样的意义下，原来对不同向量分量的（乘积后）的加和（例如式(3.5)）将变为积分。

两个函数 $u : \mathbb{R} \rightarrow \mathbb{R}$ 和 $v : \mathbb{R} \rightarrow \mathbb{R}$ 之间的内积可被定义为下面的定积分：

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

其中积分限满足 $a, b < \infty$ 。

和通常的内积一样，我们也可以通过内积定义函数的范数和正交关系。如果式(3.37)的结果为零，则两个函数 u 和 v 相互正交。如果需要给出更加严格的定义，我们需要考虑测度和积分定义的方式，这将引出 Hilbert 空间。进一步地，与有限维向量之间的内积不同，函数之间的内积可能发散（值为无穷大）。对上述情形的讨论涉及实分析和泛函分析中的细节，不是本书讨论的内容。

示例 3.9 (函数之间的内积) 假如我们令 $u = \sin(x)$, $v = \cos(x)$, 则内积定义(3.37)中的被积函数为 $f = u(x)v(x)$, 如图3.38所示。我们发现这个函数是奇函数，也即 $f(-x) = -f(x)$ 。所以积分限为 $a = -\pi, b = \pi$ 的定积分的值为零，因此我们可以得到 \sin 和 \cos 互相正交的结论。

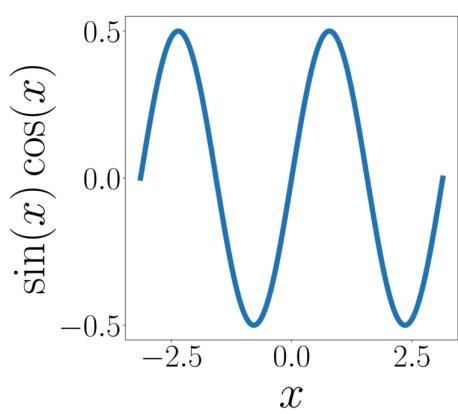


图3.8 被积函数 $f(x) = \sin(x)\cos(x)$ 的图像

注释 上述结论对于下面的函数族依然成立：

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

(如果将积分限设置为 $-\pi$ 和 π)。换句话说，这个函数族中的函数两两正交，它们张成的巨大空间是所有以区间 $[-\pi, \pi]$ 为周期的连续函数。将函数向这个子空间上投影是**Fourier级数**的核心思想。

在6.4.6节，我们还会遇见第二种不常见的内积——随机变量之间的内积。

< 上一章节

3.6 正交补

下一章节 >

3.8 正交投影



3.8 正交投影

投影是一类重要的线性变换（其他重要的线性变换还有旋转和反射），在图形学、编码理论、统计学和机器学习中占有重要的地位。在机器学习中，我们经常需要与高维数据打交道，它们往往难以进行分析和可视化。然而，高维数据往往具有大部分信息被包含在仅仅几个维度之中，其他维度对于数据关键信息的刻画并不重要的特点。当我们对高维数据进行压缩或可视化时，我们将失去一些信息。为了将压缩造成的信息损失最小化，我们往往选择数据中最关键的几个维度。

我们在第一章中提到，数据可被表示成向量。在本章中，我们将对基础的数据压缩方法进行讨论。具体而言，我们可以将原来的高维数据投影到低维**特征空间（feature space）**，然后在此空间中对数据进行处理和分析，以更好的了解数据集并抽取相关的**模式（pattern）**。

注释 “特征”是数据表示的一个常见说法。

举例来说，以主成分分析（principal component analysis, PCA）为例的机器学习算法（Pearson, 1901 和 Hotelling, 1933）以及以自编码器（auto-encoders, Deng et al., 2010）深度神经网络充分利用了降维的思想。接下来，我们将将注意力集中在第十章将被使用于线性降维和在十二章中的分类问题中的正交投影上。

即使是我们将在第九章中讨论的线性回归算法，也可以从正交投影的角度进行解读。给定一个低维子空间，来自高维空间中数据的正交投影会保留尽可能多的信息，并最小化元数据和投影数据的区别或损失。

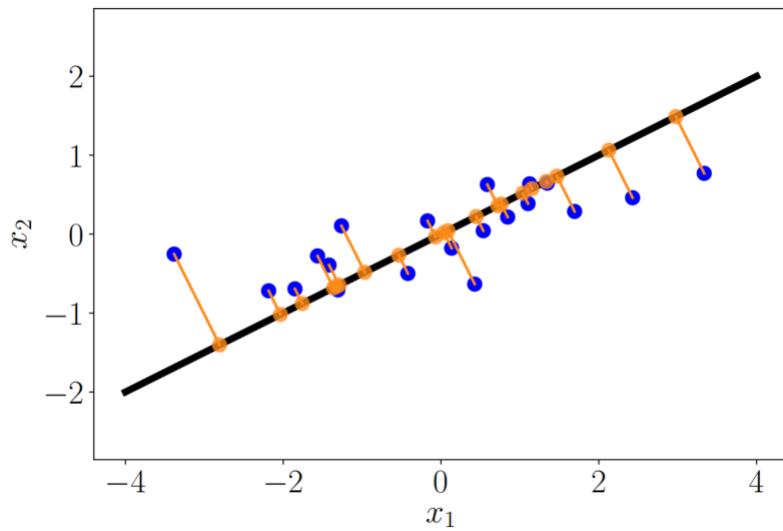


图 3.9 二维数据点（蓝色点）至一维子空间（直线）的正交投影（橙色点）

正交投影的直观几何描述可见图 3.9。在我们介绍细节之前，需要首先定义投影这个概念。

定义 3.10（投影） 令 V 为一个线性空间， $U \subset V$ 是 V 的子空间，如果一个线性映射 $\pi : V \rightarrow U$ 满足 $\pi^2 = \pi \circ \pi = \pi$ ，则称 π 为一个**投影**（projection）。

由于线性映射可以表示为矩阵（参见 2.7 节），上面的定义等价于确定了一类特殊的矩阵变换 P_π ，它们满足 $P_\pi^2 = P_\pi$ 。在接下来的内容中将推导内积空间 $(\mathbb{R}^n, \langle \cdot, \cdot \rangle)$ 中向量至其子空间的正交投影，我们将从一维子空间（也称为直线）开始。如果没有特殊说明，我们约定向量的内积为点积，即 $\langle x, y \rangle = x^\top y$ 。

3.8.1 向一维子空间（直线）投影

假设给定一条通过原点的直线（一维子空间），和该空间的一个基 $b \in \mathbb{R}^n$ 。这条直线是 b 张成的子空间 $U \subset \mathbb{R}^n$ 。当我们向量 $x \in \mathbb{R}^n$ 投影至 U 中时，我们需要在 U 中寻找距离 x 最近的向量 $\pi_U(x) \in U$ 。下面列举一些投影向量 $\pi_U(x)$ 的性质（参考图 3.10）

- 投影向量 $\pi_U(x)$ 是（子空间中）距离 x 最近的向量，“最近”的意思是距离 $\|x - \pi_U(x)\|$ 是最小的。这表示从 $\pi_U(x)$ 到 x 的线段 $\pi_U(x) - x$ 与 U 是垂直的，



也和 U 的基 b 垂直。

- x 到 U 的投影向量 $\pi_U(x)$ 一定是 U 中的元素，因此也和 U 的基 b 共线。于是存在 $\lambda \in \mathbb{R}$ ，使得 $\pi_U(x) = \lambda b$ 。

注释 λ 是 $\pi_U(x)$ 在基 b 下的坐标。

下面我们将通过三个步骤确定坐标 λ ，投影向量 $\pi_U(x) \in U$ ，以及将 $x \in \mathbb{R}^n$ 投影至子空间 U 的投影矩阵 P_π 。

1. 计算坐标 λ 的值。由正交性条件得到

$$\langle x - \pi_U(x), b \rangle = 0 \stackrel{\pi_U(x) = \lambda b}{\iff} \langle x - \lambda b, b \rangle = 0. \quad (3.39)$$

我们可以利用内积的双线性性，得到

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

注释 若使用一般的内积，如果 $\|b\| = 1$ ，我们有 $\lambda = \langle x, b \rangle$ 。

最后，我们利用内积的对称性对原式进行变换。如果我们令 $\langle \cdot, \cdot \rangle$ 为点积，我们就可以得到

$$\lambda = \frac{b^\top x}{b^\top b} = \frac{b^\top x}{\|b\|^2}. \quad (3.41)$$

如果 $\|b\| = 1$, 则 λ 的值为 $b^\top x$ 。



2. 计算投影点 $\pi_U(\mathbf{x}) \in U$ 。由于 $\pi_U(\mathbf{x}) = \lambda b$, 由 (3.40), 立刻有

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

其中最后的等号成立条件为内积取为点积。我们还可以根据定义3.1计算 $\pi_U(x)$ 的长度:

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

因此, 投影向量的长度为 $|\lambda|$ 乘以 b 的长度。这也增加了一个直观理解方式: λ 是投影向量在子空间 U 的基 b 下的坐标。如果我们的内积是点积, 就有

$$\begin{aligned} \|\pi_U(\mathbf{x})\| &\stackrel{(3.42)}{=} \frac{|b^\top x|}{\|b\|^2} \|b\| \stackrel{(3.25)}{=} |\cos \omega| \cdot \|x\| \cdot \|b\| \cdot \frac{\|b\|}{\|b\|^2} \\ &= |\cos \omega| \cdot \|x\|. \end{aligned} \quad (3.44)$$

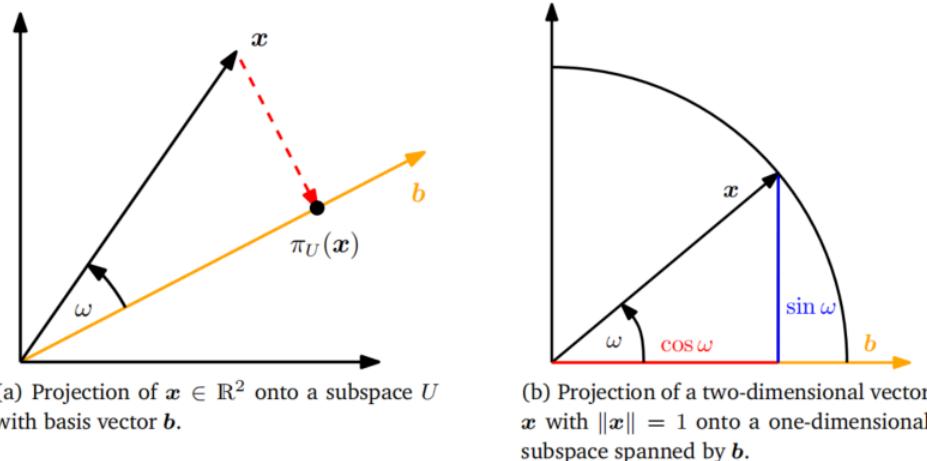


图 3.10 投影至一位子空间的示例。

这里的 ω 是向量 x 和 b 之间的夹角。如图3.10所示, 从三角学的角度看, 该结果是似曾相识的: 如果 $\|\mathbf{x}\| = 1$, 则向量 \mathbf{x} 的终点位于单位圆上。接着可以得到 \mathbf{x} 向横轴的投影在基 b 下的坐标恰好就是 $\cos \omega$, 投影向量的长度也满足 $|\pi_U(\mathbf{x})| = |\cos \omega|$ 。



注释 所谓的横轴就是一个一维子空间。

3. 计算投影矩阵 \mathbf{P}_π 。通过定义 3.10 我们知道投影是一个线性变换。因此存在一个投影矩阵 \mathbf{P}_π , 使得 $\pi_U(\mathbf{x}) = \mathbf{P}_\pi \mathbf{x}$ 。若令点积为内积, 我们有

$$\pi_U(\mathbf{x}) = \lambda b = b\lambda = b \frac{b^\top x}{\|b\|^2} = \frac{bb^\top}{\|b\|^2} x, \quad (3.45)$$

这样立刻得到

$$\mathbf{P}_\pi = \frac{bb^\top}{\|b\|^2}. \quad (3.46)$$

注意 bb^\top (也就是 \mathbf{P}_π) 是秩为 1 的对称矩阵, 而 $\|b\|^2 = \langle b, b \rangle$ 是一个标量。

投影矩阵 \mathbf{P}_π 将任意向量 $\mathbf{x} \in \mathbb{R}^n$ 投影到通过原点, 方向为 b 的直线上 (这等价于由 b 张成的子空间 U)。

注释 投影向量 $\pi_U(\mathbf{x}) \in \mathbb{R}^n$ 依然是一个 n 维向量, 不是一个标量。然而, 我们不再需要使用 n 个分量来描述它——我们只需要使用一个分量 λ , 因为这是投影向量关于子空间 U 中的基 b 的坐标。

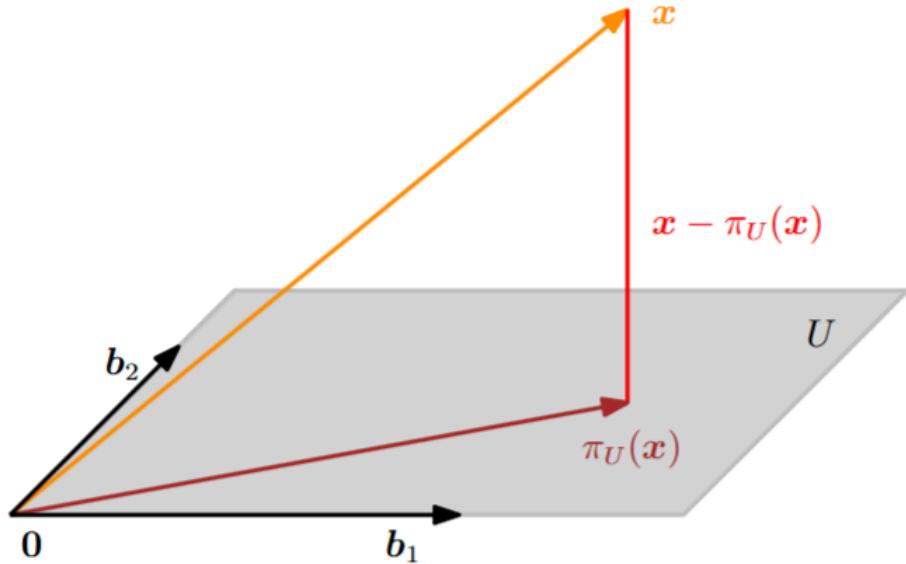


图 3.12 向二维子空间投影

示例 3.10 (向直线投影) 求投影至通过原点, 由向量 $b = [1, 2, 2]^\top$ 张成直线的投影矩阵 \mathbf{P}_π , 其中 b 是该过原点直线的方向, 也就是一维子空间的基。

通过 (3.46), 我们有

$$\mathbf{P}_\pi = \frac{bb^\top}{b^\top b} = \frac{1}{9} \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} [1, 2, 2] = \frac{1}{9} \begin{bmatrix} 1 & 2 & 2 \\ 2 & 4 & 4 \\ 2 & 4 & 4 \end{bmatrix}. \quad (3.47)$$

现在我们选一个特定的向量 x , 然后检查它的投影是否在这条直线上。不妨令 $x = [1, 1, 1]^\top$, 然后计算它的投影:

$$\pi_U(x) = \mathbf{P}_\pi(x) = \frac{1}{9} \begin{bmatrix} 1 & 2 & 2 \\ 2 & 4 & 4 \\ 2 & 4 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 5 \\ 10 \\ 10 \end{bmatrix} \in \text{span} \left\{ \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} \right\} \quad (3.48)$$

注意, \mathbf{P}_π 作用在 $\pi_U(x)$ 上的结果等于它本身, 这是说 $\mathbf{P}_\pi \pi_U(x) = \pi_U(x)$ 。这并不令我们以外, 因为根据定义 3.10, 我们知道 \mathbf{P}_π 是幂等的, 也即对于任意的 x , 有 $\mathbf{P}_\pi^2 x = \mathbf{P}_\pi$ 。



注释 在第四章，我们将证明 $\pi_U(\mathbf{x})$ 是矩阵 \mathbf{P}_π 的一个特征向量，对应的特征值为 1 。

3.8.2 向一般子空间投影

接下来我们讨论将向量 $x \in \mathbb{R}^n$ 投影至较低维度的一般子空间 $U \subset \mathbb{R}^n$ ，其中 U 满足 $\dim U = m \geq 1$ 。如图 3.11 所示。

假设 (b_1, \dots, b_m) 是 U 的一个有序基， U 上的任意投影向量 $\pi_U(\mathbf{x})$ 必定是它的元素，因此 U 中存在基向量 b_1, \dots, b_m 的一个线性组合，满足 $\pi_U(\mathbf{x}) = \sum_{i=1}^m \lambda_i b_i$ 。

注意，如果子空间 U 是通过由一些向量张成的空间而给出的，读者在进行下面的计算之前需要确定其上的一个正交基 b_1, \dots, b_m 。

和前文中投影至一维子空间类似，我们按照下面三步就可以找到投影向量 $\pi_U(\mathbf{x})$ 和投影矩阵 \mathbf{P}_π 。

1. 确定投影向量在 U 上的基下的坐标 $\lambda_1, \dots, \lambda_m$ ，使得下面的线性组合距离 $x \in \mathbb{R}^n$ 是最近的。

$$\pi_U(\mathbf{x}) = \sum_{i=1}^m \lambda_i b_i = \mathbf{B}\boldsymbol{\lambda}, \quad (3.49)$$

$$\mathbf{B} = [b_1, \dots, b_m] \in \mathbb{R}^{n \times m}, \boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_m]^\top \in \mathbb{R}^m \quad (3.50)$$

和一维的例子一样，“最近”表示距离最短，这可以推断出连接 $\pi_U(\mathbf{x})$ 和 x 的向量一定与 U 的所有基向量都垂直（假设此时的内积是点积）。



$$\langle b_1, x - \pi_U(\mathbf{x}) \rangle = b_1^\top (x - \pi_U(\mathbf{x})) = 0,$$

$$\vdots \quad (1)$$

$$\langle b_m, x - \pi_U(\mathbf{x}) \rangle = b_m^\top (x - \pi_U(\mathbf{x})) = 0, \quad (3.52)$$

依据 $\pi_U(\mathbf{x}) = \mathbf{B}\boldsymbol{\lambda}$ 对上式进行替换，有

$$b_1^\top (x - \mathbf{B}\boldsymbol{\lambda}) = 0, \quad (3.53)$$

$$\vdots \quad (2)$$

$$b_m^\top (x - \mathbf{B}\boldsymbol{\lambda}) = 0, \quad (3.53)$$

这样我们就得到了一个齐次线性方程

$$\begin{bmatrix} b_1^\top \\ \vdots \\ b_m^\top \end{bmatrix} (x - \mathbf{B}\boldsymbol{\lambda}) \iff \mathbf{B}^\top (x - \mathbf{B}\boldsymbol{\lambda}) = 0 \quad (3.55)$$

$$\iff \mathbf{B}^\top \mathbf{B}\boldsymbol{\lambda} = \mathbf{B}^\top x \quad (3.56)$$

最后得到的方程 (3.56) 叫正规方程 (**normal equation**)。由于 b_1, \dots, b_m 是 U 的一个基，因此它们线性无关，所以矩阵 $\mathbf{B}^\top \mathbf{B} \in \mathbb{R}^{m \times m}$ 是正规矩阵，存在逆矩阵。所以我们可以求得解析解

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

其中 $(\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top$ 叫矩阵 \mathbf{B} 的伪逆 (**pseudo-inverse**)，这对不是方形的矩阵也有效，唯一的要求就是 $\mathbf{B}^\top \mathbf{B}$ 是正定的，这表示 \mathbf{B} 为列满秩 (**full column rank**)。在实际操作中，我们常常对 $\mathbf{B}^\top \mathbf{B}$ 添加一个摄动项 (**jitter term**) $\varepsilon \mathbf{I}$, ($\varepsilon > 0$) 来满足其正定性和数值稳定性。这一对角线上的“山脊”将在第九章中使用 Bayesian 推断严格推导。

译者注：揭示正规矩阵和摄动后的正规矩阵的正定性是显而易见的。任取 $x \in \mathbb{R}^m$ ，构造二次型 $x^\top \mathbf{B}^\top \mathbf{B} x$ ，立刻有 $x^\top \mathbf{B}^\top \mathbf{B} x = \|Bx\|_2^2 \geq 0$ ，由范数的正定性知 $\mathbf{B}^\top \mathbf{B}$ 正定，因此满秩。对于摄动的情况，类似有 $x^\top (\mathbf{B}^\top \mathbf{B} + \varepsilon \mathbf{I}) x = \|Bx\|_2^2 + \varepsilon \|x\|_2^2 > 0$ ，可知二次型严格大于零，因此摄动后的矩阵必然正定（满秩）。

2. 计算投影向量 $\pi_U(\mathbf{x}) \in U$ 。由于我们已经得到 $\pi_U(\mathbf{x}) = \mathbf{B}\boldsymbol{\lambda}$, 因此由 (3.57), 有

$$\pi_U(\mathbf{x}) = \mathbf{B}(\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{x}. \quad (3.58)$$

3. 计算投影矩阵 \mathbf{P}_π , 从 (3.58) 中我们可以立刻看出方程的解:

$$\mathbf{P}_\pi = \mathbf{B}(\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top. \quad (3.59)$$

注: 上面对于至一般子空间的投影包含了一维的特殊情形。如果 $\dim U = 1$, 则 $\mathbf{B}^\top \mathbf{B} \in \mathbb{R}$ 是一个标量, (3.59) 可以被重写成 $\mathbf{P}_\pi = \frac{\mathbf{B}\mathbf{B}^\top}{\mathbf{B}^\top \mathbf{B}}$, 这和 (3.46) 中的矩阵完全一致。

示例 3.11 (向二维子空间投影) 对于子空间 $U =$

$\text{span} \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \right\} \subset \mathbb{R}^3$ 和向量 $x = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix} \in \mathbb{R}^3$, 找到 x 投影在 U 中的坐标 $\boldsymbol{\lambda}$, 投影向量 $\pi_U(\mathbf{x})$ 以及投影矩阵 \mathbf{P}_π

首先, 我们检查张成 U 的两个向量, 发现它们线性无关, 于是可以写成一个

矩阵 $\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}$ 。然后我们计算正规矩阵和 \mathbf{x} 对两个向量的点积:

$$\begin{aligned} \mathbf{B}^\top \mathbf{B} &= \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 3 & 5 \end{bmatrix}, \\ \mathbf{B}^\top \mathbf{x} &= \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \end{bmatrix}. \end{aligned} \quad (3.60)$$

第三步, 我们解正规方程 $\mathbf{B}^\top \mathbf{B}\boldsymbol{\lambda} = \mathbf{B}^\top \mathbf{x}$ 得到 $\boldsymbol{\lambda}$:

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



这样依赖，向量 \mathbf{x} 投影至子空间 U 的投影向量 $\pi_U(\mathbf{x})$ ，也就是向矩阵 \mathbf{B} 的列空间投影的向量可以按下式直接进行计算：

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

将原来的向量与投影后的向量作差得到向量的长度就是**投影损失** (**projection error**)：

$$\|\mathbf{x} - \pi_U(\mathbf{x})\| = \left\| [1, -2, 1]^\top \right\| = \sqrt{6}. \quad (3.63)$$

相应地，对于任意 $\mathbf{x} \in \mathbb{R}^3$ 的投影矩阵 \mathbf{P}_π 由下式给出：

$$\mathbf{P}_\pi = \mathbf{B}(\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top = -\frac{1}{6} \begin{bmatrix} 5 & 2 & -1 \\ 2 & 2 & 2 \\ -1 & 2 & 5 \end{bmatrix}. \quad (3.64)$$

我们可以通过验证残差向量 $\mathbf{x} - \pi_U(\mathbf{x})$ 是否和所有 U 的基垂直并考察 $\mathbf{P}_\pi^2 = \mathbf{P}_\pi$ (参见定义 3.10) 是否成立来验证计算结果的正确性。

注1：投影向量 $\pi_U(\mathbf{x})$ 虽然在子空间 $U \subset \mathbb{R}^m$ 中，但它依然是 \mathbb{R}^n 中的向量。但我们只需用 U 中关于基向量 $\mathbf{b}_1, \dots, \mathbf{b}_m$ 的坐标 $\lambda_1, \dots, \lambda_m$ 来表示它就足够了。

注2：在使用一般内积定义的线性空间中，我们在通过内积计算向量之间的夹角和距离是需要额外注意。

投影可以让我们对无解的线性系统 $\mathbf{Ax} = \mathbf{b}$ 进行研究。让我们回忆 \mathbf{b} 不在 \mathbf{A} 张成的空间，也就是 \mathbf{A} 所有列张成的空间（列空间）中的情形。在给出这样一个无解的线性系统时，我们可以找到一个**近似解**，也就是 \mathbf{A} 的列空间中最接近 \mathbf{b} 的向量。换句话说，我们计算 \mathbf{b} 到 \mathbf{A} 的列空间的投影，就是所求的近似解。这种问题在实作中非常常见，其得到的结果叫做超定系统 (over-determined system) 的**最小二乘估计**

(least-squares solution) , 类似地问题将在 9.4 节中继续讨论。如果再引入重构成损失 (reconstruction error) , 就构成了推导主成分分析 (10.3 节) 的一种方式。

注: 前文中我们只要求 $\{\mathbf{b}_1, \dots, \mathbf{b}_k\}$ 是子空间 U 的一个基, 如果它是标准正交基, 则 (3.33) 和 (3.34) 可以被用来化简 (3.58)。由于 $\mathbf{B}^\top \mathbf{B} = \mathbf{I}$, 我们可以得到下面更加简洁的投影表达式:

$$\pi_U(\mathbf{x}) = \mathbf{B}\mathbf{B}^\top \mathbf{x} \quad (3.65)$$

以及坐标 $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda} = \mathbf{B}^\top \mathbf{x}. \quad (3.66)$$

这意味着我们不再需要进行耗时的求逆计算了。

3.8.3 Gram-Schmidt 正交化

投影是 Gram-Schmidt 正交化的核心, 后者让我们可以从任意的 n 维线性空间 V 的一个基 $(\mathbf{b}_1, \dots, \mathbf{b}_n)$ 构造出该空间的一个标准正交基 $(\mathbf{u}_1, \dots, \mathbf{u}_n)$ 。这个正交基总是存在, 且满足 $\text{span}\{\mathbf{b}_1, \dots, \mathbf{b}_n\} = \text{span}\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ 。所谓的 Gram-Schmidt 正交化方法在给定 V 的任意基 $(\mathbf{b}_1, \dots, \mathbf{b}_n)$ 的情况下迭代地构造出正交基 $(\mathbf{u}_1, \dots, \mathbf{u}_n)$, 其过程如下:

$$\mathbf{u}_1 := \mathbf{b}_1, \quad (3.67)$$

$$\mathbf{u}_k := \mathbf{b}_k - \pi_{\text{span}\{\mathbf{u}_1, \dots, \mathbf{u}_{k-1}\}}(\mathbf{b}_k), \quad k = 2, \dots, n. \quad (3)$$

在式 (3.68) 中, 第 k 个基向量 \mathbf{b}_k 被投影至前 $k-1$ 个构造得到的单位正交向量 $\mathbf{u}_1, \dots, \mathbf{u}_{k-1}$ 张成的子空间上 (参见 3.8.2 节)。向量 \mathbf{b}_k 减去这个投影向量所得的向量 \mathbf{u}_k 与 $\mathbf{u}_1, \dots, \mathbf{u}_{k-1}$ 张成的 $k-1$ 维子空间垂直。对所有 n 个基向量 $\mathbf{b}_1, \dots, \mathbf{b}_n$ 逐个应用这个算法, 就得到了空间 V 的一个正交基 $(\mathbf{u}_1, \dots, \mathbf{u}_n)$ 。如果我们将正交基中的向量全部标准化, 使得对所有的 $k = 1, \dots, n$ 都有 $\|\mathbf{u}_k\| = 1$, 我们就得到了原空间的一个标准正交基 (ONB)。

示例 3.12 (Gram-Schmidt 正交化)

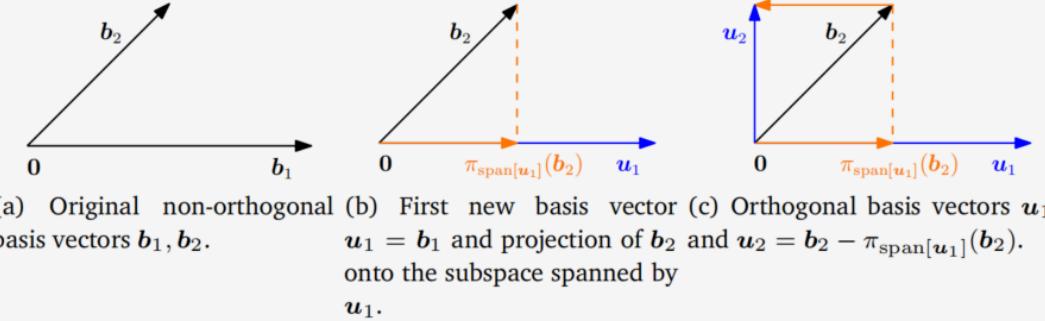


图 3.12 Gram-Schmidt 正交化

如图 3.12 所示, 考虑 \mathbb{R}^2 的一个基 $(\mathbf{b}_1, \mathbf{b}_2)$, 其中

$$\mathbf{b}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \quad (3.69)$$

使用 Gram-Schmidt 正交化方法, 我们可按照下面的过程构造 \mathbb{R}^2 的一个正交基:

$$\mathbf{u}_1 = \mathbf{b}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad (3.70)$$

$$\mathbf{u}_2 = \mathbf{b}_2 - \pi_{\text{span}\{\mathbf{u}_1\}}(\mathbf{b}_2) \quad (4)$$

$$\stackrel{(3.45)}{=} \mathbf{b}_2 - \frac{\mathbf{u}_1 \mathbf{u}_1^\top}{\|\mathbf{u}_1\|^2} \cdot \mathbf{b}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (3.71)$$

上面的步骤对应图 3.12 中的 (b) 和 (c)。我们可以立即看出 \mathbf{u}_1 和 \mathbf{u}_2 是垂直的, 也即 $\mathbf{u}_1^\top \mathbf{u}_2 = 0$ 。

3.8.4 向仿射子空间投影

直到现在我们讨论的都是如何讲一个向量投影到低维的子空间 U 上。本节将讨论如何解决投影至仿射子空间的问题。

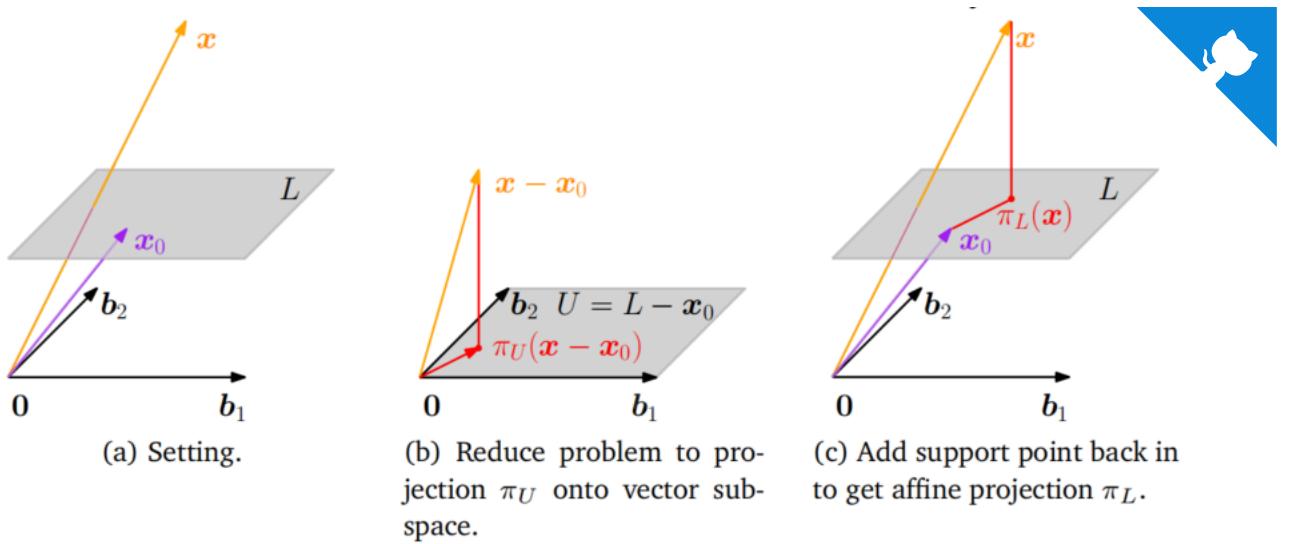


图 3.13 向仿射空间投影

考虑像图 3.13(a) 这样的问题：给定一个仿射空间 $L = \mathbf{x}_0 + U$ ，其中 $\mathbf{b}_1, \mathbf{b}_2$ 是 U 的一个基。为确定向量 \mathbf{x} 到仿射空间 L 的投影 $\pi_L(\mathbf{x})$ ，我们选择将其转换为我们已解决的投影至低维子空间的问题。我们对 \mathbf{x} 和 L 同时减去支持点 \mathbf{x}_0 ，这样一来 $L - \mathbf{x}_0$ 恰好就是子空间 U 。这样我们可以使用前文中 3.8.2 节讨论的正交投影至子空间的方法得到 $\pi_U(\mathbf{x} - \mathbf{x}_0)$ （如图 3.13 (b) 所示），然后我们可以把 \mathbf{x}_0 加回投影向量，将它重新放入 L 中，这样我们就得到了 \mathbf{x} 到 L 的投影：

$$\pi_L(\mathbf{x}) = \mathbf{x}_0 + \pi_U(\mathbf{x} - \mathbf{x}_0), \quad (3.72)$$

其中 $\pi_U(\cdot)$ 是至子空间 U 的投影，也就是 L 的方向空间（如图3.13所示）。

从图可以看出，从 \mathbf{x} 到 L 的距离和 $\mathbf{x} - \mathbf{x}_0$ 到 U 的距离相等，也就是

$$d(\mathbf{x}, L) = \|\mathbf{x} - \pi_L(\mathbf{x})\| = \|\mathbf{x} - [\mathbf{x}_0 + \pi_U(\mathbf{x} - \mathbf{x}_0)]\| \quad (3.73a)$$

$$= d(\mathbf{x} - \mathbf{x}_0, \pi_U(\mathbf{x} - \mathbf{x}_0)) = d(\mathbf{x} - \mathbf{x}_0, U). \quad (3.73b)$$

在 12.1 节，我们将会用这个方法导出分割超平面这个概念。



3.9 旋转

回忆 3.4 节中讨论的内容，保长和保角是正交矩阵所表示的变换之特征。接下来我们将详细讨论那些描述旋转变换的正交矩阵。

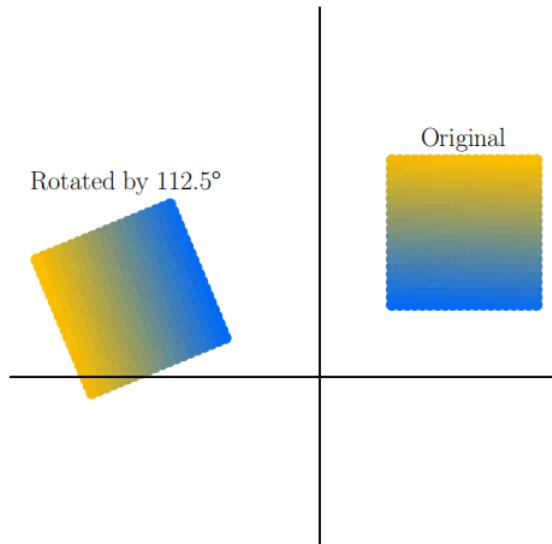


图 3.14 某旋转变换让一个图形绕原点旋转，转角为正表示逆时针旋转。

一个 **旋转 (rotation)** 是指一个将某个平面关于原点旋转角度 θ 的线性映射（具体而言，它是 Euclid 空间的**自同构(automorphism)**），也就是说旋转过程中不变的点原点。根据通常的约定，旋转角 $\theta > 0$ 表示逆时针旋转。如图 3.14 所示，其中的旋转矩阵如下：

$$\mathbf{R} = \begin{bmatrix} -0.38 & -0.92 \\ 0.92 & -0.38 \end{bmatrix} \quad (3.74)$$

旋转在机器人学和计算机图形学等领域中有重要的应用。如在机器人学中（图 3.15），我们需要知道如何旋转机器人的各关节，使其可以抓取或放置某个物件。

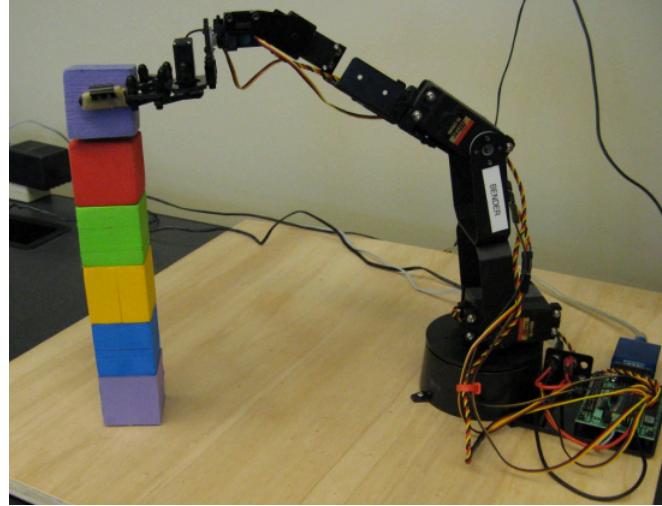


图 3.15 机械臂的各关节需要正确旋转才能正确地抓取或放置物件

图片来源于 Deisenroth et al.. 2015

3.9.1 \mathbb{R}^2 中的旋转

考虑 \mathbb{R}^2 中定义了笛卡尔坐标的标准基 $\left\{\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right\}$, 我们想要像图 3.16 那样将其旋转某个角度 θ 。注意我们可以看出旋转后的两个向量依然是线性无关的, 这说明它们还是 \mathbb{R}^2 的一个基——这说明旋转其实是一个基变换。

由于旋转操作 Φ 是线性映射, 我们也可以将其表示为一个 **旋转矩阵** (**rotation matrix**) $\mathbf{R}(\theta)$ 。我们可以通过三角函数得到旋转后向量 (旋转 Φ 的像 (**image**)) 在原坐标系下的坐标, 也就是

$$\Phi(\mathbf{e}_1) = \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}, \quad \Phi(\mathbf{e}_2) = \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}. \quad (3.75)$$

这样一来我们就得到了做上述基变换的旋转矩阵 $\mathbf{R}(\theta)$:

$$\mathbf{R}(\theta) = [\Phi(\mathbf{e}_1) \quad \Phi(\mathbf{e}_2)] = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \quad (3.76)$$

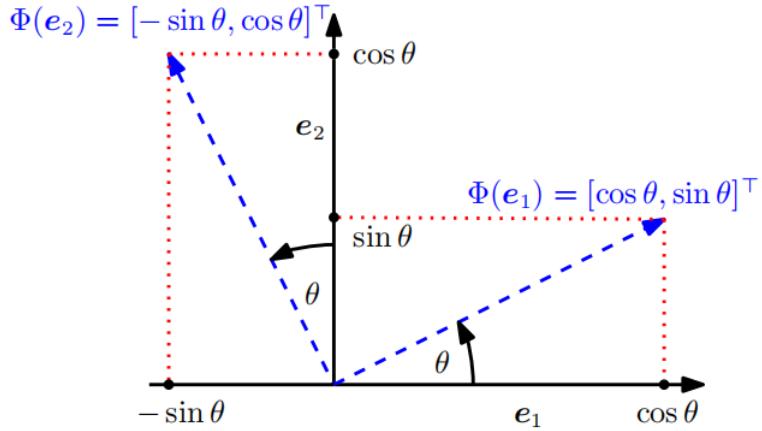


图 3.16 将二维Euclid平面中的正交基旋转角度 θ

3.9.2 \mathbb{R}^3 中的旋转

和 \mathbb{R}^2 不同的是，在 \mathbb{R}^3 中我们可以将其中的一个二维平面绕着一维的轴旋转。确定一个一般的旋转最简单的方法就是找到它是如何旋转标准基 e_1, e_2, e_3 ，并使旋转后的像 Re_1, Re_2, Re_3 两两正交。这样我们可以将三个标准基经旋转后的像并起来得到最终的旋转矩阵 R 。

为了使得所谓“旋转角”在超过二维空间中的旋转具有实际意义，我们需要定义在此情况下什么叫做“逆时针旋转”。依照常识，一个关于某轴的“逆时针”（或平面）旋转是指我们从轴的正方向末端朝着原点方向看到的旋转。如图在 \mathbb{R}^3 中存在三个关于三个标准基向量的平面旋转：

- 关于 e_1 方向的轴的旋转：

$$R_1(\theta) = [\Phi(e_1) \quad \Phi(e_2) \quad \Phi(e_3)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}. \quad (3.77)$$

其中 e_1 方向是不变的， e_2e_3 平面的旋转是逆时针的。

- 关于 e_2 方向的轴的旋转：

$$R_2(\theta) = [\Phi(e_1) \quad \Phi(e_2) \quad \Phi(e_3)] = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}. \quad (3.78)$$

如果我们旋转 e_1e_3 平面，我们需要从 e_2 轴的远端朝着原点看。

- 关于 e_3 方向的轴的旋转：

$$\mathbf{R}_3(\theta) = [\Phi(\mathbf{e}_1) \quad \Phi(\mathbf{e}_2) \quad \Phi(\mathbf{e}_3)] = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.79)$$



图 3.17 显示了这种情形。

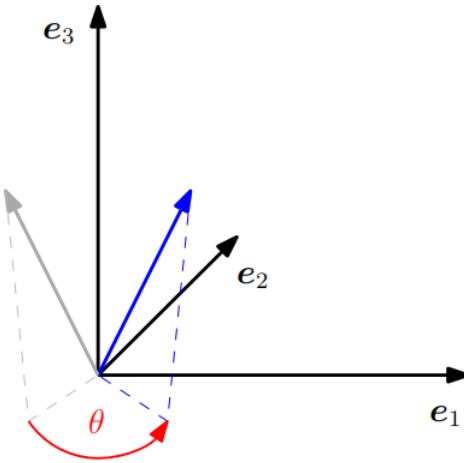


图 3.17 将三维空间中的一个向量（灰色）绕着 \mathbf{e}_3 轴旋转了 θ 角度（蓝色）

3.9.3 n 维空间中的旋转

直觉上，从二维、三维旋转到 n 维旋转的推广可被描述为将旋转限制在某个二维平面内，其他维度不变。如在前面例子中提到的三维空间中，我们可以旋转其中的二维平面。

定义 3.11 (Givens 旋转) 设 V 是 n 维 Euclid 空间，其上的自同构 Φ ：
 $V \rightarrow V$ 若可表示为

$$\mathbf{R}_{i,j}(\theta) := \begin{bmatrix} \mathbf{I}_{i-1} & & & \\ & \cos \theta & -\sin \theta & \\ & \sin \theta & \cos \theta & \\ & & & \mathbf{I}_{n-j} \end{bmatrix} \in \mathbb{R}^n \quad (3.80)$$

其中 $1 \leq i < j \leq n$, $\theta \in \mathbb{R}$, 空白处均为零，则 $\mathbf{R}_{i,j}(\theta)$ 叫做 Givens 旋转。简单来说，它就是单位矩阵 \mathbf{I}_n 加上下面的条件：

$$r_{i,i} = \cos \theta, \quad r_{i,j} = -\sin \theta, \quad r_{j,i} = \sin \theta, \quad r_{j,j} = \cos \theta \quad (3.81)$$

若 $n = 2$, 我们就得到了二维的特殊情况, 也即 (3.76)。



3.9.4 旋转算子的性质

旋转算子有不少可由正交矩阵 (定义 3.8) 推导而来的有用性质:

- 保距, 即 $\|\mathbf{x} - \mathbf{y}\| = \|\mathbf{R}_\theta(\mathbf{x}) - \mathbf{R}_\theta(\mathbf{y})\|$, 这是说旋转前后两点间距保持不变。
- 保角, 这是说 $\mathbf{R}_\theta(\mathbf{x})$ 与 $\mathbf{R}_\theta(\mathbf{y})$ 间的夹角和 \mathbf{x} 与 \mathbf{y} 间的夹角相同。
- 三维或更高维度中的旋转一般不满足交换律。因此即使是相对于同一点的旋转, 顺序也异常重要。只有二维旋转拥有可交换这一良好性质: 对任意 $\phi, \theta \in [0, 2\pi)$, 都有 $\mathbf{R}(\phi)\mathbf{R}(\theta) = \mathbf{R}(\theta)\mathbf{R}(\phi)$ 。只当它们都是绕原点旋转时, 全体二维旋转关于乘法构成 Abel 群。

译者注: 此“乘法”可理解为作为旋转作为矩阵的乘法, 也可以理解为旋转作为映射的复合。

< 上一章节

下一章节 >

3.8 正交投影

3.10 拓展阅读



3.10 拓展阅读

本章我们简要概述了解析几何的一些重要概念，将在本书后续章节中使用。对它们更广泛和深入的概述，我们推荐以下几本优秀的书籍：Axler (2015) 和 Boyd and Vandenberghe (2018)。

内积的存在使我们能用 Gram-Schmidt 正交化方法确定特定线性空间或子空间的基，基向量两两正交。这些基在优化和求解线性方程组的数值算法中非常重要。例如，Krylov 子空间方法、共轭梯度法和广义最小残差方法（generalized minimal residual method, GMRES）它最小化彼此正交的残差误差（Stoer and Burlirsch, 2002）。

在机器学习领域，内积在核方法（Schölkopf and Smola, 2002）中十分很重要。核方法利用了这样一个事实：许多线性算法可以仅通过内积计算来表达。然后，“核技巧”允许我们在（可能是无限维的）特征空间中隐式地计算这些内积，甚至不必明确知道这个特征空间。这使得许多用于机器学习的算法得以“非线性化”，例如用于降维的核PCA（kernel PCA, Schölkopf et al., 1997）。同属于核方法的范畴的高斯过程（gaussian process, Rasmussen and Williams, 2006），是概率回归（拟合曲线到数据点）的最新技术。我们将在第12章进一步探讨核的概念。

投影在计算机图形学中经常使用，如用于生成阴影。在优化中，正交投影经常用于（迭代地）最小化残差误差。这也在机器学习中有应用，例如在线性回归中，我们要找到一个（线性）函数，该函数最小化残差误差，即数据到线性函数的正交投影的长度（Bishop, 2006）。我们将在第9章进一步研究这个问题。PCA（Pearson, 1901; Hotelling, 1933）也使用投影来对高维数据降维，它们将在第10章得到更详细地讨论。

< 上一章节

3.9 旋转

下一章节 >

习题



习题

3.1

证明对所有的 $x = [x_1, x_2]^T \in \mathbb{R}^2$ 和 $y = [y_1, y_2]^T \in \mathbb{R}^2$ ，如下定义的函数 $\langle \cdot, \cdot \rangle$ 是一个内积。

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

3.2

考虑带有如下定义之函数 $\langle \cdot, \cdot \rangle$ 的 \mathbb{R}^2 ，此函数是一个内积吗？

$$\langle x, y \rangle := x^T \underbrace{\begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix}}_{=:A} y$$

3.3

用下列不同的内积定义计算 x 和 y 的距离：

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad y = \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix}$$

a. $\langle x, y \rangle := x^T y$ b. $\langle x, y \rangle := x^T A y$, $A := \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}$

3.4

用下列不同的内积定义计算 x 和 y 的夹角：

$$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad y = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$



a. $\langle x, y \rangle := x^T y$ b. $\langle x, y \rangle := x^T B y, B := \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$

3.5

考慮裝配點積的 Euclid 空間 \mathbb{R}^5 。一個子空間 $U \subseteq \mathbb{R}^5$ 和一個向量 $x \in \mathbb{R}^5$ 如下：

$$U = \text{span} \left[\begin{bmatrix} 0 \\ -1 \\ 2 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ -3 \\ 1 \\ -1 \\ 2 \end{bmatrix}, \begin{bmatrix} -3 \\ 4 \\ 1 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ -3 \\ 5 \\ 0 \\ 7 \end{bmatrix} \right], \quad x = \begin{bmatrix} -1 \\ -9 \\ -1 \\ 4 \\ 1 \end{bmatrix}$$

- a. 計算 x 到 U 的正交投影 $\pi_U(x)$ b. 計算 x 到 U 的距離 $d(x, U)$

3.6

考慮裝配有如下內積的 \mathbb{R}^3 ：

$$\langle x, y \rangle := x^T \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} y$$

記 e_1, e_2, e_3 為 \mathbb{R}^3 中的標準基。

- a. 計算 e_2 至子空間 $U = \text{span}\{e_1, e_3\}$ 的投影 $\pi_U(e_2)$ b. 計算 e_2 到 U 的距離 $d(e_2, U)$ c. 請繪制所有的標準正交基和 $d(e_2, U)$

提示：正交性是由內積決定的

3.7

令 V 為一線性空間， π 是其上的一個自同態。a. 证明： π 是投影變換，當且僅當 $\text{id}_V - \pi$ 是一個投影變換，其中 id_V 是 V 上的單位同態。b. 現假設 π 是投影變換，計算 $\text{Im}(\text{id}_V - \pi)$ 和 $\ker(\text{id}_V - \pi)$ 作為 $\text{Im}(\pi)$ 和 $\ker(\pi)$ 的函數。



3.8

使用 Gram-Schmidt 正交化方法，将某二维子空间 $U \subseteq \mathbb{R}^3$ 的基 $B = (b_1, b_2)$ 转换为 U 中的标准正交基 $C = (c_1, c_2)$ ，其中

$$b_1 := \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad b_2 := \begin{bmatrix} -1 \\ 2 \\ 2 \end{bmatrix}$$

3.9

令 $n \in \mathbb{N}$ 同时令 $x_1, \dots, x_n > 0$ 为 n 个正实数，且满足 $x_1 + \dots + x_n = 1$. 使
用 Cauchy-Schwartz 不等式证明： a. $\sum_{i=1}^n x_i^2 \geq 1$ b. $\sum_{i=1}^n \frac{1}{x_i} \geq n^2$

提示：考虑 \mathbb{R}^n 上的内积。然后选择恰当的 $x, y \in \mathbb{R}^n$ 。

3.10

将下列向量旋转 30° 。

$$x_1 := \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad x_2 := \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

< 上一章节

3.10 拓展阅读



第四章 矩阵分解

译者注：这一章是线性代数运算中比较重要的章节，也是后续很多算法的核心原理例如PCA等。在这一节中要注意体会矩阵分解的基本思想，并能给出不同分解方法的实现。

在第2章和第3章中，我们研究了向量的运算与度量、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，可以用矩阵的行表示不同的人，列描述人的不同特征，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：如何对矩阵组合，如何分解矩阵，以及如何将这些分解用于矩阵近似。

我们首先考虑允许我们用几个数字来描述矩阵特征的方法，这些数字表征了矩阵的整体性质。对于方阵的重要特例，我们将在行列式（第4.1节）和特征值（第4.2节）部分进行讨论。这些特征值具有重要的数学意义，使我们能够快速掌握矩阵具有哪些有用的性质。在这里我们将继续讨论矩阵分解方法：矩阵分解可以类比为数字的因式分解，例如将21因式分解为素数7和3。因此，矩阵分解（matrix decomposition）也常被称为**matrix factorization**。矩阵分解用于通过使用可解释矩阵的因子的不同表示来描述矩阵。

我们将首先介绍对称正定矩阵的平方根运算，即Cholesky分解（第4.3节）。从这里，我们将看看将矩阵分解为规范形式的两种相关方法。第一种称为矩阵对角化（第4.4节），如果我们选择合适的基，它允许我们使用对角变换矩阵来表示线性映射。第二种方法，奇异值分解（第4.5节），将这种因式分解扩展到非方阵，它被认为是线性代数中的基本概念之一。这些分解是有帮助的，因为表示数值数据的矩阵通常非常大，很难分析。我们以矩阵分类的形式系统地概述了矩阵的类型和区分它们的特征属性（第4.7节）来结束本章。

这一章中讲述的一些方法对后续的一些数学理论性章节例如第6章以及一些应用性章节例如第10章中的降维和第11章中的密度估计都有重要作用。本章的整体结构如图4.1所示：

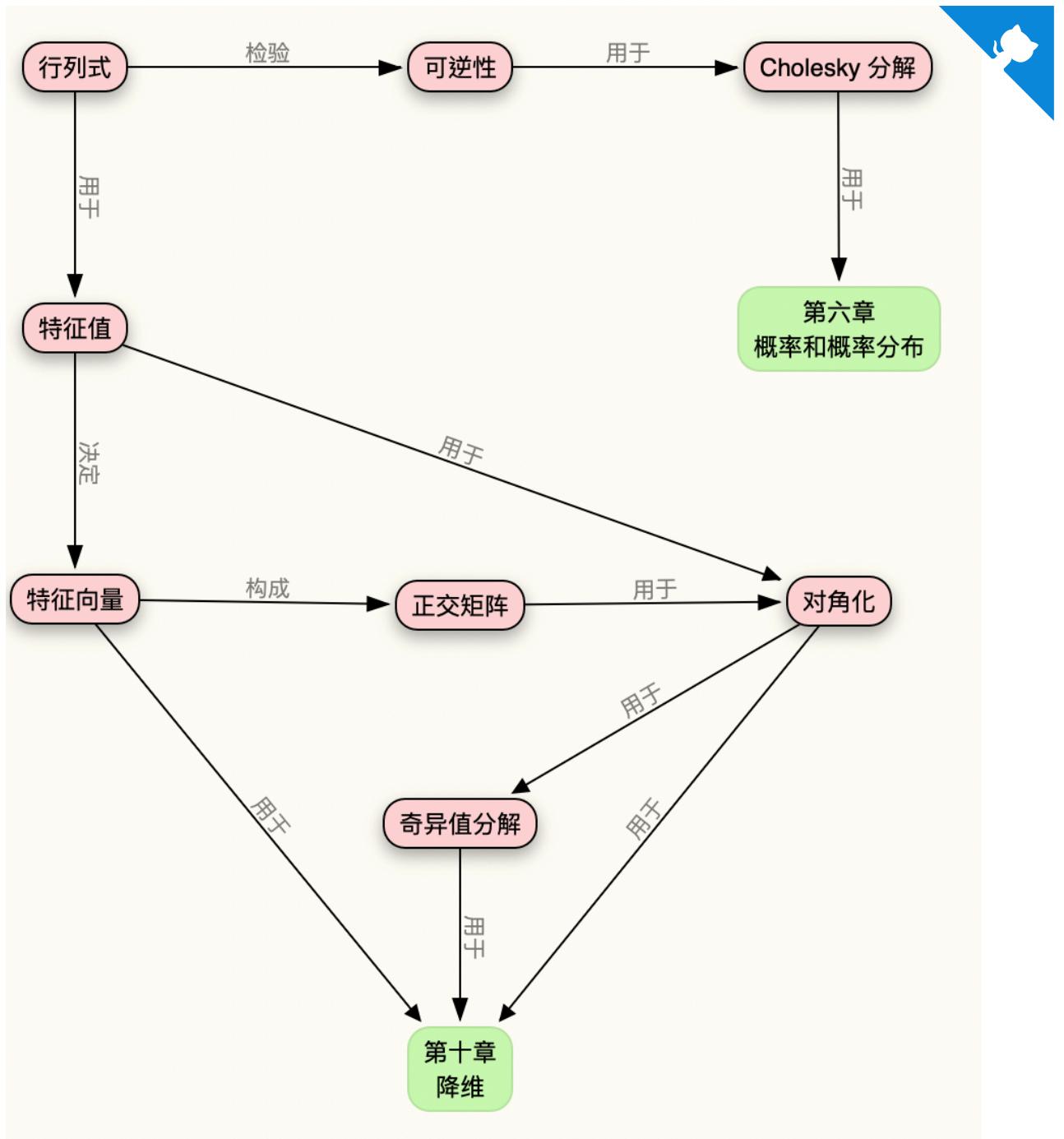


图4.1

本章介绍的概念的思维导图，以及它们在本书其他部分的使用位置。



< 上一章节

下一章节 >

第三章 解析几何

第五章 向量微积分



4.1 矩阵的行列式与迹

行列式是线性代数中的重要概念。行列式是线性方程组分析和求解中的数学对象。行列式仅在方阵 $A \in R^{n \times n}$ 上定义，即具有相同行数和列数的矩阵。在这本书中，我们把行列式写成 $\det(A)$ ，有时写成 $|A|$ ：

$$\det(A) = \begin{vmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{vmatrix} \quad (4.1)$$

方阵 $A \in R^{n \times n}$ 的行列式是一个将 A 映射为一个实数的函数。在给出一般 $n \times n$ 矩阵行列式的定义之前，让我们来看一些典型的例子，并定义一些特殊矩阵的行列式。

例4.1 检验矩阵是否可逆

让我们从一个方阵 A 是否可逆（参考2.2.2节）开始。对于最小的二维方阵，我们已经知道何时矩阵是可逆的了。如果 A 是一个 1×1 矩阵，即它是一个标量，那么 $A = a \Rightarrow A^{-1} = \frac{1}{a}$ ，当且仅当 $a \neq 0$ 时 $a \frac{1}{a} = 1$ 成立。

对于一个 2×2 矩阵，由逆矩阵的定义（式2.3）我们知道 $AA^{-1} = I$ ，随即，由式2.24， A 的逆可以定义为：

$$A^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \quad (4.2)$$

这里， A 是可逆矩阵当且仅当：

$$a_{11}a_{22} - a_{12}a_{21} \neq 0 \quad (4.3)$$

这个式子就是对于 $A \in R^{2 \times 2}$ 的行列式：

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \quad (4.4)$$

例4.1已经指出了行列式和逆矩阵存在之间的关系。下一个定理陈述了 $n \times n$ 矩阵的相同结果。

定理4.1 对于任意方阵 $A \in R^{n \times n}$, 当且仅当 $\det(A) \neq 0$ 时 A 可逆。

我们有针对小型矩阵的行列式的显式（封闭形式）表达式。对于 $n=1$,

$$\det(A) = \det(a_{11}) = a_{11} \quad (4.5)$$

对于 $n=2$,

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \quad (4.6)$$

对于 $n=3$ （也就是大家熟知的 Sarrus 规则）：

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{21}a_{32}a_{13} + a_{31}a_{12}a_{23} - a_{31}a_{22}a_{13} - a_{12}a_{23} - a_{21}a_{32}a_{13} - a_{31}a_{12}a_{23} - a_{11}a_{22}a_{33}$$

为了帮助记忆 Sarrus 规则中的乘积项，请尝试在矩阵中追踪三乘积的元素。

如果对于每个 $i > j$ 都有 $T_{ij} = 0$, 我们称一个方阵 T 为上三角矩阵, 即矩阵对角线以下为零。类似地, 我们定义下三角矩阵, 即对角线上方为零的矩阵。对于一个三角矩阵 $T \in R^{n \times n}$, 它的行列式就是对角线上元素的乘积:

$$\det(T) = \prod_{i=1}^n T_{ii} \quad (4.8)$$

例4.2 行列式可以作为体积的度量

当我们把行列式视为 R^n 中跨越对象的 n 个向量集的映射时, 行列式的概念是很自然的。结果表明, 行列式 $\det(A)$ 是由矩阵 A 的列形成的 n 维平行六面体的有符号体积。

对于 $n=2$, 矩阵的列形成平行四边形; 见图4.2。随着向量之间的角度变小, 平行四边形的面积也会缩小。



图4.2

向量 b 和 g 所跨越的平行四边形（阴影区域）的面积为 $|\det([b,g])|$

考虑将两个列向量 b, g 构成一个矩阵 $A = [b, g]$ ，那么， A 行列式的绝对值是顶点为 $0, b, g, b+g$ 的平行四边形的面积。特别的，如果 b, g 线性相关，也就有 $b = \lambda g, \lambda \in R$ ，它们无法构成二维平行四边形，因此对应的区域面积为0。相反地，如果 b, g 线性无关且构成一组正交基向量 e_1, e_2 ，那么它们也可以写作 $b = \begin{bmatrix} b \\ 0 \end{bmatrix}$ 和 $g = \begin{bmatrix} 0 \\ g \end{bmatrix}$ ，那么这个行列式就是 $\begin{vmatrix} b & 0 \\ 0 & g \end{vmatrix} = bg - 0 = bg$ 。

行列式的符号表示生成向量 b, g 相对于标准基 (e_1, e_2) 的方向。在我们的图中，把顺序翻转为 g, b 就可以交换 A 的列并翻转阴影区域面积的方向。这就是我们所熟悉的：平行四边形面积等于底乘高。这样的直觉可以扩展到更高的维度。在 R^3 中，我们考虑三个向量 $r, b, g \in R^3$ 构成平行六面体的三条边，即六个面都是平行四边形的立体图形，如图4.3所示：



图4.3

向量 r, b 和 g 所跨越的平行六面体（阴影区域）的体积为 $|\det([r, b, g])|$ ，行列式的符号表示生成向量的方向。

这个 3×3 矩阵 $[r, b, g]$ 行列式的绝对值就是平行六面体的体积。因此，行列式充当一个函数，用于测量由矩阵中组成的列向量形成的有符号体积。

考虑三个线性无关的向量 $r, g, b \in R^3$ ：

$$r = \begin{bmatrix} 2 \\ 0 \\ -8 \end{bmatrix}, g = \begin{bmatrix} 6 \\ 1 \\ 0 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 4 \\ -1 \end{bmatrix} \quad (4.9)$$

把这些向量作为矩阵的列：

$$A = [r, g, b] = \begin{bmatrix} 2 & 6 & 1 \\ 0 & 1 & 4 \\ -8 & 0 & -1 \end{bmatrix} \quad (4.10)$$

这样我们就可以计算所需的体积：

$$V = |\det(A)| = 186. \quad (4.11)$$

计算 $n \times n$ 矩阵的行列式需要一个通用的算法来解决 $n > 3$ 的情况，我们将在下面进行探索。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。通过递归应用拉普拉斯展开（定理4.2），我们可以通过最终计算 2×2 矩阵的行列式来计算 $n \times n$ 矩阵的行列式。

定理4.2（拉普拉斯展开）考虑一个矩阵 $A \in R^{n \times n}$ ，那么，对于 $j = 1, 2, \dots, n$ ，有：

1. 按第j列展开：

$$\det(A) = \sum_{k=1}^n (-1)^{k+j} a_{kj} \det(A_{k,j}) \quad (4.12)$$

2. 按第i行展开：

$$\det(A) = \sum_{k=1}^n (-1)^{k+i} a_{ik} \det(A_{i,k}) \quad (4.13)$$

这里 $A_{k,j}$ 是矩阵A删除第i行和第j列得到的子矩阵。

例4.3 拉普拉斯展开

让我们计算这样一个矩阵的行列式：

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

按式（4.13）的规则对第一行应用一次拉普拉斯展开：

$$\begin{vmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 0 & 0 & 1 \end{vmatrix} = (-1)^{1+1} 1 \begin{vmatrix} 1 & 2 \\ 0 & 1 \end{vmatrix} + (-1)^{1+2} 2 \begin{vmatrix} 3 & 2 \\ 0 & 1 \end{vmatrix} + (-1)^{1+3} 3 \begin{vmatrix} 3 & 1 \\ 0 & 0 \end{vmatrix}$$

我们利用(4.6)来计算所有的二阶行列式：

$$\det(A) = 1(1 - 0) - 2(3 - 0) + 3(0 - 0) = -5 \quad (4.16)$$

为了完整起见，我们可以将这个结果与使用Sarrus规则（4.7）计算行列式进行比较：

$$\det(A) = 1 \cdot 1 \cdot 1 + 3 \cdot 0 \cdot 3 + 0 \cdot 2 \cdot 2 - 0 \cdot 1 \cdot 3 - 1 \cdot 0 - 2 - 3 \cdot 2 \cdot 1 \quad (4.17)$$

对于 $A \in R^{n \times n}$, 行列式具有以下性质:

- 矩阵乘积的行列式等于行列式的乘积, 也就是 $\det(AB) = \det(A)\det(B)$
- 矩阵转置后求行列式与自身行列式相等, $\det(A) = \det(A^T)$
- 如果矩阵 A 是正规矩阵 (可逆), 那么 $\det(A^{-1}) = \frac{1}{\det(A)}$
- 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_Φ 具有相同的行列式。因此, 行列式对于线性映射的基的选择是不变的。
- 将列/行的倍数添加到另一列/行不会改变 $\det(A)$
- 将某一列/行放大 λ 倍会使得行列式被放大 $\lambda \in R$ 倍。特别地, $\det(\lambda A) = \lambda^n \det(A)$
- 交换两行/两列会改变 $\det(A)$ 的符号

由于最后三个性质, 我们可以使用高斯消元法 (见第2.1节) 通过将 A 转化为行阶梯形式来计算 $\det(A)$ 。当 A 呈三角矩阵, 也就是对角线以下的元素都为 0 时, 我们可以停止高斯消元。回想一下 (4.8), 三角矩阵的行列式是对角元素的乘积。

定理4.3 一个方阵 $A \in R^{n \times n}$ 有 $\det(A) = 0$, 当且仅当 $\text{rank}(A) = n$ 。换言之, 当且仅当 A 满秩时 A 可逆。

当数学主要由手工完成时, 行列式计算被认为是分析矩阵可逆性的一种基本方法。然而, 现代机器学习方法使用直接数值方法, 取代了行列式的显式计算。例如, 在第2章中, 我们了解到逆矩阵可以通过高斯消元法计算。因此, 高斯消元法可用于计算矩阵的行列式。

行列式将在后续章节中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时 (第4.2节) 更是如此。

定义4.4 一个方阵 $A \in R^{n \times n}$ 的迹为:

$$tr(A) = \sum_{i=1}^n a_{ii} \quad (4.18)$$

即, 一个矩阵的迹是 A 的对角线元素之和。

迹满足如下性质:



- 对于 $A, B \in R^{n \times n}$, $\text{tr}(A + B) = \text{tr}(A) + \text{tr}(B)$
- 对于 $A \in R^{n \times n}$, $\alpha \in R$, $\text{tr}(\alpha A) = \alpha \text{tr}(A)$
- $\text{tr}(I_n) = n$
- 对于 $A \in R^{n \times k}$, $B \in R^{k \times n}$, $\text{tr}(AB) = \text{tr}(BA)$

可以证明只有一个函数能同时满足上述四条性质，就是矩阵的迹(Gohberg et al., 2012)。

矩阵相乘求迹的性质可以更泛化。具体而言，该性质在循环置换下是不变的。即：

$$\text{tr}(AKL) = \text{tr}(KLA) \quad (4.19)$$

对于 $A \in R^{a \times k}$, $B \in R^{k \times l}$, $C \in R^{l \times a}$ 成立。这个性质可以推广到任意数量的矩阵的乘积。作为 (4.19) 的特例，对于两个向量 $x, y \in R^n$,

$$\text{tr}(xy^T) = \text{tr}(y^Tx) = y^Tx \in R \quad (4.20)$$

给定一个线性映射 $\Phi : V \rightarrow V$, 其中 V 是一个线性空间，我们通过使用 Φ 的矩阵表示的轨迹来定义这个映射的迹。对于给定 V 的基，我们可以用变换矩阵 A 来描述 Φ 。那么 Φ 的迹就是 A 的迹。对于不同的 V 基来说，它认为 Φ 的相应变换矩阵 B 可以对适当的 S 通过 $S^{-1}AS$ 形式的基变化来获得（见第 2.7.2 节）。对于 Φ 的对应迹，这也就是说：

$$\text{tr}(B) = \text{tr}(S^{-1}AS) = \text{tr}(AS^{-1}S) = \text{tr}(A) \quad (4.21)$$

因此，虽然线性映射的矩阵表示是基相关的，但线性映射 Φ 的轨迹与基无关。

在本节中，我们介绍了行列式和迹作为表征方阵的函数。结合我们对行列式和迹的理解，我们现在可以定义一个用多项式描述矩阵 A 的重要方程，我们将在后续章节中广泛使用。

定义 4.5 (特征多项式) 对于 $\lambda \in R$ 和 $A \in R^{n \times n}$,

$$\begin{aligned} p_A(\lambda) &= \det(A - \lambda I) \\ &= c_0 + c_1\lambda + c_2\lambda^2 + \cdots + c_{n-1}\lambda^{n-1} + c_n\lambda^n \end{aligned} \quad (4.22)$$

$c_0, c_1, \dots, c_n \in R$, 这被称作 A 的特征多项式。特别地，

$$c_0 = \det(A) \quad (4.23)$$

$$c_{n-1} = (-1)^{n-1} \text{tr}(A) \quad (4.24)$$

特征多项式（4.22）将允许我们计算特征值和特征向量，下一节将介绍。



[下一章节 >](#)

4.2 特征值与特征向量



4.2 特征值与特征向量

现在，我们将了解一种新的方式来描述矩阵及其相关的线性映射。回想一下第2.7.1节的内容，给定一个有序基，每个线性映射都有一个唯一的变换矩阵。我们可以通过进行“特征”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（即特征向量）是如何被线性映射变换的。

定义4.6. 设 $A \in \mathbb{R}^{n \times n}$ 是一个方阵。那么，如果 $\lambda \in \mathbb{R}$ 满足

$$Ax = \lambda x \quad (4.25)$$

则称 λ 为 A 的特征值，而 $x \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ 为对应的特征向量。我们称(4.25)为特征值方程。

备注：在线性代数文献和软件中，通常约定特征值按降序排列，因此最大的特征值及其对应的特征向量被称为第一特征值和第一特征向量，次大的被称为第二特征值和第二特征向量，以此类推。然而，教科书和出版物可能有不同的排序观念，或者根本没有排序。如果本书中没有明确说明排序，我们则不假定任何排序。

以下陈述是等价的：

- λ 是 $A \in \mathbb{R}^{n \times n}$ 的特征值。
- 存在一个 $x \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ 使得 $Ax = \lambda x$ ，或者等价地， $(A - \lambda I_n)x = \mathbf{0}$ 有非零解，即 $x \neq \mathbf{0}$ 。
- $\text{rk}(A - \lambda I_n) < n$ 。
- $\det(A - \lambda I_n) = 0$ 。

定义4.7（共线性和同向性）。两个指向相同方向的向量称为同向的。如果两个向量指向相同或相反的方向，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 相关联的 A 的特征向量，那么对于任意 $c \in \mathbb{R} \setminus \{0\}$ ， cx 也是与相同特征值 λ 相关联的 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 共线的向量也都是 A 的特征向量。

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 是矩阵 A 的特征多项式 $p_A(\lambda)$ 的根。

定义4.9. 设方阵 A 有一个特征值 λ_i , 则 λ_i 的代数重数是指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 相关联的所有特征向量张成的 \mathbb{R}^n 的子空间称为 A 关于 λ 的特征空间, 记作 E_λ 。矩阵 A 的所有特征值的集合称为 A 的特征谱或简称谱。

如果 λ 是 $A \in \mathbb{R}^{n \times n}$ 的特征值, 则对应的特征空间 E_λ 是齐次线性方程组 $(A - \lambda I)x = 0$ 的解空间。从几何角度看, 与非零特征值相对应的特征向量指向一个被线性映射拉伸的方向, 而特征值则是拉伸的因子。如果特征值为负, 则拉伸的方向会反转。

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 $p_I(\lambda) = \det(I - \lambda I) = (1 - \lambda)^n = 0$, 它只有一个特征值 $\lambda = 1$, 且该特征值出现 n 次。此外, 对于所有非零向量 $x \in \mathbb{R}^n$, 都有 $Ix = \lambda x = 1x$ 。因此, 单位矩阵的唯一特征空间 E_1 张成 n 维空间, 且 \mathbb{R}^n 的所有标准基向量都是 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^\top 具有相同的特征值, 但不一定具有相同的特征向量。
- 特征空间 E_λ 是 $A - \lambda I$ 的零空间, 因为

$$\begin{aligned} Ax = \lambda x &\iff Ax - \lambda x = 0 \\ &\iff (A - \lambda I)x = 0 \iff x \in \ker(A - \lambda I). \end{aligned} \tag{4.27}$$

- 相似矩阵 (见定义2.22) 具有相同的特征值。因此, 线性映射 Φ 的特征值与其变换矩阵的基选择无关。这使得特征值、行列式和迹成为线性映射的关键特征参数, 因为它们在基变换下是不变的。
- 对称、正定矩阵总是具有正实特征值。

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

$$\mathbf{A} = \begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix} \quad (4.28)$$

的特征值和特征向量。

步骤 1: 特征多项式。根据特征向量 $\mathbf{x} \neq \mathbf{0}$ 和特征值 λ 的定义, 存在向量使得 $A\mathbf{x} = \lambda\mathbf{x}$, 即 $(A - \lambda I)\mathbf{x} = \mathbf{0}$ 。由于 $\mathbf{x} \neq \mathbf{0}$, 这要求 $A - \lambda I$ 的核 (零空间) 包含除 0 以外的元素。这意味着 $A - \lambda I$ 不可逆, 因此 $\det(\mathbf{A} - \lambda \mathbf{I}) = 0$ 。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

$$\begin{aligned} p_{\mathbf{A}}(\lambda) &= \det(\mathbf{A} - \lambda \mathbf{I}) \\ &= \det \left(\begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right) = \begin{vmatrix} 4 - \lambda & 2 \\ 1 & 3 - \lambda \end{vmatrix} \quad (4.29) \\ &= (4 - \lambda)(3 - \lambda) - 2 \cdot 1. \end{aligned}$$

我们分解特征多项式得到

$$p(\lambda) = (4 - \lambda)(3 - \lambda) - 2 \cdot 1 = 10 - 7\lambda + \lambda^2 = (2 - \lambda)(5 - \lambda)$$

给出根 $\lambda_1 = 2$ 和 $\lambda_2 = 5$ 。

步骤 3: 特征向量和特征空间。我们通过查看满足以下条件的向量 \mathbf{x} 来找到与这些特征值相对应的特征向量

$$\begin{bmatrix} 4 - \lambda & 2 \\ 1 & 3 - \lambda \end{bmatrix} \mathbf{x} = \mathbf{0}. \quad (4.31)$$

对于 $\lambda = 5$, 我们得到

$$\begin{bmatrix} 4 - 5 & 2 \\ 1 & 3 - 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 & 2 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{0}. \quad (4.32)$$

我们解这个齐次系统, 得到解空间

$$E_5 = \text{span} \left[\begin{bmatrix} 2 \\ 1 \end{bmatrix} \right]. \quad (4.33)$$

这个特征空间是一维的, 因为它只有一个基向量。类似地, 我们通过解齐次方程组来找到对应于 $\lambda = 2$ 的特征向量



$$\begin{bmatrix} 4-2 & 2 \\ 1 & 3-2 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 2 & 2 \\ 1 & 1 \end{bmatrix} \mathbf{x} = \mathbf{0}. \quad (4.34)$$

这意味着任何形式为 $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ 的向量，其中 $x_2 = -x_1$ ，比如 $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ ，都是对应于特征值2的特征向量。对应的特征空间由下式给出：

$$E_2 = \text{span}\left[\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right]. \quad (4.35)$$

这意味着特征空间 E_2 是一维的，并且由单个向量 $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ 张成 (span)。任何满足 $x_2 = -x_1$ 的向量都可以表示为 $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ 的标量倍，因此它们都是特征值 $\lambda = 2$ 的特征向量。

在例4.5中，两个特征空间 E_5 和 E_2 都是一维的，因为它们各自都仅由一个向量张成。然而，在其他情况下，我们可能有多个相同的特征值（参见定义4.9），并且特征空间可能具有超过一个的维度。

定义4.11. 设 λ_i 是方阵 A 的一个特征值。那么， λ_i 的几何重数是与 λ_i 相关联的线性无关特征向量的数量。换句话说，它是与 λ_i 相关联的特征向量所张成的特征空间的维度。

备注。 一个特定特征值的几何重数必须至少为1，因为每个特征值都至少有一个相关联的特征向量。一个特征值的几何重数不能超过其代数重数，但可能更低。

例4.6 矩阵 $A = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$ 有两个重复的特征值 $\lambda_1 = \lambda_2 = 2$ ，且代数重数为2。然而，该特征值只有一个不同的单位特征向量 $\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ ，因此，几何重数为1。

图形直觉在二维空间中的应用

让我们通过不同的线性映射来获得关于行列式、特征向量和特征值的一些直观理解。图4.4描绘了五个变换矩阵 A_1, \dots, A_5 以及它们对以原点为中心的方形网格点的影响：

 1723777178409

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概览**

$\$A_{\{i\}} \in \mathbb{R}^{2 \times 2}$ 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)。中间列展示了第一个特征向量被其对应的特征值 λ_1 拉伸，以及第二个特征向量被其对应的特征值 λ_2 拉伸。每一行展示了五个变换矩阵 A_i 之一在标准基下的效果。

- $A_1 = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 2 \end{bmatrix}$ 。这个矩阵的两个特征向量的方向对应于 \mathbb{R}^2 中的标准基向量，即两个坐标轴。垂直轴被放大了2倍（特征值 $\lambda_1 = 2$ ），而水平轴被压缩了 $\frac{1}{2}$ 倍（特征值 $\lambda_2 = \frac{1}{2}$ ）。这个映射是保面积的，因为行列式 $\det(A_1) = 1 = 2 \cdot \frac{1}{2}$ 。
- $A_2 = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{bmatrix}$ 这个矩阵对应于一个剪切映射。它将网格点沿着水平轴向右剪切，如果它们在正y轴的一侧；如果它们在负y轴的一侧，则向左剪切。这种剪切不改变网格的面积（因为行列式为1），但改变了网格的形状。特征值 $\lambda_1 = \lambda_2 = 1$ 是重复的，且特征向量是共线的（尽管在这里为了强调，我们在两个相反的方向上绘制了它们）。这表明映射仅沿着一个方向（水平轴）作用，但实际上，由于剪切的存在，它也在垂直方向上产生了影响，只是没有改变面积。
- $A_3 = \begin{bmatrix} \cos(\frac{\pi}{6}) & -\sin(\frac{\pi}{6}) \\ \sin(\frac{\pi}{6}) & \cos(\frac{\pi}{6}) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \sqrt{3} & -1 \\ 1 & \sqrt{3} \end{bmatrix}$ 这个矩阵 A_3 将点逆时针旋转了 $\frac{\pi}{6}$ 弧度（即 30° ）。由于旋转是面积保持的，所以行列式为1。旋转矩阵的特征值是复数，反映了旋转的性质（因此没有绘制特征向量）。关于旋转的更多细节，请参考第3.9节。
- $A_4 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$ 这个矩阵表示一个映射，它将二维域压缩到一维。由于一个特征值为0 ($\lambda_1 = 0$)，与这个特征值对应的（蓝色）特征向量方向上的空间会塌陷，而与之正交的（红色）特征向量则会使空间拉伸一个因子 $\lambda_2 = 2$ 。然而，由于有一个特征值为0，整个变换后的图像面积实际上是0。



- $A_5 = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}$ 这是一个剪切和拉伸的映射。由于行列式的绝对值为 $\frac{3}{4}$ ，它表示空间被放大了 75%（注意这里的“放大”是指行列式绝对值与 1 的关系，实际上在某些方向上可能是压缩的）。它沿着（红色）特征向量 λ_2 的方向拉伸了空间 1.5 倍，并沿着与之正交的（蓝色）特征向量压缩了 0.5 倍。这种变换既改变了网格的形状，也改变了其面积。

特别是，行列式的绝对值表示变换后图形面积的缩放比例（在二维中），而特征值和特征向量则揭示了变换在特定方向上的行为。例如，特征值大于 1 表示该方向上的放大，小于 1 表示压缩，而复数特征值则通常与旋转或振荡相关。

例4.7 生物神经网络的特征谱



图4.5 *Caenorhabditis elegans* (线虫) 神经网络 (Kaiser 和 Hilgetag, 2006)

(a) 对称化连接性矩阵 **(b) 特征谱**

分析和学习网络数据的方法是机器学习方法的重要组成部分。理解网络的关键在于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，研究能够捕获这种连接性数据的矩阵通常非常有用。

我们构建了线虫 *C. elegans* 完整神经网络的连接性/邻接矩阵 $A \in \mathbb{R}^{277 \times 277}$ 。矩阵的每一行/列代表线虫大脑中的一个神经元（共 277 个）。如果神经元 i 通过突触与神经元 j 相连，则连接性矩阵 A 中对应元素 $a_{ij} = 1$ ，否则 $a_{ij} = 0$ 。由于连接性矩阵 A 可能不是对称的（即可能存在单向连接），其特征值可能不是实数。因此，我们计算了连接性矩阵的对称版本，记为 $A_{sym} := A + A^\top$ 。这个新的对称矩阵 A_{sym} 在图 4.5(a) 中展示，如果两个神经元之间存在连接（无论连接方向如何），则矩阵中对应元素 a_{ij} 为非零值（以白色像素表示）。在图 4.5(b) 中，我们展示了 A_{sym} 对应的特征谱。横轴表示按降序排列的特征值索引，纵轴表示对应的特征值。该特征谱呈现出典型的“S”形，这在许多生物神经网络中都很常见。关于这一现象背后的机制，是神经科学研究中的一个活跃领域。



定理4.12. 一个矩阵 $A \in \mathbb{R}^{n \times n}$ 的具有 n 个不同特征值 $\lambda_1, \dots, \lambda_n$ 的特征向量 x_1, \dots, x_n 是线性无关的。

这个定理表明，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. 如果一个方阵 $A \in \mathbb{R}^{n \times n}$ 拥有的线性无关特征向量少于 n 个，则称该矩阵是缺陷的。一个非缺陷的矩阵 $A \in \mathbb{R}^{n \times n}$ 不一定需要 n 个不同的特征值，但它确实需要其特征向量构成 \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 。特别是，缺陷矩阵至少有一个特征值 λ_i ，其代数重数 $m > 1$ 但几何重数小于 m 。

备注 缺陷矩阵不能有 n 个不同的特征值，因为不同的特征值具有线性无关的特征向量（定理4.12）。

定理4.14. 给定矩阵 $A \in \mathbb{R}^{m \times n}$ ，我们总可以通过定义

$$S := A^\top A \quad (4.36)$$

来获得一个对称且半正定的矩阵 $S \in \mathbb{R}^{n \times n}$ 。

备注. 如果 $\text{rk}(A) = n$ ，则 $S := A^\top A$ 是对称且正定的。

理解定理4.14为何成立对我们如何使用对称化矩阵有很大启示：对称性要求 $S = S^\top$ ，通过插入(4.36)我们得到 $S = A^\top A = A^\top (A^\top)^\top = (A^\top A)^\top = S^\top$ 。此外，半正定性（第3.2.3节）要求 $x^\top S x \geq 0$ ，插入(4.36)我们得到 $x^\top S x = x^\top A^\top A x = (x^\top A^\top)(Ax) = (Ax)^\top (Ax) \geq 0$ ，因为点积计算的是平方和（它们本身是非负的）。

定理4.15（谱定理）. 如果 $A \in \mathbb{R}^{n \times n}$ 是对称的，则存在由 A 的特征向量构成的对应线性空间 V 的一个正交规范基，且每个特征值都是实数。

谱定理的一个直接推论是，对称矩阵 A 的特征分解存在（具有实数特征值），并且我们可以找到一个由特征向量构成的正交规范基（ONB），使得 $A = PDP^\top$ ，其中 D 是对角矩阵， P 的列包含特征向量。

例4.8

考虑矩阵

$$A = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & 2 \\ 2 & 2 & 3 \end{bmatrix} \quad (4.37)$$

矩阵 A 的特征多项式为

$$p_A(\lambda) = -(\lambda - 1)^2(\lambda - 7) \quad (4.38)$$

由此，我们可以确定矩阵 A 的特征值为 $\lambda_1 = 1$ （这是一个重复的特征值）和 $\lambda_2 = 7$ 。

接下来，我们按照标准程序计算特征向量，得到与这些特征值对应的特征空间：

$$E_1 = \text{span} \left(\underbrace{\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}}_{=: \mathbf{x}_1}, \underbrace{\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}}_{=: \mathbf{x}_2} \right), \quad E_7 = \text{span} \left(\underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}_{=: \mathbf{x}_3} \right) \quad (4.39)$$

我们注意到 \mathbf{x}_3 与 \mathbf{x}_1 和 \mathbf{x}_2 都是正交的，但 \mathbf{x}_1 和 \mathbf{x}_2 之间不是正交的（因为 $\mathbf{x}_1^\top \mathbf{x}_2 = 1 \neq 0$ ）。然而，根据谱定理（定理4.15），我们知道存在一个由正交特征向量构成的基，但我们目前得到的基并不满足这个条件。不过，我们可以构造一个这样的基。

为了构造这样的基，我们利用 \mathbf{x}_1 和 \mathbf{x}_2 都是与同一特征值 $\lambda_1 = 1$ 相关联的特征向量这一事实。因此，对于任意的 $\alpha, \beta \in \mathbb{R}$ ，都有

$$A(\alpha \mathbf{x}_1 + \beta \mathbf{x}_2) = A\mathbf{x}_1\alpha + A\mathbf{x}_2\beta = \lambda_1(\alpha \mathbf{x}_1 + \beta \mathbf{x}_2) \quad (4.40)$$

即， \mathbf{x}_1 和 \mathbf{x}_2 的任何线性组合仍然是与 λ_1 相关联的特征向量。Gram-Schmidt 算法（第3.8.3节）是一种通过迭代地从一组基向量中构造正交/单位正交基的方法，它使用这样的线性组合。因此，即使 \mathbf{x}_1 和 \mathbf{x}_2 不是正交的，我们也可以应用 Gram-Schmidt 算法来找到与 $\lambda_1 = 1$ 相关联且相互正交（以及与 \mathbf{x}_3 正交）的特征向量。

在我们的例子中，应用 Gram-Schmidt 算法后，我们可能会得到（注意这里的 \mathbf{x}'_2 可能不是唯一的，因为它取决于 Gram-Schmidt 算法的具体实现）：

$$\mathbf{x}'_1 = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}'_2 = \frac{1}{2} \begin{bmatrix} -1 \\ -1 \\ 2 \end{bmatrix} \quad (4.41)$$

这两个向量是相互正交的，与 \mathbf{x}_3 也正交，并且是矩阵 A 与特征值 $\lambda_1 = 1$ 相关联的特征向量。

在结束对特征值和特征向量的讨论之前，将矩阵的这些特性与行列式和迹的概念联系起来是非常有用的。

定理 4.16: 一个矩阵 $A \in \mathbb{R}^{n \times n}$ 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。

 1723782864917

图4.6 几何上关于特征值的解释。矩阵 A 的特征向量被对应的特征值拉伸。单位正方形的面积变化了 $|\lambda_1 \lambda_2|$ 倍，周长变化了 $\frac{1}{2}(|\lambda_1| + |\lambda_2|)$ 倍。

定理4.17. 矩阵 $A \in \mathbb{R}^{n \times n}$ 的迹是其特征值的和，即

$$tr(A) = \sum_{i=1}^n \lambda_i \quad (4.43)$$

其中， $\lambda_i \in \mathbb{C}$ 是 A 的（可能重复的）特征值。

现在，我们来为这两个定理提供一个几何上的直观理解。考虑一个矩阵 $\dot{A} \in \mathbb{R}^{2 \times 2}$ ，它有两个线性无关的特征向量 $\mathbf{x}_1, \mathbf{x}_2$ 。为了这个例子，我们假设 $(\mathbf{x}_1, \mathbf{x}_2)$ 是 \mathbb{R}^2 的一个正交归一基（ONB），因此它们是正交的，并且它们所张成的正方形的面积是 1；见图4.6。从第4.1节我们知道，行列式计算的是在变换 A 下单位正方形面积的变化。在这个例子中，我们可以明确地计算出面积的变化：使用 A 映射特征向量得到向量 $v_1 = A\mathbf{x}_1 = \lambda_1 \mathbf{x}_1$ 和 $v_2 = A\mathbf{x}_2 = \lambda_2 \mathbf{x}_2$ ，即新的向量 v_i 是特征向量 \mathbf{x}_i 的缩放版本，缩放因子是对应的特征值 λ_i 。 $\mathbf{v}_1, \mathbf{v}_2$ 仍然是正交的，并且它们所张成的矩形的面积是 $|\lambda_1 \lambda_2|$ 。

鉴于在我们的例子中 $\mathbf{x}_1, \mathbf{x}_2$ 是正交归一的，我们可以直接计算出单位正方形的周长为 $2(1 + 1)$ 。映射特征向量后，新的矩形周长为 $2(|\lambda_1| + |\lambda_2|)$ 。因此，特征值绝对值的和告诉我们，在单位正方形经过变换矩阵 A 的变换后，其周长如何变化。



例4.9 谷歌的PageRank - 网页作为特征向量

谷歌使用矩阵 A 的最大特征值对应的特征向量来确定搜索页面的排名。

PageRank算法是由拉里·佩奇（Larry Page）和谢尔盖·布林（Sergey Brin）于1996年在斯坦福大学开发的，其核心理念是任何网页的重要性都可以通过链接到它的网页的重要性来近似。为此，他们将所有网站写成一个巨大的有向图，展示了哪些页面链接到哪些页面。PageRank通过计算指向网页 a_i 的页面数量来计算该网站 a_i 的权重（重要性） $x_i \geq 0$ 。此外，PageRank还考虑了链接到 a_i 的网站的重要性。

然后，用户的导航行为被建模为这个图的一个转移矩阵 A ，它告诉我们用户以什么（点击）概率会最终到达另一个网站。矩阵 A 具有这样的性质：对于网站的任何初始排名/重要性向量 x ，序列 x, Ax, A^2x, \dots 都会收敛到一个向量 x^* 。这个向量被称为PageRank，并满足 $Ax^* = x^*$ ，即它是 A 的一个特征向量（对应的特征值为1）。在对 x^* 进行归一化，使得 $\|x^*\| = 1$ 之后，我们可以将元素解释为概率。关于PageRank的更多细节和不同的视角可以在原始技术报告（Page et al., 1999）中找到。

< 上一章节

下一章节 >

4.1 矩阵的行列式与迹

4.3 Cholesky分解



4.3 Cholesky 分解

在机器学习中，我们经常遇到需要分解特殊类型矩阵的情况。对于正实数，我们有平方根运算，它可以将一个数分解为相同的因子，例如 $9 = 3 \cdot 3$ 。然而，对于矩阵，我们需要小心处理，确保我们在正数或正定矩阵上执行类似平方根的操作。

对于对称正定矩阵（见第3.2.3节），我们可以选择多种与平方根等效的操作。其中，Cholesky分解提供了一种在对称正定矩阵上进行类似平方根操作的方法，这在实践中非常有用。

定理4.18 (Cholesky分解)：一个对称正定矩阵 A 可以分解为两个矩阵的乘积，即 $A = LL^\top$ ，其中 L 是一个下三角矩阵，且其对角线元素为正。

具体来说，如果 A 是一个 $n \times n$ 的对称正定矩阵，那么存在一个唯一的下三角矩阵 L （称为 A 的 Cholesky 因子），其对角线元素为正，使得 $A = LL^\top$ 。

Cholesky分解的矩阵形式如下：

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & \cdots & l_{n1} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & l_{nn} \end{bmatrix} \quad (4.44)$$

这里， L 是下三角矩阵，其元素 l_{ij} （对于 $i > j$, $l_{ij} = 0$ ）可以通过递归方式计算得到，通常使用 Cholesky 算法的迭代步骤来求解。

译者注：Cholesky 分解在数值分析和统计中非常有用，特别是在处理大规模矩阵时，因为它提供了一种有效的方法来计算矩阵的逆和行列式，同时避免了直接计算可能导致的数值不稳定性。此外，Cholesky 分解还广泛应用于求解线性方程组、优化问题和蒙特卡洛模拟等领域。

例 4.10 Cholesky 分解



考虑一个对称正定矩阵 $A \in \mathbb{R}^{3 \times 3}$ 。我们对其进行Cholesky分解，即找到矩阵 L 使得 $A = LL^\top$ ，具体形式为：

$$A = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{21} & a_{22} & a_{32} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = LL^\top = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix} \quad (4.45)$$

将右侧矩阵相乘，我们得到：

$$A = \begin{bmatrix} l_{11}^2 & l_{11}l_{21} & l_{11}l_{31} \\ l_{11}l_{21} & l_{21}^2 + l_{22}^2 & l_{21}l_{31} + l_{22}l_{32} \\ l_{11}l_{31} & l_{21}l_{31} + l_{22}l_{32} & l_{31}^2 + l_{32}^2 + l_{33}^2 \end{bmatrix} \quad (4.46)$$

比较(4.45)的左侧和(4.46)的右侧，我们可以看到对角元素 l_{ii} 有一个简单的模式：

$$l_{11} = \sqrt{a_{11}}, \quad l_{22} = \sqrt{a_{22} - l_{21}^2}, \quad l_{33} = \sqrt{a_{33} - (l_{31}^2 + l_{32}^2)} \quad (4.47)$$

类似地，对于下三角元素 l_{ij} （其中 $i > j$ ），也存在一个重复的模式：

$$l_{21} = \frac{1}{l_{11}}a_{21}, \quad l_{31} = \frac{1}{l_{11}}a_{31}, \quad l_{32} = \frac{1}{l_{22}}(a_{32} - l_{31}l_{21}) \quad (4.48)$$

因此，我们为任何对称正定的 3×3 矩阵构造了Cholesky分解。关键之处在于，给定矩阵 A 的元素 a_{ij} 和之前已计算的 l_{ij} 值，我们可以反向计算出 L 的各个分量 l_{ij} 。

当然，以下是继续的中文翻译，保持对原文的忠实：

Cholesky 分解是机器学习底层数值计算中的一个重要工具。在这里，对称正定矩阵经常需要被处理，例如，多元高斯变量的协方差矩阵（见第6.5节）就是对称且正定的。这个协方差矩阵的Cholesky分解允许我们从高斯分布中生成样本。此外，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，Jimenez Rezende et al., 2014; Kingma 和 Welling, 2014）中的梯度时被大量利用。Cholesky 分解还允许我们非常高效地计算行列式。给定Cholesky分解 $A = LL^\top$ ，我们知道 $\det(A) = \det(L)\det(L^\top) = \det(L)^2$ 。由于 L 是一个三角矩阵，其行列式简单地等于其对角线元素的乘积，即 $\det(A) = \prod_i l_{ii}^2$ 。因此，许多数值软件包使用Cholesky分解来提高计算效率。

Cholesky 分解的这些特性使其成为处理大规模数据和复杂模型时不可或缺的工具，尤其是在机器学习和统计计算领域。通过减少计算复杂度和提高数值稳定性，**Cholesky** 分解促进了更高效和准确的算法开发。



< 上一章节

下一章节 >

4.2 特征值与特征向量

4.4 特征值分解与对角化



4.4 特征值分解与对角化

一个对角矩阵（Diagonal Matrix）是一个在所有非对角线上元素都为零的矩阵，即它们的形式为：

$$D = \begin{bmatrix} c_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & c_n \end{bmatrix}. \quad (4.49)$$

对角矩阵允许我们快速计算行列式、矩阵的幂以及逆矩阵。具体来说，对角矩阵的行列式等于其对角线上元素的乘积；矩阵的幂 D^k 是通过对每个对角元素求 k 次幂得到的；如果对角矩阵的所有对角元素都不为零，那么它的逆矩阵 D^{-1} 是其对角元素的倒数构成的矩阵。

在这一节中，我们将讨论如何将矩阵化为对角形式。这是我们在第2.7.2节讨论的基本变换和第4.2节讨论的特征值的一个重要应用。

回忆一下，如果存在一个可逆矩阵 P ，使得 $D = P^{-1}AP$ ，则称两个矩阵 A, D 是相似的（定义2.22）。更具体地说，我们将研究那些与对角矩阵 D 相似的矩阵 A ，其中对角矩阵 D 的对角线上包含矩阵 A 的特征值。

定义4.19（可对角化）：一个矩阵 $A \in \mathbb{R}^{n \times n}$ 是可对角化的，如果它与一个对角矩阵相似，即如果存在一个可逆矩阵 $P \in \mathbb{R}^{n \times n}$ ，使得 $D = P^{-1}AP$ 。

接下来，我们将看到，对角化一个矩阵 $A \in \mathbb{R}^{n \times n}$ 是表达相同线性映射但使用另一个基（见第2.6.1节）的一种方式，这个基将证明是由矩阵 A 的特征向量组成的。

译者注：对角化的过程实质上是找到一个新的坐标系（或基），在这个坐标系下，线性变换（由矩阵 A 表示）变得非常简单，即仅是对每个坐标轴（或基向量）进行伸缩变换，伸缩的比例由特征值给出。这种变换不仅简化了计算，还揭示了矩阵的固有性质，如特征值和特征向量的信息。

令 $A \in R^{n \times n}$, $\lambda_1, \lambda_2, \dots, \lambda_n$ 为一系列标量, $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$ 为分布在 R^n 空间上的向量。我们定义矩阵 $P := [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n]$ 并令矩阵 $D \in R^{n \times n}$ 为一个对角线为 $\lambda_1, \lambda_2, \dots, \lambda_n$ 的对角矩阵。于是我们可以得到:

$$AP = PD \quad (4.50)$$

当且仅当 $\lambda_1, \dots, \lambda_n$ 是矩阵 A 的特征值, 且 p_1, \dots, p_n 是 A 对应的特征向量时, 以下等式成立:

$$A = PDP^{-1} \quad (4.51)$$

我们可以观察到这一结论的成立是因为:

$$\begin{aligned} AP &= A[\mathbf{p}_1, \dots, \mathbf{p}_n] = [Ap_1, \dots, Ap_n] \\ P D &= [\mathbf{p}_1, \dots, \mathbf{p}_n] \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix} = [\lambda_1 \mathbf{p}_1, \dots, \lambda_n \mathbf{p}_n] \end{aligned} \quad (4.52)$$

因此, (4.50) 表明:

$$\begin{aligned} Ap_1 &= \lambda_1 p_1 \\ &\vdots \\ Ap_n &= \lambda_n p_n \end{aligned} \quad (4.53, 4.54)$$

所以, 矩阵 P 的列必须是 A 的特征向量。

对角化的定义要求 $P \in \mathbb{R}^{n \times n}$ 是可逆的, 即 P 具有满秩 (定理 4.3)。这要求我们有 n 个线性独立的特征向量 p_1, \dots, p_n , 即 p_i 构成 \mathbb{R}^n 的一个基。

定理 4.20 (特征分解)。一个 $n \times n$ 的方阵 $A \in \mathbb{R}^{n \times n}$ 可以被分解为

$$A = PDP^{-1} \quad (4.55)$$

其中 $P \in \mathbb{R}^{n \times n}$, D 是一个对角矩阵, 其对角线上的元素是 A 的特征值, 当且仅当 A 的特征向量构成 \mathbb{R}^n 的一个基。

1723798378994

图4.7 特征分解背后的直觉作为连续变换

左上角到左下角: $\$P^{-1}\$$ 执行了一个基变换 (此处在 $\$R^2\$$ 中绘制并表现为类似旋转的操作), 从标准基变换到特征基。左下角到右下角: $\$D\$$ 沿着重新映射的正交特

征向量进行缩放，形成一个椭圆。右下角到右上角：\$P\$撤销了基变换（表现为反向旋转），并恢复了原始的坐标系。

定理4.20意味着只有非缺陷矩阵才能被对角化，且\$P\$的列是\$A\$的\$n\$个特征向量。对于对称矩阵，我们可以得到特征值分解的更强结果。

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直接来自谱定理4.15。此外，谱定理指出我们可以找到\$\mathbb{R}^n\$的一个正交归一化的特征向量基。这使得\$P\$成为一个正交矩阵，从而\$D = P^\top AP\$。

备注：矩阵的Jordan标准型提供了一种适用于缺陷矩阵的分解（Lang, 1987），但这超出了本书的范围。

特征值分解的图形表示

我们可以将矩阵的特征分解解释如下（也见图4.7）：设\$A\$是关于标准基\$e_i\$（蓝色箭头）的线性映射的变换矩阵。\$P^{-1}\$执行从标准基到特征基的基变换。然后，对角矩阵\$D\$沿着这些轴通过特征值\$\lambda_i\$缩放向量。最后，\$P\$将这些缩放后的向量转换回标准/规范坐标，得到\$\lambda_i p_i\$。

例4.11（特征分解）

让我们计算\$A = \frac{1}{2} \begin{bmatrix} 5 & -2 \\ -2 & 5 \end{bmatrix}\$的特征分解。

步骤1：计算特征值和特征向量。

\$A\$的特征多项式是

$$\begin{aligned} \det(A - \lambda I) &= \det \left(\begin{bmatrix} \frac{5}{2} - \lambda & -1 \\ -1 & \frac{5}{2} - \lambda \end{bmatrix} \right) \\ &= \left(\frac{5}{2} - \lambda \right)^2 - 1 = \lambda^2 - 5\lambda + \frac{21}{4} = (\lambda - \frac{7}{2})(\lambda - \frac{3}{2}). \end{aligned} \quad (4.56)$$

因此，\$A\$的特征值是\$\lambda_1 = \frac{7}{2}\$和\$\lambda_2 = \frac{3}{2}\$（特征多项式的根），并且相关联的（归一化）特征向量通过

$$Ap_1 = \frac{7}{2}p_1, \quad Ap_2 = \frac{3}{2}p_2 \quad (4.57)$$

得到

$$\mathbf{p}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \mathbf{p}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (4.58)$$

步骤2: 检查存在性。

特征向量 p_1, p_2 构成 \mathbb{R}^2 的一个基。因此， A 可以被对角化。

步骤3: 构造矩阵 P 以对角化 A 。

我们将 A 的特征向量收集到 P 中，使得

$$P = [p_1, p_2] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \quad (4.59)$$

然后有

$$P^{-1}AP = \begin{bmatrix} \frac{7}{2} & 0 \\ 0 & \frac{3}{2} \end{bmatrix} = \mathbf{D} \quad (4.60)$$

我们得到

$$A = PDP^{-1} \quad (4.61)$$

或者等价地（利用在这个例子中特征向量 p_1 和 p_2 形成一个正交归一基，所以 $P^{-1} = P^\top$ ）

$$\underbrace{\frac{1}{2} \begin{bmatrix} 5 & -2 \\ -2 & 5 \end{bmatrix}}_A = \underbrace{\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}}_P \underbrace{\begin{bmatrix} \frac{7}{2} & 0 \\ 0 & \frac{3}{2} \end{bmatrix}}_D \underbrace{\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}}_{P^{-1}} \quad (4.62)$$

对角矩阵 D 可以有效地进行幂运算。因此，我们可以通过特征分解（如果存在）来找到一个矩阵 $A \in \mathbb{R}^{n \times n}$ 的幂，使得

$$A^k = (PDP^{-1})^k = PD^kP^{-1} \quad (4.62)$$

计算 D^k 是高效的，因为我们可以单独对每个对角元素进行此操作。

- 假设特征分解 $A = PDP^{-1}$ 存在。那么，

$$\begin{aligned}\det(\mathbf{A}) &= \det(PDP^{-1}) = \det(P)\det(D)\det(P^{-1}) \\ &= \det(D) = \prod_i d_{ii}\end{aligned}\tag{4.63}$$

这允许我们高效地计算矩阵 \mathbf{A} 的行列式。

特征分解要求矩阵是方阵。对一般矩阵进行分解会很有用。在下一节中，我们将介绍一种更一般的矩阵分解技术，即奇异值分解。

< 上一章节

下一章节 >

4.3 Cholesky 分解

4.5 奇异值分解



4.5 奇异值分解

矩阵的奇异值分解（SVD）是线性代数中的一种核心矩阵分解方法。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，而不仅仅是方阵，并且它总是存在。此外，正如我们将在下文探讨的，矩阵 A 的 SVD，它代表了一个线性映射 $\Phi : V \rightarrow W$ ，量化了这两个线性空间底层几何之间的变化。我们推荐阅读 Kalman (1996) 以及 Roy 和 Banerjee (2014) 的工作，以更深入地了解 SVD 的数学原理。

定理 4.22 (SVD 定理) 设 $A \in \mathbb{R}^{m \times n}$ 是一个秩为 $r \in [0, \min(m, n)]$ 的矩形矩阵。 A 的 SVD 是一种形式如下的分解：

$$\begin{array}{c|c|c|c} & \textcolor{brown}{n} & & \\ \textcolor{brown}{m} & \boxed{A} & = & \textcolor{brown}{m} \quad \textcolor{brown}{m} \\ & & & \boxed{U} \end{array} \quad \begin{array}{c|c|c|c} & & \textcolor{brown}{n} & \\ & & \boxed{\Sigma} & \\ & & & \boxed{V^\top} \end{array}$$

(4.64)

其中， $U \in \mathbb{R}^{m \times m}$ 是一个正交矩阵，其列向量为 $u_i, i = 1, \dots, m$ ； $V \in \mathbb{R}^{n \times n}$ 也是一个正交矩阵，其列向量为 $v_j, j = 1, \dots, n$ 。此外， Σ 是一个 $m \times n$ 矩阵，其对角线元素 $\Sigma_{ii} = \sigma_i \geq 0$ ，且当 $i \neq j$ 时， $\Sigma_{ij} = 0$ 。

Σ 的对角线元素 $\sigma_i, i = 1, \dots, r$ 被称为奇异值； u_i 被称为左奇异向量；而 v_j （在原文中错误地标记为左奇异向量，实际上应为右奇异向量）被称为右奇异向量。按照惯例，奇异值是有序的，即 $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$ 。

奇异值矩阵 Σ 是唯一的，但需要注意。观察到 $\Sigma \in \mathbb{R}^{m \times n}$ 是矩形的。特别是， Σ 与 A 具有相同的矩阵尺寸。这意味着 Σ 有一个包含奇异值的对角子矩阵，并且需要额外的零填充。具体来说，如果 $m > n$ ，则矩阵 Σ 在行 n 之前具有对角结构，然后从 $n + 1$ 行到 m 行由零行向量组成，使得



$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n \\ 0 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & 0 \end{bmatrix} \quad (4.65)$$

如果 $m < n$, 则矩阵 Σ 在列 m 之前具有对角结构, 而从 $m + 1$ 列到 n 列由零列向量组成:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & \ddots & 0 & \vdots & & \vdots \\ 0 & 0 & \sigma_m & 0 & \dots & 0 \end{bmatrix} \quad (4.66)$$

备注。对于任何矩阵 $A \in \mathbb{R}^{m \times n}$, 其SVD都是存在的。

1723803492248

图4.8

4.5.1 奇异值分解的图形表示

奇异值分解 (SVD) 提供了几何直观性来描述变换矩阵 A 。接下来, 我们将讨论 SVD 作为在基上顺序执行的线性变换。在示例 4.12 中, 我们将 SVD 的变换矩阵应用于 \mathbb{R}^2 中的一组向量, 这使我们能够更清晰地看到每个变换的效果。

矩阵的 SVD 可以被解释为相应线性映射 (回顾第 2.7.1 节) $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 分解为三个操作; 见图 4.8。SVD 的直观理解在表面上与我们的特征分解直观理解具有相似的结构, 见图 4.7。广义而言, SVD 通过 V^\top 进行基变换, 随后通过奇异值矩阵 Σ 进行尺度变换和维度增加 (或减少)。最后, 它通过 U 进行第二次基变换。SVD 包含了许多重要的细节和注意事项, 因此我们将更详细地回顾我们的直观理解。

1. 矩阵 V 在域 \mathbb{R}^n 中从基 \tilde{B} (在图 4.8 的左上角由红色和橙色向量 v_1 和 v_2 表示) 变换到标准基 B 。 $V^\top = V^{-1}$ 从基 \tilde{B} 变换到基 \tilde{B} 。现在, 红色和橙色向量与图 4.8 左下角的规范基对齐。
2. 将坐标系更改为 \tilde{B} 后, Σ 通过奇异值 σ_i 缩放新的坐标 (并增加或删除维度), 即 Σ 是相对于 \tilde{B} 和 \tilde{C} 的 Φ 的变换矩阵, 在图 4.8 的右下角, 由红色和橙色向量被拉伸并位于 e_1 - e_2 平面上 (现在嵌入在第三个维度中) 来表示。

3. U 在陪域 \mathbb{R}^m 中从基 \bar{C} 变换到 \mathbb{R}^m 的规范基，这表现为红色和橙色向量从 $e_1 - e_2$ 平面旋转出来，如图4.8的右上角所示。

SVD在域和陪域中都表示了基的变换。这与同一线性空间内操作的特征分解形成对比，在特征分解中，应用相同的基变换然后撤销它。SVD的特殊之处在于，这两个不同的基同时通过奇异值矩阵 Σ 相互关联。

例4.12 向量与SVD

考虑一个向量方格 $\mathcal{X} \in \mathbb{R}^2$ 的映射，这些向量适合位于以原点为中心的 2×2 大小的盒子中。使用标准基，我们使用以下公式映射这些向量：

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 1 & -0.8 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \mathbf{U}\Sigma\mathbf{V}^\top \\ &= \begin{bmatrix} -0.79 & 0 & -0.62 \\ 0.38 & -0.78 & -0.49 \\ -0.48 & -0.62 & 0.62 \end{bmatrix} \begin{bmatrix} 1.62 & 0 \\ 0 & 1.0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -0.78 & 0.62 \\ -0.62 & -0.78 \end{bmatrix} \end{aligned} \quad (4.67)$$

我们从一组以网格形式排列的向量 χ （彩色点；见图4.9的左上角面板）开始。然后，我们应用 $\mathbf{V}^\top \in \mathbb{R}^{2 \times 2}$ ，它旋转了 \mathcal{X} 。旋转后的向量显示在图4.9的左下角面板中。现在，我们使用奇异值矩阵 Σ 将这些向量映射到陪域 \mathbb{R}^3 （见图4.9的右下角面板）。注意，所有向量都位于 $x_1 - x_2$ 平面上。第三个坐标始终为0。 $x_1 - x_2$ 平面上的向量已被奇异值拉伸。

向量 χ 通过 \mathbf{A} 直接映射到陪域 \mathbb{R}^3 等于通过 $\mathbf{U}\Sigma\mathbf{V}^\top$ 对 \mathcal{X} 进行变换，其中 U 在陪域 \mathbb{R}^3 内进行旋转，使得映射后的向量不再局限于 $x_1 - x_2$ 平面；它们仍然位于一个平面上，如图4.9的右上角面板所示。

1723803798438

图4.9 SVD与向量映射（用圆盘表示）。各面板遵循与图4.8相同的逆时针结构

4.5.2 奇异值分解（SVD）的构建

接下来，我们将讨论为什么奇异值分解（SVD）存在，并详细展示如何计算它。一般矩阵的SVD与方阵的特征分解有一些相似之处。

注释：比较对称正定（SPD）矩阵的特征分解

$$S = S^\top = PDP^\top$$

(4.68) 与相应的SVD

$$S = U\Sigma V^\top.$$

(4.69)

如果我们设置

$$U = P = V, \quad D = \Sigma,$$

(4.70)

◇

我们可以看到，SPD矩阵的SVD就是其特征分解。

接下来，我们将探讨为什么定理4.22成立以及SVD是如何构建的。计算 $A \in \mathbb{R}^{m \times n}$ 的SVD等价于找到陪域 \mathbb{R}^m 和定义域 \mathbb{R}^n 的两组正交归一基 $U = (\mathbf{u}_1, \dots, \mathbf{u}_m)$ 和 $V = (v_1, \dots, v_n)$ 。从这些有序基中，我们将构建矩阵 U 和 V 。

我们的计划是从构建右奇异向量的正交归一集合 $v_1, \dots, v_n \in \mathbb{R}^n$ 开始。然后，我们构建左奇异向量的正交归一集合 $u_1, \dots, u_m \in \mathbb{R}^m$ 。之后，我们将两者联系起来，并要求在 A 的变换下保持 v_i 的正交性。这很重要，因为我们知道 Av_i 形成的集合是正交向量。然后，我们将通过标量因子对这些图像进行归一化，这些标量因子最终将是奇异值。

让我们从构建右奇异向量开始。谱定理（定理4.15）告诉我们，对称矩阵的特征向量形成一个正交归一基（ONB），这也意味着它可以被对角化。此外，根据定理4.14，我们可以从任何矩形矩阵 $A \in \mathbb{R}^{m \times n}$ 构造一个对称、半正定矩阵 $A^\top A \in \mathbb{R}^{n \times n}$ 。因此，我们总是可以对 $A^\top A$ 进行对角化，并得到



$$A^\top A = PDP^\top = P \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} P^\top,$$

(4.71)

其中 P 是一个正交矩阵，由正交归一的特征基组成。 $\lambda_i \geq 0$ 是 $A^\top A$ 的特征值。假设 A 的SVD存在，并将(4.64)代入(4.71)，得到

$$A^\top A = (U\Sigma V^\top)^\top (U\Sigma V^\top) = V\Sigma^\top U^\top U\Sigma V^\top,$$

(4.72)

其中 U, V 是正交矩阵。因此，由于 $U^\top U = I$ ，我们得到

$$A^\top A = V\Sigma^\top \Sigma V^\top = V \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n^2 \end{bmatrix} V^\top.$$

(4.73)

现在比较(4.71)和(4.73)，我们可以确定

$$\begin{aligned} V^\top &= P^\top, \\ \sigma_i^2 &= \lambda_i. \end{aligned}$$

(4.74) (4.75)

因此，组成 P 的 $A^\top A$ 的特征向量是 A 的右奇异向量 V （见(4.74)）。 $A^\top A$ 的特征值是 Σ 的奇异值的平方（见(4.75)）。

为了得到左奇异向量 U ，我们遵循类似的程序。我们首先计算对称矩阵 $AA^\top \in \mathbb{R}^{m \times m}$ （而不是之前的 $A^\top A \in \mathbb{R}^{n \times n}$ ）的SVD。 A 的SVD得到

(4.76a)

$$\begin{aligned} AA^\top &= (U\Sigma V^\top)(U\Sigma V^\top)^\top = U\Sigma V^\top V\Sigma^\top U^\top \\ &= U \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_m^2 \end{bmatrix} U^\top. \end{aligned}$$

(4.76b) 的翻译：

谱定理告诉我们， $\mathbf{A}\mathbf{A}^\top = \mathbf{S}\mathbf{D}\mathbf{S}^\top$ 可以被对角化，我们可以找到 $\mathbf{A}\mathbf{A}^\top$ 的特征向量的一个正交归一基（ONB），这些特征向量被收集在 \mathbf{S} 中。 $\mathbf{A}\mathbf{A}^\top$ 的正交归一特征向量是左奇异向量 \mathbf{U} ，并在奇异值分解（SVD）的值域中形成一个正交归一基。

接下来是矩阵 Σ 的结构问题。由于 $\mathbf{A}\mathbf{A}^\top$ 和 $\mathbf{A}^\top\mathbf{A}$ 有相同的非零特征值（见第106页），因此在两种情况的SVD中， Σ 矩阵的非零元素必须相同。最后一步是将我们到目前为止所触及的所有部分连接起来。我们在 \mathbf{V} 中有一个右奇异向量的正交归一集。为了完成SVD的构建，我们将它们与正交归一向量 \mathbf{U} 连接起来。为了达到这个目标，我们使用了一个事实，即 \mathbf{A} 下的 \mathbf{v}_i 的像也必须是正交的。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 $\mathbf{A}\mathbf{v}_i$ 和 $\mathbf{A}\mathbf{v}_j$ 之间的内积必须为0，对于 $i \neq j$ 。对于任何两个正交的特征向量 $\mathbf{v}_i, \mathbf{v}_j, i \neq j$ ，都有

$$(\mathbf{A}\mathbf{v}_i)^\top (\mathbf{A}\mathbf{v}_j) = \mathbf{v}_i^\top (\mathbf{A}^\top \mathbf{A})\mathbf{v}_j = \mathbf{v}_i^\top (\lambda_j \mathbf{v}_j) = \lambda_j \mathbf{v}_i^\top \mathbf{v}_j = 0$$

(4.77)

对于 $m \geq r$ 的情况， $\{\mathbf{A}\mathbf{v}_1, \dots, \mathbf{A}\mathbf{v}_r\}$ 是 \mathbb{R}^m 中一个 r 维子空间的基础。为了完成 SVD 的构建，我们需要左奇异向量是正交归一的：我们将右奇异向量 $\mathbf{A}\mathbf{v}_i$ 的像进行归一化，得到

(4.78)

$$u_i := \frac{\mathbf{A}\mathbf{v}_i}{\|\mathbf{A}\mathbf{v}_i\|} = \frac{1}{\sqrt{\lambda_i}} \mathbf{A}\mathbf{v}_i = \frac{1}{\sigma_i} \mathbf{A}\mathbf{v}_i,$$

其中最后一个等式是从(4.75)和(4.76b)中得出的，表明 $\mathbf{A}\mathbf{A}^\top$ 的特征值满足 $\sigma_i^2 = \lambda_i$ 。因此， $\mathbf{A}^\top\mathbf{A}$ 的特征向量（我们知道它们是右奇异向量 \mathbf{v}_i ）和它们在 \mathbf{A} 下的归一化像（左奇异向量 u_i ）形成了两个通过奇异值矩阵 Σ 连接的自洽正交归一基（ONBs）。

让我们重新排列(4.78)以得到奇异值方程

$$\mathbf{A}\mathbf{v}_i = \sigma_i u_i, \quad i = 1, \dots, r.$$

(4.79)

这个方程与特征值方程(4.25)非常相似，但左右两边的向量并不相同。

对于 $n < m$ ，(4.79) 仅对 $i \leq n$ 成立，但(4.79)对 $i > n$ 的 u_i 没有说明。然而，我们通过构造知道它们是正交归一的。相反，对于 $m < n$ ，(4.79) 仅对 $i \leq m$ 成立。对于 $i > m$ ，我们有 $\mathbf{A}\mathbf{v}_i = \mathbf{0}$ ，并且我们仍然知道 \mathbf{v}_i 形成一个正交归一集。

这意味着SVD还提供了 A 的核（零空间）的一个正交归一基，即满足 $Ax = 0$ 的向量集（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

$$AV = U\Sigma,$$

(4.80)

其中 Σ 与 A 有相同的维度，并且在第1到第 r 行具有对角结构。因此，右乘 V^\top 得到 $A = U\Sigma V^\top$ ，这就是 A 的SVD。

例 4.13 (计算奇异值分解)

让我们找到矩阵

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

的奇异值分解。

奇异值分解 (SVD) 要求我们计算右奇异向量 v_j ，奇异值 σ_k ，以及左奇异向量 u_i 。

步骤 1：右奇异向量作为 $A^\top A$ 的特征基。

首先，我们计算

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

接着，我们通过 $A^\top A$ 的特征值分解来计算奇异值和右奇异向量 v_j 。 $A^\top A$ 的特征值分解为

$$A^\top A = \begin{bmatrix} \frac{5}{\sqrt{30}} & 0 & \frac{-1}{\sqrt{6}} \\ \frac{-2}{\sqrt{30}} & \frac{1}{\sqrt{5}} & \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} 6 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{5}{\sqrt{30}} & \frac{-2}{\sqrt{30}} & \frac{1}{\sqrt{30}} \\ 0 & \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{-1}{\sqrt{6}} & \frac{-2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{bmatrix} = PDP^\top$$

从中我们可以得到右奇异向量为 P 的列向量，即



$$V = P = \begin{bmatrix} \frac{5}{\sqrt{30}} & 0 & \frac{-1}{\sqrt{6}} \\ \frac{-2}{\sqrt{30}} & \frac{1}{\sqrt{5}} & \frac{-2}{\sqrt{6}} \\ \frac{\sqrt{30}}{30} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{6}} \end{bmatrix}$$

步骤 2: 奇异值矩阵。

奇异值 σ_i 是 $A^\top A$ 特征值的平方根，我们可以直接从 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 的大小相同，我们得到

$$\Sigma = \begin{bmatrix} \sqrt{6} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

步骤 3: 左奇异向量作为右奇异向量在 A 下的归一化像。

我们通过计算右奇异向量在 A 下的像并将它们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

(4.86)

$$\mathbf{u}_1 = \frac{1}{\sigma_1} A \mathbf{v}_1 = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{5}{\sqrt{30}} \\ \frac{-2}{\sqrt{30}} \\ \frac{1}{\sqrt{30}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{5}} \\ -\frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix},$$

$$\mathbf{u}_2 = \frac{1}{\sigma_2} A \mathbf{v}_2 = \frac{1}{1} \begin{bmatrix} 1 & 0 & 1 \\ -2 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix},$$

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2] = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}.$$

(4.87)

(4.88)

注意到在一台计算机上执行这里描述的操作时数值计算性能并不好，因此矩阵 A 的 SVD 分解通常是在没有对 $A^\top A$ 进行特征值分解并重排的情况下计算的。

4.5.3 特征值分解与奇异值分解

让我们考虑特征值分解 $A = PDP^{-1}$ 和奇异值分解 (SVD) $A = U\Sigma V^\top$, 并回顾前几节的核心内容。

- 对于任意矩阵 $R^{m \times n}$, 奇异值分解总是存在的。而特征值分解仅对方阵 $\mathbb{R}^{n \times n}$ 定义, 并且仅当我们能找到 R^n 的特征向量基时才存在。
- 在特征值分解矩阵 P 中的向量不一定是正交的, 即基变换不仅仅是旋转和缩放。另一方面, 在SVD中的矩阵 U 和 V 的向量是正交归一的, 因此它们确实表示旋转。
- 特征值分解和奇异值分解都是三个线性映射的组合:
 1. 域中的基变换
 2. 对每个新基向量的独立缩放以及从域到陪域的映射
 3. 陪域中的基变换

1723805847048

图4.10 四部电影的三人电影评分及其SVD分解

特征值分解与奇异值分解之间的一个关键区别在于, 在奇异值分解中, 域和陪域可以是不同维度的线性空间。

- 在奇异值分解中, 左奇异向量矩阵 U 和右奇异向量矩阵 V 通常不是彼此的逆(它们在不同的线性空间中进行基变换)。在特征值分解中, 基变换矩阵 P 和 P^{-1} 是彼此的逆。
- 在奇异值分解中, 对角矩阵 Σ 中的元素都是实数且非负, 这在特征值分解的对角矩阵中通常不成立。
- 奇异值分解和特征值分解通过它们的投影密切相关:
 - A 的左奇异向量是 AA^\top 的特征向量。
 - A 的右奇异向量是 $A^\top A$ 的特征向量。
 - A 的非零奇异值是 AA^\top 和 $A^\top A$ 的非零特征值的平方根。
- 对于对称矩阵 $A \in \mathbb{R}^{n \times n}$, 根据谱定理 4.15, 其特征值分解和奇异值分解是相同的。



例 4.14 (在电影评分和消费者中寻找结构)

让我们通过分析人们对电影偏好的数据，对奇异值分解 (**SVD**) 进行实际解释。考虑三位观众（阿里、贝阿特丽克丝、钱德拉）对四部不同电影（星球大战、银翼杀手、天使爱美丽、美味）的评分。他们的评分介于 0（最差）和 5（最好）之间，并编码在一个数据矩阵 $A \in \mathbb{R}^{4 \times 3}$ 中，如图 4.10 所示。每一行代表一部电影，每一列代表一个用户。因此，每部电影评分的列向量（每位观众一个）分别是 $x_{\text{Ali}}, x_{\text{Beatrix}}, x_{\text{Chandra}}$ 。

使用 **SVD** 对 A 进行因式分解，可以帮助我们捕捉人们如何对电影进行评分的关系，特别是如果存在将哪些人喜欢哪些电影联系起来的结构。将 **SVD** 应用到我们的数据矩阵 A 上，我们做出了一系列假设：

1. 所有观众都使用相同的线性映射来一致地评分电影。
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，我们假设任何观众对特定电影的偏好都可以表示为 v_j 的线性组合。同样地，任何电影的喜爱程度也可以表示为 u_i 的线性组合。**SVD** 域中的向量可以被解释为“典型观众空间”中的观众，而 **SVD** 陪域中的向量则相应地被解释为“典型电影空间”中的电影。让我们检查我们电影-用户矩阵的 **SVD**。第一个左奇异向量 u_1 在两部科幻电影上有较大的绝对值，并且具有较大的第一个奇异值（图 4.10 中的红色阴影）。因此，这将一类具有特定电影集（科幻主题）的用户进行了分组。类似地，第一个右奇异向量 v_1 显示阿里和贝阿特丽克丝具有较大的绝对值，他们给科幻电影打了高分（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。

同样地， u_2 似乎捕捉了法国艺术电影的主题，而 v_2 则表明钱德拉接近于这种电影的理想化爱好者。一个理想化的科幻爱好者是纯粹的，只喜欢科幻电影，所以科幻爱好者 v_1 除了科幻主题的电影外，对其他一切都打零分——这是由奇异值矩阵 Σ 的对角子结构所隐含的逻辑。因此，一部特定的电影通过它如何（线性地）分解为典型电影来表示。同样地，一个人也会通过他们如何（通过线性组合）分解为电影主题来表示。

值得简要讨论一下**SVD**（奇异值分解）的术语和约定，因为文献中存在不同的版本。尽管这些差异可能会令人困惑，但数学本质是不变的。

- 为了方便表示和抽象，我们使用一种**SVD**表示法，其中**SVD**被描述为具有两个方形的左奇异向量矩阵和右奇异向量矩阵，但奇异值矩阵是非方形的。我们对**SVD**的定义 (4.64) 有时被称为“完全**SVD** (fullSVD) ”。
- 一些作者以略有不同的方式定义**SVD**，并关注方形奇异矩阵。然后，对于 $A \in \mathbb{R}^{m \times n}$ 且 $m \geq n$,

(4.89)

$$A_{m \times n} = U_{m \times n} \sum_{n \times n} V_{n \times n}^\top.$$

这里， $A_{m \times n}$ 是原始矩阵， $U_{m \times n}$ 是一个 $m \times n$ 的矩阵，其列是左奇异向量； $\sum_{n \times n}$ （注意这里通常使用大写希腊字母 Σ 表示，但在这里用求和符号的简化形式表示）是一个 $n \times n$ 的对角矩阵，其对角线上的元素是奇异值； $V_{n \times n}^\top$ 是 $V_{n \times n}$ 的转置， $V_{n \times n}$ 是一个 $n \times n$ 的矩阵，其列是右奇异向量。这种表示法在某些文献中也被使用，尽管它并不是**SVD**的唯一表示方式。

有时，这种表述被称为简化**SVD**（例如，Datta (2010)）或仅称为**SVD**（例如，Press et al. (2007)）。这种替代格式仅仅改变了矩阵的构建方式，但保留了**SVD**的数学结构不变。这种替代表述的便利之处在于 Σ 是对角矩阵，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**。
- 可以定义一个秩为 r 的矩阵 A 的**SVD**，使得 U 是一个 $m \times r$ 矩阵， Σ 是一个 $r \times r$ 的对角矩阵，而 V 是一个 $n \times r$ 矩阵。这种构造与我们的定义非常相似，并确保了对角矩阵 Σ 的对角线上只有非零元素。这种替代记法的主要便利之处在于 Σ 是对角矩阵，就像特征值分解一样。
- 实际上，对于 A 的**SVD** 仅适用于 $m \times n$ 矩阵且 $m > n$ 的限制是不必要的。当 $m < n$ 时，**SVD** 分解将产生一个 Σ ，其零列的数量多于行的数量，因此，奇异值 $\sigma_{m+1}, \dots, \sigma_n$ 都是 0。

SVD 在机器学习中有多种应用，从曲线拟合中的最小二乘问题到线性方程组的求解。这些应用利用了**SVD** 的各种重要属性，包括它与矩阵秩的关系，以及它用低秩矩阵近似给定秩矩阵的能力。用**SVD** 替换矩阵通常具有使计算对数值舍入误差更鲁棒的优点。

势。正如我们将在下一节中探讨的那样，SVD能够以原则性的方式用“更简单”的矩阵近似矩阵，从而开辟了从降维和主题建模到数据压缩和聚类的各种机器学习应用。

< 上一章节

下一章节 >

4.4 特征值分解与对角化

4.6 矩阵近似





4.6 矩阵近似

我们将SVD视为一种将 $A = U\Sigma V^\top \in \mathbb{R}^{m \times n}$ 分解为三个矩阵乘积的方法，其中 $U \in \mathbb{R}^{m \times m}$ 和 $V \in \mathbb{R}^{n \times n}$ 是正交矩阵，而 Σ 在其主对角线上包含奇异值。现在，我们不进行完整的SVD分解，而是研究SVD如何允许我们将矩阵 A 表示为更简单（低秩）的矩阵 A_i 之和，这种表示法构成了一种矩阵近似方案，其计算成本低于完整的SVD。

我们构造一个秩为1的矩阵 $A_i \in \mathbb{R}^{m \times n}$ ，形式为

$$A_i := u_i v_i^\top$$

(4.90)

这是由 U 和 V 的第*i*个正交列向量的外积形成的。图4.11展示了巨石阵的图像，该图像可以由一个矩阵 $A \in \mathbb{R}^{1432 \times 1910}$ 来表示，以及根据(4.90)定义的一些外积 A_i 。

1723806749553

图4.11 使用SVD进行图像处理。(a) 原始灰度图像是一个 $1,432 \times 1,910$ 的矩阵，其值介于0（黑色）和1（白色）之间。(b)-(f) 秩为1的矩阵 A_1, \dots, A_5 及其对应的奇异值 $\sigma_1, \dots, \sigma_5$ 。每个秩为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形成的。

一个秩为*r*的矩阵 $A \in \mathbb{R}^{m \times n}$ 可以表示为秩为1的矩阵 A_i 之和，即

$$A = \sum_{i=1}^r \sigma_i u_i v_i^\top = \sum_{i=1}^r \sigma_i A_i$$

(4.91)

其中，外积矩阵 A_i 由第*i*个奇异值 σ_i 加权。我们可以理解为什么(4.91)成立：奇异值矩阵 Σ 的对角结构仅将匹配的左奇异向量和右奇异向量 $u_i v_i^\top$ 相乘，并按相应的奇异值 σ_i 进行缩放。所有 $i \neq j$ 的项 $\sum_i j u_i v_j^\top$ 都消失，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，我们引入了秩为1的矩阵 A_i 。我们将 r 个单独的秩为1的矩阵相加，以得到一个秩为 r 的矩阵 A ；参见(4.91)。如果求和不是遍历所有矩阵 $A_i, i = 1, \dots, r$ ，而是仅到某个中间值 $k < r$ ，则我们得到一个秩为 k 的近似

(4.92)

$$\hat{A}(k) := \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^\top = \sum_{i=1}^k \sigma_i A_i$$

其中， $\text{rk}(\hat{A}(k)) = k$ 。图4.12展示了巨石阵原始图像 A 的低秩近似 $\hat{A}(k)$ 。在秩为5的近似中，岩石的形状变得越来越清晰可辨。虽然原始图像需要 $1,432 \cdot 1,910 = 2,735,120$ 个数字来表示，但秩为5的近似仅需要存储五个奇异值以及五个左奇异向量和右奇异向量（每个都是1,432维和1,910维），总共需要 $5 \cdot (1,432 + 1,910 + 1) = 16,715$ 个数字——仅为原始数据的0.6%多一点。

为了测量矩阵 A 与其秩为 k 的近似 $\hat{A}(k)$ 之间的差异（误差），我们需要范数的概念。在3.1节中，我们已经使用了向量的范数来衡量向量的长度。类似地，我们也可以定义矩阵的范数。



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，其中近似由 $\tilde{A}(k) = \sum_{i=1}^k \sigma_i A_i$ 给出。

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

$$\|A\|_2 := \max_x \frac{\|Ax\|_2}{\|x\|_2}.$$

(4.93)

我们在矩阵范数（左侧）中引入了下标的符号，这与向量的 Euclid 范数（右侧）类似，后者有下标2。谱范数 (4.93) 决定了任何向量 x 在乘以 A 之后可能达到的最大长度。

定理 4.24。矩阵 A 的谱范数是其最大的奇异值 σ_1 。

此定理的证明我们留作练习。

Eckart-Young 定理 4.25 (Eckart 和 Young, 1936)。考虑一个秩为 r 的矩阵 $A \in \mathbb{R}^{m \times n}$ ，以及一个秩为 k 的矩阵 $B \in \mathbb{R}^{m \times n}$ 。对于任意 $k \leq r$ ，且 $\hat{A}(k) =$



$\sum_{i=1}^k \sigma_i u_i v_i^\top$, 则 (4.94)

$$\begin{aligned}\widehat{\mathbf{A}}(k) &= \operatorname{argmin}_{\operatorname{rk}(\mathbf{B})=k} \|\mathbf{A} - \mathbf{B}\|_2, \\ \|\mathbf{A} - \widehat{\mathbf{A}}(k)\|_2 &= \sigma_{k+1}.\end{aligned}$$

(4.95)

Eckart-Young 定理明确指出了我们使用秩为 k 的近似来近似 \mathbf{A} 时引入的误差量。我们可以将使用 SVD 获得的秩 k 近似解释为全秩矩阵 \mathbf{A} 在秩至多为 k 的矩阵构成的低维空间上的投影。在所有可能的投影中, SVD 使 \mathbf{A} 与任何秩 k 近似之间的误差 (就谱范数而言) 最小化。

我们可以通过回顾一些步骤来理解为什么 (4.95) 应该成立。我们观察到, $\mathbf{A} - \widehat{\mathbf{A}}(k)$ 之间的差异是一个矩阵, 它包含了剩余秩为 1 的矩阵的总和。

(4.96)

$$\mathbf{A} - \widehat{\mathbf{A}}(k) = \sum_{i=k+1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top.$$

根据定理 4.24, 我们立即得到 σ_{k+1} 作为差异矩阵的谱范数。现在让我们更仔细地看一下 (4.94)。如果我们假设存在另一个矩阵 \mathbf{B} , 其秩 $\operatorname{rk}(\mathbf{B}) \leq k$, 使得

(4.97)

$$\|\mathbf{A} - \mathbf{B}\|_2 < \|\mathbf{A} - \widehat{\mathbf{A}}(k)\|_2,$$

那么存在一个至少 $(n - k)$ -维的零空间 $Z \subseteq \mathbb{R}^n$, 使得对于任意 $x \in Z$, 都有 $Bx = 0$ 。由此可得

$$\|\mathbf{A}x\|_2 = \|(A - B)x\|_2,$$

(4.98)

并使用柯西-施瓦茨不等式 (3.17) 的一个版本, 该版本涵盖了矩阵的范数, 我们得到

$$\|\mathbf{A}x\|_2 \leq \|\mathbf{A} - \mathbf{B}\|_2 \|x\|_2 < \sigma_{k+1} \|x\|_2.$$

(4.99)

然而，存在一个 $(k+1)$ -维子空间，其中 $\|Ax\|_2 \geq \sigma_{k+1}\|x\|_2$ ，这个子空间由 A 的右奇异向量 $v_j, j \leq k+1$ 张成。将这两个空间的维度相加会得到一个大于 n 的数，因为这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-零度定理（定理2.24）相矛盾。

Eckart-Young定理意味着我们可以使用SVD以有原则且最优（在谱范数意义上）的方式将秩为 r 的矩阵 A 减少到秩为 k 的矩阵 \hat{A} 。我们可以将 A 由秩为 k 的矩阵近似视为一种有损压缩的形式。因此，矩阵的低秩近似出现在许多机器学习应用中，例如图像处理、噪声过滤和不适定问题的正则化。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发挥着关键作用。

例4.15（在电影评分和消费者中寻找结构（续））

回到我们的电影评分示例中，我们现在可以应用低秩近似的概念来近似原始数据矩阵。回想一下，我们的第一个奇异值捕捉了电影中科幻主题和科幻爱好者的概念。因此，通过仅使用电影评分矩阵的秩-1分解中的第一个奇异值项，我们得到预测的评分

$$\begin{bmatrix} -0.6710 \\ -0.7197 \\ -0.0939 \end{bmatrix} A_1 = u_1 v_1^\top = \begin{bmatrix} -0.7367 & -0.6515 & -0.1811 \end{bmatrix} \quad (4.100a) \quad \begin{bmatrix} -0.1515 \end{bmatrix}$$

(4.100b)

$$= \begin{bmatrix} 0.4943 & 0.4372 & 0.1215 \\ 0.5302 & 0.4689 & 0.1303 \\ 0.0692 & 0.0612 & 0.0170 \\ 0.1116 & 0.0987 & 0.0274 \end{bmatrix}.$$

这个第一个秩-1近似 A_1 是富有洞察力的：它告诉我们阿里和贝阿特丽克斯喜欢科幻电影，如《星球大战》和《银翼杀手》（条目值>0.4），但未能捕捉到钱德拉对其他电影的评分。这并不奇怪，因为钱德拉喜欢的电影类型没有被第一个奇异值捕捉到。第二个奇异值为我们提供了这些电影主题爱好者的更好的秩-1近似：

(4.101a)



$$\begin{aligned} \mathbf{A}_2 &= \mathbf{u}_2 \mathbf{v}_2^\top = \begin{bmatrix} 0.0236 \\ 0.2054 \\ -0.7705 \\ -0.6030 \end{bmatrix} \begin{bmatrix} 0.0852 & 0.1762 & -0.980 \end{bmatrix} \\ &= \begin{bmatrix} 0.0020 & 0.0042 & -0.0231 \\ 0.0175 & 0.0362 & -0.2014 \\ -0.0656 & -0.1358 & 0.7556 \\ -0.0514 & -0.1063 & 0.5914 \end{bmatrix}. \end{aligned}$$

(4.101b)

在这个第二个秩-1近似 \mathbf{A}_2 中，我们很好地捕捉到了钱德拉的评分和电影类型，但没有捕捉到科幻电影。这促使我们考虑秩-2近似 $\hat{\mathbf{A}}(2)$ ，其中我们结合了前两个秩-1近似

$$\hat{\mathbf{A}}(2) = \sigma_1 \mathbf{A}_1 + \sigma_2 \mathbf{A}_2 = \begin{bmatrix} 4.7801 & 4.2419 & 1.0244 \\ 5.2252 & 4.7522 & -0.0250 \\ 0.2493 & -0.2743 & 4.9724 \\ 0.7495 & 0.2756 & 4.0278 \end{bmatrix}.$$

(4.102)

$\hat{\mathbf{A}}(2)$ 与原始电影评分表相似

(4.103)

$$\mathbf{A} = \begin{bmatrix} 5 & 4 & 1 \\ 5 & 5 & 0 \\ 0 & 0 & 5 \\ 1 & 0 & 4 \end{bmatrix},$$

这表明我们可以忽略 \mathbf{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。这也意味着在我们示例中，电影主题/电影爱好者的整个空间是一个由科幻电影和法国艺术电影及其爱好者所跨越的二维空间。

1723808265139

图4.13 在机器学习中遇到的矩阵的计算方法演化



< 上一章节

下一章节 >

4.5 奇异值分解

4.7 矩阵的演化



4.7 矩阵的演化

在第2章和第3章中，我们介绍了线性代数和分析“系统发育”几何学的基础知识。在本章中，我们研究了矩阵和线性映射的基本特性。图4.13描绘了捕捉矩阵和线性映射之间关系的系统发育树（黑色箭头表示个体或“是……的子集”），以及我们可以在这些矩阵上执行的操作（蓝色表示群和派生操作）。我们考虑所有实数矩阵 $A \in \mathbb{R}^{n \times m}$ 。对于非方阵（其中 $n \neq m$ ），如本章所见，奇异值分解（SVD）总是存在的。我们专注于方阵 $A \in \mathbb{R}^{n \times n}$ ，行列式告诉我们一个方阵是否具有逆矩阵，即它是否属于正则、可逆矩阵的类别。如果 $n \times n$ 方阵具有 n 个线性独立的特征向量，则该矩阵是“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，这类矩阵不能对角化。

非奇异矩阵和非缺陷矩阵是不同的。例如，旋转矩阵将是可逆的（行列式不为零），但在实数范围内不可对角化（特征值不一定是实数）。

我们进一步深入探讨非缺陷型方阵 $n \times n$ 矩阵的分支。如果满足条件 $A^\top A = AA^\top$ ，则矩阵 A 是正规的（注：原文中的 $A^\top A_\top$ 应为 $A^\top A$ ，且下划线部分应为格式错误，已更正）。此外，如果满足更严格的条件 $A^\top A = AA^\top = I$ ，则称 A 为正交矩阵（见定义3.8）。正交矩阵的集合是正则（可逆）矩阵的子集，并满足 $A^\top = A^{-1}$ 。

正规矩阵中经常遇到的一个子集是对称矩阵 $S \in \mathbb{R}^{n \times n}$ ，它们满足 $S = S^\top$ 。对称矩阵只有实数特征值。对称矩阵的一个子集是正定矩阵 P ，它们满足对于所有 $x \in \mathbb{R}^n \setminus \{0\}$ ，都有 $x^\top Px > 0$ 。在这种情况下，存在唯一的乔莱斯基分解（定理4.18）。正定矩阵只有正特征值，并且总是可逆的（即行列式不为零）。

对称矩阵的另一个子集是对角矩阵 D 。对角矩阵在乘法和加法下是封闭的，但不一定形成群（只有当所有对角元素都不为零，从而使矩阵可逆时，才成立）。一个特殊的对角矩阵是单位矩阵 I 。

4.6 矩阵近似

4.8 拓展阅读





4.8 扩展阅读

本章的大部分内容建立了基础数学，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支撑软件解决方案和几乎所有机器学习理论构建块层面上的核心。使用行列式、特征谱和特征空间对矩阵进行表征，为矩阵的分类和分析提供了基本特征和条件。这扩展到数据和涉及数据的映射的所有形式的表示，以及评估在这些矩阵上进行的计算操作的数值稳定性（Press et al., 2007）。

行列式是反转矩阵和“手动”计算特征值的基本工具。然而，对于几乎所有但不是最小的实例，通过高斯消元法进行的数值计算都优于行列式（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，可以根据行列式的符号直观地了解基的方向。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的坐标。同样，当我们计算或模拟随机事件时，矩阵分解方法（如楚列斯基分解）经常再次出现（Rubinstein和Kroese, 2016）。因此，楚列斯基分解使我们能够计算重参数化技巧，其中我们希望在随机变量上进行连续微分，例如在变分自编码器

（Jimenez Rezende et al., 2014; Kingma和Welling, 2014）中。

特征分解对于使我们能够提取表征线性映射的有意义和可解释的信息至关重要。

因此，特征分解构成了一类称为谱方法的机器学习算法的基础，这类算法对正定核进行特征分解。这些谱分解方法涵盖了统计数据分析的经典方法，例如：

- 主成分分析（PCA）（Pearson, 1901，也见第10章），它寻求一个低维子空间，该子空间能解释数据中的大部分变异性。
- 费舍尔判别分析（Fisher discriminant analysis），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和Chang, 1970）。

这些方法的计算效率通常来源于找到对称正半定矩阵的最佳k秩近似。谱方法的更现代例子有不同的起源，但每个例子都需要计算正定核的特征向量和特征值，如Isomap（Tenenbaum et al., 2000）、拉普拉斯特征映射（Laplacian eigenmaps）（Belkin和Niyogi, 2003）、海森特征映射（Hessian eigenmaps）（Donoho和Grimes, 2003）和谱聚类（Shi和Malik, 2000）。这些算法的核心计算通常基于低秩矩阵近似技术（Belabbas和Wolfe, 2009），正如我们在那里通过奇异值分解（SVD）所遇到的那样。



SVD允许我们发现与特征分解相同类型的一些信息。然而，SVD更普遍地适用于非方阵和数据表。当我们想要通过近似进行数据压缩时（例如，不存储 $n \times m$ 个值，而只存储 $(n + m)k$ 个值），或者当我们想要进行数据预处理（例如，去相关设计矩阵的预测变量）（Ormoneit et al., 2001）时，这些矩阵分解方法变得相关。SVD作用于矩阵，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更高维度的数组称为张量。事实证明，SVD是作用于此类张量的更一般分解族的一个特例（Kolda和Bader, 2009）。在张量上进行的类似SVD的操作和低秩近似，例如，有Tucker分解（Tucker, 1966）或CP分解（Carroll和Chang, 1970）。

出于计算效率的原因，SVD低秩近似在机器学习中经常被使用。这是因为它减少了我们可能需要在非常大的数据矩阵上执行的非零乘法操作的内存量和操作量（Trefethen和Bau III, 1997）。此外，低秩近似还用于处理可能包含缺失值的矩阵，以及用于有损压缩和降维（Moonen和De Moor, 1995; Markovsky, 2011）。

< 上一章节

下一章节 >

4.7 矩阵的演化

习题



习题

4.1 使用拉普拉斯展开（使用第一行）和萨鲁斯法则计算行列式

对于矩阵

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 0 & 2 & 4 \end{bmatrix}$$

4.2 高效地计算以下行列式:

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

4.3 计算以下矩阵的特征空间:

$$A := \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad B := \begin{bmatrix} -2 & 2 \\ 2 & 1 \end{bmatrix}$$

4.4 计算以下矩阵的所有特征空间:

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

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:

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

4.6 计算以下变换矩阵的特征空间。它们是否可对角化?

a. 对于



$$\mathbf{A} = \begin{bmatrix} 2 & 3 & 0 \\ 1 & 4 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

b. 对于

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

4.7 以下矩阵是否可对角化？如果是，确定它们的对角形式以及使变换矩阵对角化的基。如果不是，给出它们不可对角化的原因。

a.

$$A = \begin{bmatrix} 0 & 1 \\ -8 & 4 \end{bmatrix}$$

b.

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

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

$$\mathbf{A} = \begin{bmatrix} 5 & -6 & -6 \\ -1 & 4 & 2 \\ 3 & -6 & -4 \end{bmatrix}$$

4.8 找到矩阵的奇异值分解 (SVD) :

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix}$$

4.9 找到以下矩阵的奇异值分解:

$$\mathbf{A} = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}$$

4.10 找到以下矩阵的秩-1近似：

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

4.11 证明对于任意 $A \in \mathbb{R}^{m \times n}$, 矩阵 $A^\top A$ 和 AA^\top 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。

< 上一章节

4.8 拓展阅读

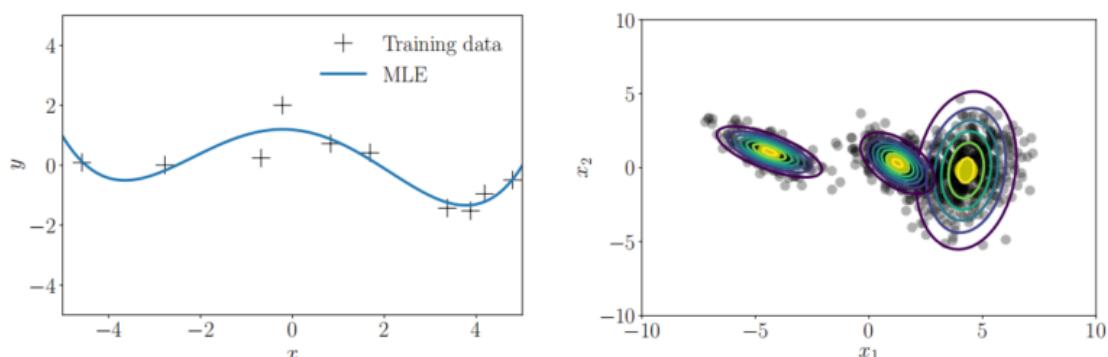


第五章 向量微积分

许多机器学习算法都在优化一个目标函数，即相对于一组模型参数进行优化，这些参数控制着模型解释数据的好坏。如何寻找好的参数可被表述为一个优化问题（见 8.2 节和 8.3 节）。优化的例子包括：

1. 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可能性；
2. 神经网络自编码器用于降维和数据压缩，其中参数是每层的权重和偏差，我们通过反复应用链式法则来最小化重建误差；
3. Gauss 混合模型（见第11章）用于建模数据分布，我们优化每个混合组件的位置和形状参数，以最大化模型的可能性。图5.1展示了我们通常使用利用梯度信息（第7.1节）的优化算法来解决这些问题。图5.2概述了本章概念之间以及它们与书中其他章节的联系。

本章的核心概念是函数。一个函数 f 是一个数学对象，它将两个数学对象进行联系。本书中涉及的数学对象即为模型输入 $\mathbf{x} \in \mathbb{R}^D$ 以及拟合目标（函数值） $f(\mathbf{x})$ ，若无额外说明，默认拟合目标都是实数。这里 \mathbb{R}^D 称为 f 的定义域（domain），而相对应的函数值 $f(\mathbf{x})$ 所在的集合被称为 f 的像集（image）或陪域（codomain）。



(a) Regression problem: Find parameters, such that the curve explains the observations well.
(b) Density estimation with a Gaussian mixture model: Find means and covariances, such that the data (dots) can be explained well.

图 5.1 向量微积分在 (a) 回归问题（曲线拟合）和 (b) 分布密度估计（建模数据分布）中有重要应用。

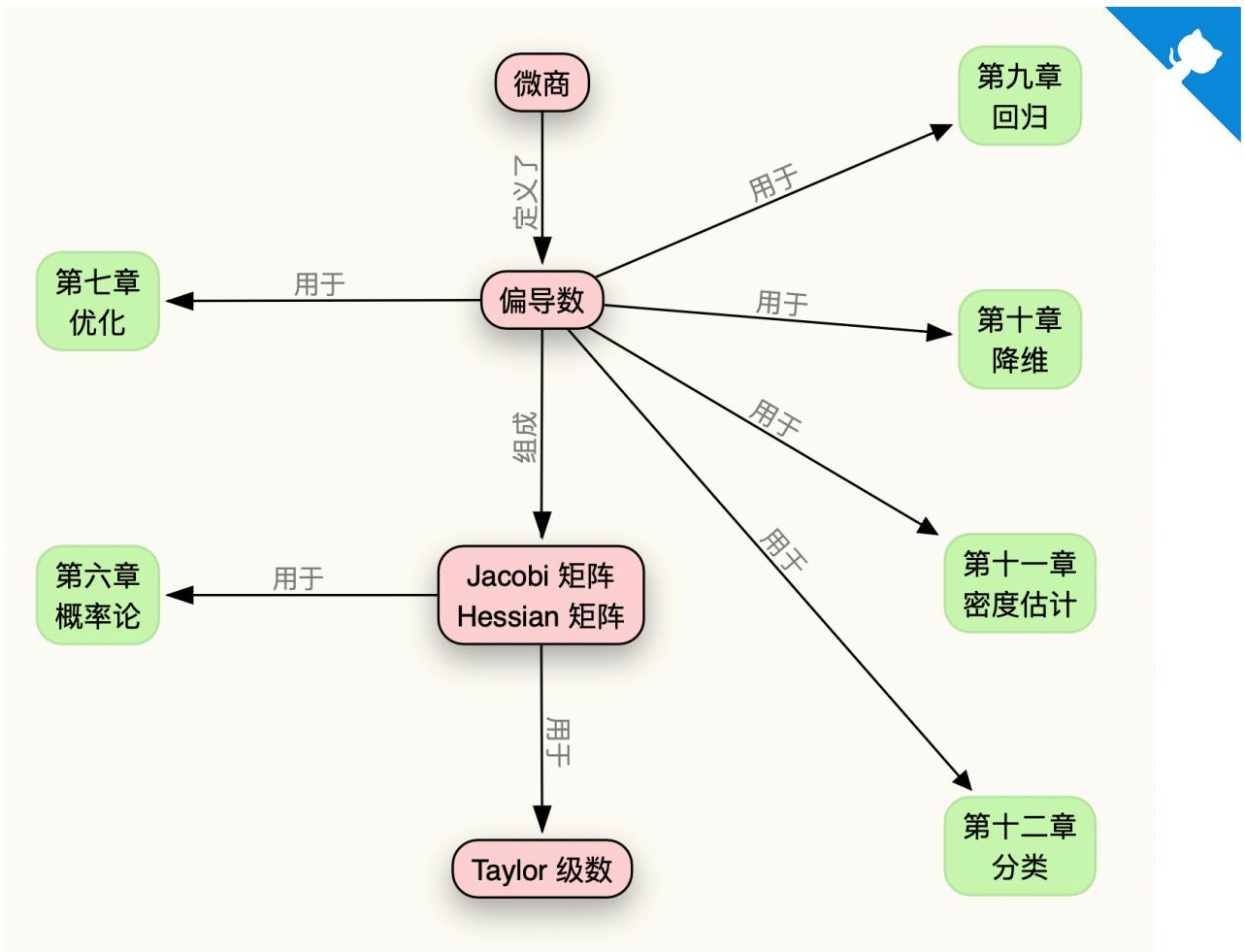


图 5.2 本章的概念地图及与其他章节的联系

2.7.3 节中有对线性函数更为细致的讨论，但一般而言，我们将函数写为下面的形式

$$f : \mathbb{R}^D \rightarrow \mathbb{R} \quad (5.1a)$$

$$\mathbf{x} \mapsto f(\mathbf{x}) \quad (5.1b)$$

其中 (5.1a) 说的是 f 是一个由 \mathbb{R}^D 至 \mathbb{R} 的映射，而 (5.2b) 指的是 f 将每一个输入 \mathbf{x} 对应于唯一的函数值 $f(\mathbf{x})$ 。

示例 5.1 请回忆在 3.2 节中我们谈到点积是一种特殊的内积。沿用之前的记号，函数 $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{x}, \mathbf{x} \in \mathbb{R}^2$ 相当于

$$f : \mathbb{R}^2 \rightarrow \mathbb{R} \quad (5.2a)$$

$$\mathbf{x} \mapsto x_1^2 + x_2^2. \quad (5.2b)$$

本章将介绍如何计算函数的梯度——这在机器学习中如何充分利用学习非常重要，因为梯度指向函数值提升的最陡峭方向。所以向量微积分是机器学习中所需的基础数学工具。我们在全书中都默认函数是可微的，但若具备一些尚未提及的额外定义，很多机器学习方法可被扩展至次梯度（**sub-differentials**，当函数连续但在某些点不可微时）。我们将在第七章探讨带有条件限制的此类函数。

< 上一章节

第四章 矩阵分解

下一章节 >

第六章 概率与统计





5.1 一元函数的微分

接下来我们简要复习一下学过高中数学读者较为熟悉的一元函数的微分。我们从用于定义微分的重要概念——一元函数 $y = f(x), x, y \in \mathbb{R}$ 的差商开始。

图 5.3

定义 5.1 (差商) 一元函数的差商

$$\frac{\delta y}{\delta x} := \frac{f(x + \delta x) - f(x)}{\delta x} \quad (5.3)$$

计算连接函数 f 之图像上的两点的割线之斜率。如图 5.3 所示，两点的横坐标分别为 x_0 和 $x_0 + \delta x$ 。

若 f 是线性函数，差商也可以看做函数 f 上从点 x 至 $x + \delta x$ 之间的平均斜率。若对 δx 去极限 $\delta x \rightarrow 0$ ，我们得到了 f 在 x 处的切线斜率；如果 f 可微，这个切线斜率就是 f 在 x 处的导数。

定义 5.2 (导数) 对正实数 $h > 0$ ，函数 f 在 x 处的导数由下面的极限定义：

$$\frac{df}{dx} := \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h} \quad (5.4)$$

对应到图 5.3 中，割线将变为切线。

f 的导数时刻指向 f 提升最快的方向。



示例 5.2 (多项式函数的导数) 我们想计算多项式函数 $f(x) = x^n, n \in \mathbb{N}$ 的导数。即使我们可能已经知道答案是 nx^{n-1} , 但我们依然希望使用导数和差商的定义导出它。使用导数的定义 (5.4), 我们有

$$\frac{df}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (5.5a)$$

$$= \lim_{h \rightarrow 0} \frac{(x+h)^n - x^n}{h} \quad (5.5b)$$

$$= \lim_{h \rightarrow 0} \frac{\sum_{i=0}^n \binom{n}{i} x^{n-i} h^i - x^n}{h}. \quad (5.5c)$$

注意到 $x^n = \binom{n}{0} x^{n-0} h^0$, 所以上式分子相当于求和项从 1 开始, 于是上式变为

$$\frac{df}{dx} = \lim_{h \rightarrow 0} \frac{\sum_{i=1}^n \binom{n}{i} x^{n-i} h^i}{h} \quad (5.6a)$$

$$= \lim_{h \rightarrow 0} \sum_{i=1}^n \binom{n}{i} x^{n-i} h^{i-1} \quad (5.6b)$$

$$= \lim_{h \rightarrow 0} \left(\binom{n}{1} x^{n-1} + \underbrace{\sum_{i=2}^n \binom{n}{i} x^{n-i} h^{i-1}}_{\rightarrow 0 \text{ as } h \rightarrow 0} \right) \quad (5.6c)$$

$$= \frac{n!}{1!(n-1)!} x^{n-1} = nx^{n-1}. \quad (5.6d)$$

5.1.1 Taylor 级数

所谓 Taylor 级数是将函数 f 表示成的那个无限项求和式, 其中的所有的项都和 f 在点 x_0 处的导数相关。



定义 5.3 (Taylor 多项式) 函数 $f : \mathbb{R} \rightarrow \mathbb{R}$ 在点 x_0 的 n 阶 Taylor 多项式是

$$T_n(x) := \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k, \quad (5.7)$$

其中 $f^{(k)}(x_0)$ 是 f 在 x_0 处的 k 阶导数（假设其存在），而 $\frac{f^{(k)}(x_0)}{k!}$ 是多项式各项的系数。

对于所有的 $t \in \mathbb{R}$ 我们约定 $t^0 := 1$

定义 5.4 (Taylor 级数) 对于光滑函数 $f \in \mathcal{C}^\infty, f : \mathbb{R} \rightarrow \mathbb{R}$, 它在点 x_0 处的 Taylor 级数定义为

$$T_\infty(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k. \quad (5.8)$$

若 $x_0 = 0$, 我们得到了一个 Taylor 级数的特殊情况——Maclaurin 级数。

如果 $f(x) = T_\infty(x)$, 则我们称 f 是解析函数。

注：一般而言，某个不一定为多项式函数的 n 阶 Taylor 多项式是这个函数的近似，它在 x_0 的邻域中与 f 接近。事实上，对于阶数为 $k \leq n$ 的多项式函数 f , n 阶 Taylor 多项式就是这个多项式函数本身，因为对所有的 $i > k$, 多项式函数 f 的 i 阶导数 $f^{(i)}$ 均为零。

示例 5.3 (Taylor 多项式) 考虑多项式

$$f(x) = x^4 \quad (5.9)$$



并求它在 $x_0 = 1$ 处的 Taylor 多项式 T_6 。我们先求函数在该点的各阶导数 $f^{(k)}(1), k = 0, \dots, 6$:

$$f(1) = 1 \quad (5.10)$$

$$f'(1) = 4 \quad (5.11)$$

$$f''(1) = 12 \quad (5.12)$$

$$f^{(3)}(1) = 24 \quad (5.13)$$

$$f^{(4)}(1) = 24 \quad (5.14)$$

$$f^{(5)}(1) = 0 \quad (5.15)$$

$$f^{(6)}(1) = 0 \quad (5.16)$$

于是所求的 Taylor 多项式为

$$T_6(x) = \sum_{k=0}^6 \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k \quad (5.17a)$$

$$= 1 + 4(x - 1) + 6(x - 1)^2 + 4(x - 1)^3 + (x - 1)^4 \quad (5.17b)$$

整理得到

$$T_6(x) = (1 - 4 + 6 - 4 + 1) + x(4 - 12 + 12 - 4) \quad (1)$$

$$+ x^2(6 - 12 + 6) + x^3(4 - 4) + x^4 \quad (5.18a)$$

$$= x^4 = f(x). \quad (5.18b)$$

我们得到了和原函数一模一样的 Taylor 多项式。

图 5.4

示例 5.4 (Taylor 级数) 如图 5.4, 考虑函数

$$f(x) = \sin(x) + \cos(x) \in \mathcal{C}^\infty.$$

我们计算其在 $x_0 = 0$ 处的 Taylor 级数, 也就是 Maclaurin 级数。我们可以求得 f 在 0 处的各阶导数:

$$f(0) = \sin(0) + \cos(0) = 1 \quad (5.20)$$

$$f'(0) = \cos(0) - \sin(0) = 1 \quad (5.21)$$

$$f''(0) = -\sin(0) - \cos(0) = -1 \quad (5.22)$$

$$f^{(3)}(0) = -\cos(0) + \sin(0) = -1 \quad (5.23)$$

$$f^{(4)}(0) = \sin(0) + \cos(0) = f(0) = 1 \quad (5.24)$$

$$\vdots \quad (2)$$

从上面的结果我们可以找到一些规律。首先由于 $\sin(0) = 0$, 有级数的各项系数只能为 ± 1 , 其中每个不同的值在切换为其相反数时都连续出现两次, 进而有 $f^{(k+4)}(0) = f^{(k)}(0)$ 。因此我们可以得到函数 f 在 $x_0 = 0$ 处的 Taylor 级数:

$$T_\infty(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k \quad (5.25a)$$

$$= 1 + x - \frac{1}{2!}x^2 - \frac{1}{3!}x^3 + \frac{1}{4!}x^4 + \frac{1}{5!}x^5 - \dots \quad (5.25b)$$

$$= 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 \mp \dots + x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 \mp \dots \quad (5.25c)$$

$$= \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} x^{2k} + \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1} \quad (5.25d)$$

$$= \cos(x) + \sin(x), \quad (5.25e)$$

其中我们使用了三角函数的幂级数表示:

$$\cos(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} x^{2k}, \quad (5.26)$$

$$\sin(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1}. \quad (5.27)$$

图 5.4 展示了上述条件下的前几个 Taylor 多项式 T_n , 其中 $n = 0, 1, 5, 10$ 。

注: Taylor 级数是一种特殊形式的幂级数:

$$f(x) = \sum_{k=0}^{\infty} a_k (x - c)^k, \quad (5.28)$$

其中 a_k 是系数, c 是常数。不难看出这与定义 5.4 形式的一致性。

5.1.2 微分法则

下面我们简要介绍基本的微分法则, 其中我们使用 f' 表示 f 的导数。

$$\text{乘法法则: } [f(x)g(x)]' = f'(x)g(x) + f(x)g'(x) \quad (5.29)$$

$$\text{除法法则: } \left[\frac{f(x)}{g(x)} \right]' = \frac{f'(x)g(x) - f(x)g'(x)}{[g(x)]^2} \quad (5.30)$$

$$\text{加法法则: } [f(x) + g(x)]' = f'(x) + g'(x) \quad (5.31)$$

$$\text{链式法则: } (g[f(x)])' = (g \circ f)'(x) = g'[f(x)]f'(x) \quad (5.32)$$

其中 $g \circ f$ 表示函数的复合: $x \mapsto f(x) \mapsto g[f(x)]$ 。

示例 5.5 (链式法则) 使用链式法则计算函数 $h(x) = (2x + 1)^4$ 的导数。

不难看出

$$h(x) = (2x + 1)^4 = g[f(x)], \quad (5.33)$$

$$f(x) = 2x + 1, \quad (5.34)$$

$$g(f) = f^4, \quad (5.35)$$

然后计算 f 和 g 的导数:

$$f'(x) = 2, \quad (5.36)$$

$$g'(f) = 4f^3, \quad (5.37)$$

这样我们就得到 h 的导数:

$$h'(x) \stackrel{(5.32)}{=} g'(f)f'(x) = (4f^3) \cdot 2 \stackrel{(5.34)}{=} 4(2x + 1)^3 \cdot 2 = 8(2x + 1)^3,$$

其中我们用到了链式法则, 并在 $g'(f)$ 中的 f 代换为 (5.34) 中的表达式。



下一章节 >

5.2 偏导数和梯度



5.2 偏导数和梯度

在 5.1 节中讨论了标量元 $x \in \mathbb{R}$ 的函数 f 的微分之后，本节将考虑函数 f 的自变量含有多个元的一般情形，即 $\mathbf{x} \in \mathbb{R}^n$ ；例如 $f(x_1, x_2)$ 。相应地，函数的导数就推广到多元情形就变成了**梯度**。

我们可以通过保持其他变量不动，然后改变变元 x 来获取函数的梯度：将对各变元的偏导数组合起来。

定义 5.5（偏导数） 给定 n 元函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathbf{x} \mapsto f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$, 它的各偏导数为

$$\begin{aligned}\frac{\partial f}{\partial x_1} &= \lim_{h \rightarrow 0} \frac{f(x_1 + h, x_2, \dots, x_n) - f(\mathbf{x})}{h} \\ &\vdots \\ \frac{\partial f}{\partial x_n} &= \lim_{h \rightarrow 0} \frac{f(x_1, \dots, x_{n-1}, x_n + h) - f(\mathbf{x})}{h}\end{aligned}\tag{5.39}$$

然后将各偏导数组合为向量，就得到了梯度向量

$$\nabla_x f = \text{grad } f = \frac{df}{d\mathbf{x}} = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right] \in \mathbb{R}^n\tag{5.40}$$

其中 n 是变元数， 1 是 f 像集（陪域）的维数。我们在此定义列向量 $\mathbf{x} = [x_1, \dots, x_n]^\top \in \mathbb{R}^n$ 。行向量 (5.40) 称为 f 的**梯度**或者**Jacobi 矩阵**，是 5.1 节中的导数的推广。

注：此处的 Jacobi 矩阵是其特殊情况。在 5.3 节中我们将讨论向量值函数的 Jacobi 矩阵。



译者注：可以看到，梯度向量是一个线性变换： $D : \mathbb{R}^n \rightarrow \mathbb{R}$ 。这样的行向量又被称为 **余向量** (**covector**)，其中的余 (co-) 表示行和列的对偶关系。

示例 5.6（使用链式法则计算偏导数） 给定函数 $f(x, y) = (x + 2y^3)^2$ ，我们可以这样计算它的偏导数：

$$\frac{\partial f(x, y)}{\partial x} = 2(x + 2y^3) \cdot \frac{\partial}{\partial x}(x + 2y^3) = 2(x + 2y^3), \quad (5.41)$$

$$\frac{\partial f(x, y)}{\partial y} = 2(x + 2y^3) \cdot \frac{\partial}{\partial y}(x + 2y^3) = 12(x + 2y^3)y^2. \quad (5.42)$$

上述过程中我们使用了链式法则 (5.32)。

注（作为行向量的梯度）：文献中并不常像一般的向量表示那样将梯度写为列向量。这样做的原因有两个：首先，这样的定义方便拓展为向量值函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 的情形，这样梯度就变为矩阵；其次，我们可以方便地对其使用多变元的链式法则而不用注意梯度的维数。我们将在 5.3 节中进一步讨论以上两点。

示例 5.7（梯度） 给定函数 $f(x, y) = x_1^2x_2 + x_1x_2^3 \in \mathbb{R}$ ，它的各偏导数（相对于 x_1 和 x_2 求偏导）为

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2x_1x_2 + x_2^3 \quad (5.43)$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = x_1^2 + 3x_1x_2^2 \quad (5.44)$$

于是我们可以得到梯度

$$\frac{df}{dx} = \left[\frac{\partial f(x_1, x_2)}{\partial x_1}, \frac{\partial f(x_1, x_2)}{\partial x_2} \right] = [2x_1 x_2 + x_2^3, x_1^2 + 3x_1 x_2^2] \in \mathbb{R}^{1 \times 2}.$$

5.2.1 偏导数的基本法则

当 $\mathbf{x} \in \mathbb{R}^n$ 时，即在多元函数的情况下微分法则（如加法、乘法、链式法则）和我们在学校中学到的无异。但在对向量 $\mathbf{x} \in \mathbb{R}^n$ 求导时，我们需要额外注意，因为我们现在得到的梯度包括向量和矩阵，而矩阵乘法是非交换的。下面是一般的加法、乘法、和链式法则：

$$\text{Product rule: } \frac{\partial}{\partial \mathbf{x}} [f(\mathbf{x})g(\mathbf{x})] = \frac{\partial f}{\partial \mathbf{x}} g(\mathbf{x}) + f(\mathbf{x}) \frac{\partial g}{\partial \mathbf{x}} \quad (5.46)$$

$$\text{Sum rule: } \frac{\partial}{\partial \mathbf{x}} [f(\mathbf{x}) + g(\mathbf{x})] = \frac{\partial f}{\partial \mathbf{x}} + \frac{\partial g}{\partial \mathbf{x}} \quad (5.47)$$

$$\text{Chain rule: } \frac{\partial}{\partial \mathbf{x}} (g \circ f)(\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}} g[f(\mathbf{x})] = \frac{\partial g}{\partial f} \frac{\partial f}{\partial \mathbf{x}} \quad (5.48)$$

我们仔细观察链式法则 (5.48)，可以通过它看到相应矩阵乘法的规律，即相邻相乘矩阵的相邻维度需要相等（见 2.2.1 节）。从左往右看，可以发现 ∂f 先出现在第一项的“分母”，然后出现在第二项“分子”，按照通常乘法的定义可以理解， ∂f 对应的维数对应则可以消去，剩下的就是 $\frac{\partial g}{\partial \mathbf{x}}$ 。

注意， $\frac{\partial f}{\partial \mathbf{x}}$ 并不是严格意义上的分数，上述说法只是为了增进理解

5.2.2 链式法则 (chain rule)

考虑变元为 x_1, x_2 函数 $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ ，而 $x_1(t)$ 和 $x_2(t)$ 又是变元 t 的函数。为了计算 f 对 t 的梯度，需要用到链式法则 (5.48)：

$$\frac{df}{dt} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} \begin{bmatrix} \frac{\partial x_1(t)}{\partial t} \\ \frac{\partial x_2(t)}{\partial t} \end{bmatrix} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial t}, \quad (5.49)$$

其中 d 表示梯度，而 ∂ 表示偏导数。



示例 5.8 考虑函数 $f(x_1, x_2) = x_1^2 + 2x_2$, 其中 $x_1 = \sin t$, $x_2 = \cos t$, 则

$$\frac{df}{dt} = \frac{\partial f}{\partial x_1} \frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2} \frac{\partial x_2}{\partial t} \quad (5.50a)$$

$$= 2 \sin t \frac{\partial \sin t}{\partial t} + 2 \frac{\partial \cos t}{\partial t} \quad (5.50b)$$

$$= 2 \sin t \cos t - 2 \sin t = 2 \sin t(\cos t - 1) \quad (5.50c)$$

就是 f 关于 t 的梯度。

如果 $f(x_1, x_2)$ 是 x_1 和 x_2 的函数, 而 $x_1(s, t)$ 和 $x_2(s, t)$ 又分别为 s 和 t 的函数, 那么根据链式法则会得到下面的结果:

$$\frac{\partial f}{\partial s} = \frac{\partial f}{\partial \mathbf{x}_1} \frac{\partial \mathbf{x}_1}{\partial s} + \frac{\partial f}{\partial \mathbf{x}_2} \frac{\partial \mathbf{x}_2}{\partial s} \quad (5.51)$$

$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial \mathbf{x}_1} \frac{\partial \mathbf{x}_1}{\partial t} + \frac{\partial f}{\partial \mathbf{x}_2} \frac{\partial \mathbf{x}_2}{\partial t} \quad (5.52)$$

而函数的梯度为

$$\begin{aligned} \frac{df}{d(s, t)} &= \frac{\partial f}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial (s, t)} = \underbrace{\left[\begin{array}{cc} \frac{\partial f}{\partial \mathbf{x}_1} & \frac{\partial f}{\partial \mathbf{x}_2} \end{array} \right]}_{=\frac{\partial f}{\partial \mathbf{x}}} \underbrace{\left[\begin{array}{cc} \frac{\partial \mathbf{x}_1}{\partial s} & \frac{\partial \mathbf{x}_1}{\partial t} \\ \frac{\partial \mathbf{x}_2}{\partial s} & \frac{\partial \mathbf{x}_2}{\partial t} \end{array} \right]}_{=\frac{\partial \mathbf{x}}{\partial (s, t)}}. \end{aligned} \quad (5.53)$$

以上的写法 (5.53) 当且仅当梯度被写为行向量时才是正确的, 否则我们需要对结果进行转置, 以保证矩阵的维度对应。在梯度为向量或矩阵时这样看来似乎比较显然, 但当之后讨论中涉及的梯度变成 **张量 (tensor)** 时对其进行转置就不那么容易了。

验证梯度是否正确 将差商取极限而得到梯度的方法在计算机程序中的数值算法处被加以利用。当我们计算函数梯度时, 我们可以通过数值的微小改变计算差商, 然后校验梯度的正确性: 取一个较小的值 (例如 $h = 10^{-4}$) 然后计算有限差商和梯度的解析计算结果, 如果误差足够小则说明梯度的解析结



果大概率是正确的。误差足够小是指 $\sqrt{\frac{\sum_i (dh_i - df_i)^2}{\sum_i (dh_i + df_i)^2}} < 10^{-6}$, 其中 dh_i 是指 f 关于 x_i 得到的有限差商的估计结果, df_i 是指 f 关于 x_i 的解析梯度的计算结果。

< 上一章节

下一章节 >

5.1 一元函数的微分

5.3 向量值函数的梯度



5.3 向量值函数的梯度

一直以来我们讨论的都是实值函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 的偏导数和梯度，接下来我们将将此概念扩展至向量值函数（向量场） $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 的情形，其中 $n \geq 1, m \geq 1$ 。

给定向量值函数 $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 和向量 $\mathbf{x} = [x_1, \dots, x_n]^\top \in \mathbb{R}^n$ ，则该函数的函数值可以写为

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{bmatrix} \in \mathbb{R}^m. \quad (5.54)$$

这样写可以让我们将向量值函数 $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 看成一个全部由实值函数 $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ 构成的向量 $[f_1, \dots, f_m]^\top$ ，而对于每一个 f_i 我们可以不加修改的直接应用 5.2 节中的所有微分法则。这样一来，向量值函数对变元 $x_i \in \mathbb{R}, i = 1, \dots, n$ 的偏导数由下式给出

$$\frac{\partial \mathbf{f}}{\partial x_i} = \begin{bmatrix} \frac{\partial f_1}{\partial x_i} \\ \vdots \\ \frac{\partial f_m}{\partial x_i} \end{bmatrix} = \begin{bmatrix} \lim_{h \rightarrow 0} \frac{f_1(x_1, \dots, x_{i-1}, x_i + h, x_{i+1}, x_n) - f_1(\mathbf{x})}{h} \\ \vdots \\ \lim_{h \rightarrow 0} \frac{f_m(x_1, \dots, x_{i-1}, x_i + h, x_{i+1}, x_n) - f_m(\mathbf{x})}{h} \end{bmatrix} \quad (5.55)$$

从 (5.40) 中我们了解到函数 \mathbf{f} 对向量求导得到的是由一系列偏导数组合得到的行向量。在 (5.55) 中，每个偏导数 $\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_i}$ 自己就是一个列向量，于是我们可以将它们组合起来得到函数 $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 对向量 $\mathbf{x} \in \mathbb{R}^n$ 的梯度：

$$\begin{aligned} \frac{d\mathbf{f}}{d\mathbf{x}} &= \begin{bmatrix} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_n} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m \times n} \end{aligned} \quad (5.56)$$



定义 5.6 (Jacobi 矩阵)

向量值函数 $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 的各一阶偏微分的合集称为 Jacobi 矩阵，它的形状是 $m \times n$ ，定义如下：

$$\mathbf{J} = \nabla_{\mathbf{x}} \mathbf{f} = \frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \dots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} \quad (5.57)$$

$$= \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{bmatrix}, \quad (5.58)$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad J(i, j) = \frac{\partial f_i}{\partial x_j}. \quad (5.59)$$

作为 (5.58) 的一个特例，标量值的向量变元函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}^1$ (如 $f(\mathbf{x}) = \sum_{i=1}^n x_i$) 的 Jacobi 矩阵是一个行向量 (形状为 $1 \times n$)；见 (5.40)。

注：本书中的微分使用 **分子布局 (numerator layout)**。这是说函数 $\mathbf{f} \in \mathbb{R}^m$ 对 $\mathbf{x} \in \mathbb{R}^n$ 的微分 $\frac{d\mathbf{f}}{d\mathbf{x}}$ 得到矩阵的形状为 $m \times n$ ——如 (4.58)——其中 \mathbf{f} 决定这矩阵的行， \mathbf{x} 决定矩阵的列。当然也有所谓的 **分母布局 (denominator layout)**，得到的结果是分子布局的转置。

Jacobi 矩阵将在 6.7 节中概率分布的变量变换方法中起作用，而其中的缩放大小取决于其行列式 (**determinant**)。

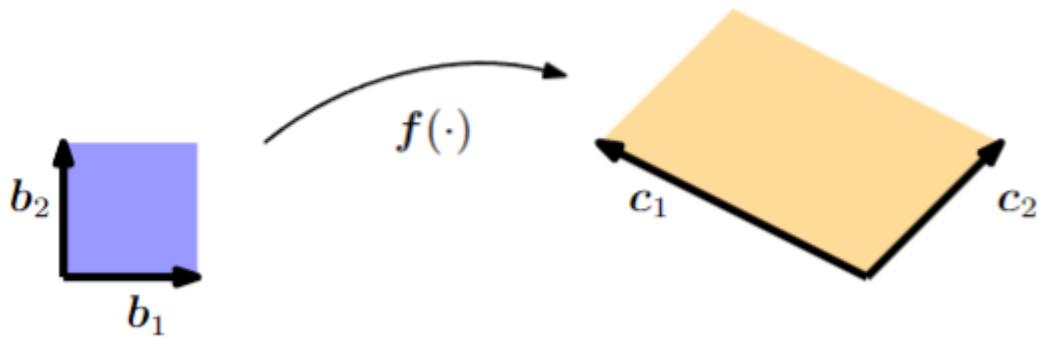


图 5.5

在 4.1 节中，我们已使用行列式计算平行四边形的面积：如果给定正方形的两边所对应的两个向量 $\mathbf{b}_1 = [1, 0]^\top$ 和 $\mathbf{b}_2 = [0, 1]^\top$ ，那么它们构成的正方形的面积是

$$\left| \det \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right| = 1. \quad (5.60)$$

如果我们取平行四边形的两边 $\mathbf{c}_1 = [-2, 1]^\top$ 和 $\mathbf{c}_2 = [1, 1]^\top$ （如图 5.5 所示），其面积等于下面行列式的绝对值：

$$\left| \det \begin{pmatrix} -2 & 1 \\ 1 & 1 \end{pmatrix} \right| = |-3| = 3, \quad (5.61)$$

其刚好是单位正方形面积的三倍。我们可以通过测量单位正方形映射后得到的图形面积得到对应的缩放比例。如果使用线性代数的语言，我们做了一个从 $(\mathbf{b}_1, \mathbf{b}_2)$ 到 $(\mathbf{c}_1, \mathbf{c}_2)$ 的变量变换。在本例中，这个变换是线性的，变换本身的行列式就给出了缩放比例。

现在我们介绍两种确认这样的映射的方法。首先我们假设这个变换是线性的，这样就可以使用第二张中的内容确定它。随后我们将使用本章介绍的偏导数计算这个映射。

方法 1

为了使用线性代数的工具，我们假定 $\{\mathbf{b}_1, \mathbf{b}_2\}$ 和 $\{\mathbf{c}_1, \mathbf{c}_2\}$ 是 \mathbb{R}^2 的一个基（见 2.6.1 节），可见事实上我们做了一个从 $(\mathbf{b}_1, \mathbf{b}_2)$ 到 $(\mathbf{c}_1, \mathbf{c}_2)$ 的基变换，我们要找的变换矩阵就是执行这一基变换的矩阵。使用 2.7.2 节的结论，可以得到变换矩阵

$$\mathbf{J} = \begin{bmatrix} -2 & 1 \\ 1 & 1 \end{bmatrix}, \quad (5.62)$$

满足 $\mathbf{J}\mathbf{b}_1 = \mathbf{c}_1$, $\mathbf{J}\mathbf{b}_2 = \mathbf{c}_2$ 。该矩阵行列式的绝对值为 $|\det(\mathbf{J})| = 3$ ，这就是所求的缩放参数，也即基 $(\mathbf{c}_1, \mathbf{c}_2)$ 张成的平行四边形的面积是基 $(\mathbf{b}_1, \mathbf{b}_2)$ 张成的平行四边形的面积的三倍。

方法 2

线性代数的方法可用于解线性函数的 Jacobi 矩阵，而对于非线性函数（在 6.7 节中涉及），我们使用一种更具一般性的方法——使用偏导数。考虑一个变量转换函数 $\mathbf{f} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ ，它将基 $(\mathbf{b}_1, \mathbf{b}_2)$ 下表示的向量 $\mathbf{x} \in \mathbb{R}^2$ 转换为基 $(\mathbf{c}_1, \mathbf{c}_2)$ 表示下的向量 $\mathbf{y} \in \mathbb{R}^2$ ，我们通过计算映射 \mathbf{f} 作用前后单位面积/体积的变化来确定这个映射。从这个角度出发，我们可以研究当我们稍稍改变 \mathbf{x} 一点后 $\mathbf{f}(\mathbf{x})$ 的变化，而这恰好就是 $\frac{d\mathbf{f}}{d\mathbf{x}} \in \mathbb{R}^{2 \times 2}$ 。我们可以写出 \mathbf{x} 和 \mathbf{y} 的联系如下：

$$y_1 = -2x_1 + x_2 \quad (5.63)$$

$$y_2 = x_1 + x_2 \quad (5.64)$$

就可以容易的写出各项偏导数：

$$\frac{\partial y_1}{\partial x_1} = -2, \quad \frac{\partial y_1}{\partial x_2} = 1, \quad \frac{\partial y_2}{\partial x_1} = 1, \quad \frac{\partial y_2}{\partial x_2} = 1 \quad (5.65)$$

并得到表示坐标变换的 Jacobi 矩阵

$$\mathbf{J} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1 & 1 \end{bmatrix}. \quad (5.66)$$

如果我们处理的是线性函数，则刚刚的 Jacobi 矩阵即为所求（注意 (5.66) 和 (5.62) 完全一样）；如若不然，Jacobi 矩阵是非线性映射的局部线性近似。Jacobi 矩阵的行列式 $|\mathbf{J}|$ 称为 Jacobian 行列式，它的值就是面积或体积变换前后的缩放比例，在这个例子中有 $|\mathbf{J}| = 3$ 。

上面提到的 Jacobian 行列式和变量替换在 6.7 节中对随机变量和分布进行变换时会涉及，它们在机器学习和深度学习中的 **重参数技巧 (Reparametrization Trick)** 中十分重要，也被称为 **无穷摄动分析 (Infinite Perturbation Analysis)**。

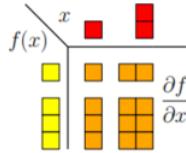


图5.6 (偏) 导数的维度和形状

在本章讨论了函数的导数，图5.6给出它们的形状。如果 $f : \mathbb{R} \rightarrow \mathbb{R}$ ，其梯度只是一个标量（左上角）。如果 $f : \mathbb{R}^D \rightarrow \mathbb{R}$ ，它的梯度是一个形状为 $1 \times D$ 的行向量（右上角）。如果 $f : \mathbb{R} \rightarrow \mathbb{R}^E$ ，它的梯度是一个形状为 $E \times 1$ 的列向量，而如果 $f : \mathbb{R}^D \rightarrow \mathbb{R}^E$ ，梯度则是一个形状为 $E \times D$ 的矩阵。

示例 5.9 (向量值函数的梯度)

给定 $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$, $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^M$, $\mathbf{A} \in \mathbb{R}^{M \times N}$, $\mathbf{x} \in \mathbb{R}^N$ 。为了计算梯度 $\frac{d\mathbf{f}}{d\mathbf{x}}$ ，我们首先确定 $\frac{d\mathbf{f}}{d\mathbf{x}}$ 的维数：由于 $\mathbf{f} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ ，所以有 $\frac{d\mathbf{f}}{d\mathbf{x}} \in \mathbb{R}^{M \times N}$ 。为了计算梯度，我们接下来计算 \mathbf{f} 相对于每个变元 x_j 的偏导数

$$f_i(\mathbf{x}) = \sum_{j=1}^N A_{i,j} x_j \implies \frac{\partial f_i}{\partial x_j} = A_{i,j} \quad (5.67)$$

知道了这些偏导数，我们就得到了梯度

$$\frac{d\mathbf{f}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_M}{\partial x_1} & \cdots & \frac{\partial f_M}{\partial x_N} \end{bmatrix} = \begin{bmatrix} A_{1,1} & \cdots & A_{1,N} \\ \vdots & \ddots & \vdots \\ A_{M,1} & \cdots & A_{M,N} \end{bmatrix} = \mathbf{A} \in \mathbb{R}^{M \times N}.$$

示例 5.10 (链式法则)

考虑函数 $h : \mathbb{R} \rightarrow \mathbb{R}$, $h(t) = (f \circ g)(t)$, 其中

$$f : \mathbb{R}^2 \rightarrow \mathbb{R} \quad (5.69)$$

$$g : \mathbb{R} \rightarrow \mathbb{R}^2 \quad (5.70)$$

$$f(\mathbf{x}) = \exp(x_1, x_2^2), \quad (5.71)$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = g(t) = \begin{bmatrix} t \cos t \\ t \sin t \end{bmatrix} \quad (5.72)$$

计算 h 关于 t 的梯度。因为 $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ 和 $g : \mathbb{R} \rightarrow \mathbb{R}^2$, 于是我们有

$$\frac{\partial f}{\partial \mathbf{x}} \in \mathbb{R}^{1 \times 2}, \quad \frac{\partial g}{\partial t} \in \mathbb{R}^{2 \times 1}. \quad (5.73)$$

复合函数的梯度可通过链式法则求得:

$$\frac{dh}{dt} = \frac{\partial f}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial t} = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \right] \begin{bmatrix} \frac{\partial x_1}{\partial t} \\ \frac{\partial x_2}{\partial t} \end{bmatrix} \quad (5.74a)$$

$$= [\exp(x_1 x_2^2) x_2^2 \quad 2 \exp(x_1 x_2^2) x_1 x_2] \begin{bmatrix} \cos t - t \sin t \\ \sin t + t \cos t \end{bmatrix} \quad (5.74b)$$

$$= \exp(x_1 x_2^2) [x_2^2 (\cos t - t \sin t) + 2x_1 x_2 (\sin t + t \cos t)] \quad (5.74c)$$

其中, $x_1 = t \cos t$, $x_2 = t \sin t$; 见 (5.72)。

示例 5.11 (线性模型中最小二乘损失之梯度)

考虑下面的线性模型

$$\mathbf{y} = \Phi \boldsymbol{\theta}, \quad (5.75)$$

其中 $\boldsymbol{\theta} \in \mathbb{R}^D$ 是参数向量, $\Phi \in \mathbb{R}^{N \times D}$ 是输入特征, $\mathbf{y} \in \mathbb{R}^N$ 是对应的观测值。为方便叙述, 我们定义

$$L(\mathbf{e}) := \|\mathbf{e}\|^2, \quad (5.76)$$

$$\mathbf{e}(\boldsymbol{\theta}) := \mathbf{y} - \Phi \boldsymbol{\theta}. \quad (5.77)$$

我们使用链式法则计算偏导数 $\frac{\partial L}{\partial \boldsymbol{\theta}}$, 其中 L 称为最小二乘损失函数 (Least-squares loss function)。开始计算之前, 我们首先确定梯度的维数:

$$\frac{\partial L}{\partial \theta} \in \mathbb{R}^{L \times D}. \quad (5.78)$$

然后使用链式法则：

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial e} \frac{\partial e}{\partial \theta}, \quad (5.79)$$

其中等式左边向量从左往右的第 d 个元素是由

$$\frac{\partial L}{\partial \theta}[1, d] = \sum_{n=1}^N \frac{\partial L}{\partial e}[n] \frac{\partial e}{\partial \theta}[n, d] \quad (5.80)$$

给出的。我们知道 $\|e\|^2 = e^\top e$ (见 3.2 节) 因此有

$$\frac{\partial L}{\partial e} = 2e^\top \in \mathbb{R}^{1 \times N}. \quad (5.81)$$

进一步，我们有

$$\frac{\partial e}{\partial \theta} = -\Phi \in \mathbb{R}^{N \times D}, \quad (5.82)$$

结合起来即为所求：

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial e} \frac{\partial e}{\partial \theta} = 2e^\top (-\Phi) = -2e^\top \Phi = -2(\Phi^\top e)$$

$$= -2(\Phi^\top (\Phi \theta)) = -2(\Phi^\top \Phi) \theta = -2R \theta$$

$$\boxed{-2R \theta} \quad (5.83)$$

注：如果不使用链式法则，我们就可以通过展开下面的函数进行求导以得到相同的结果

$$L_2(\theta) := \|y - \Phi\theta\|^2 = (y - \Phi\theta)^\top (y - \Phi\theta). \quad (5.84)$$

然而这种方法虽然可用于这样的简单函数，但当我们面对深度复合的复杂函数时这样做就不现实了。



< 上一章节

下一章节 >

5.2 偏导数和梯度

5.4 矩阵的梯度



5.4 矩阵的梯度

接下来我们将会看见需要求矩阵对向量（或其他矩阵）的梯度的情形。它们的结果是一个多维度的张量 (*tensor*)，我们可以将其看做装有偏导数的多维数组。例如，如果我们计算一个 $m \times n$ 形状的矩阵 \mathbf{A} 对 $p \times q$ 形状的矩阵 \mathbf{B} 的梯度，结果 Jacobian 矩阵的形状将是 $(m \times n) \times (p \times q)$ ，即一个四维张量 \mathbf{J} ，它的每个分量可以写为 $J_{i,j,k,l} = \frac{\partial A_{i,j}}{\partial B_{k,l}}$ 。

由于矩阵代表着线性变换。我们可以用这样的事实构造形状为 $m \times n$ 矩阵空间 $\mathbb{R}^{m \times n}$ 到 mn 长度的线性空间 \mathbb{R}^{mn} 的线性空间同构（可逆的线性映射）。这样一来我们就可以调整矩阵 \mathbf{A} 和 \mathbf{B} 的形状，使其分别变成长度为 mn 和长度为 pq 的向量。因此对这样的向量求梯度就得到形状为 $mn \times pq$ 的 Jacobi 矩阵。图 5.7 画出了上面两种方法的示意图。实际操作中，将矩阵压扁成向量然后继续处理 Jacobi 矩阵的方法较受欢迎，因为这样一来链式法则 (5.48) 就变成简单的矩阵乘法；而如果处理的是 Jacobi 张量，我们就得对于二者相乘时求和的维度倍加小心。

示例 5.12 (向量对矩阵的梯度) 考虑下面的例子：

$$\mathbf{f} = \mathbf{Ax}, \quad \mathbf{f} \in \mathbb{R}^M, \quad \mathbf{A} \in \mathbb{R}^{M \times N}, \quad \mathbf{x} \in \mathbb{R}^N, \quad (5.85)$$

求梯度 $\frac{d\mathbf{f}}{d\mathbf{A}}$ 。首先确定梯度的维数：

$$\frac{d\mathbf{f}}{d\mathbf{A}} \in \mathbb{R}^{M \times (M \times N)}. \quad (5.86)$$

按照定义，梯度里面装着一族偏导的结果：

$$\frac{d\mathbf{f}}{d\mathbf{A}} = \begin{bmatrix} \frac{\partial f_1}{\partial \mathbf{A}} \\ \vdots \\ \frac{\partial f_M}{\partial \mathbf{A}} \end{bmatrix}, \quad \frac{\partial f_i}{\partial \mathbf{A}} \in \mathbb{R}^{1 \times (M \times N)}. \quad (5.87)$$

接下来我们求每一项的值。我们首先根据矩阵乘法分别展开每个结果分量 f_i

:

$$f_i = \sum_{j=1}^N A_{i,j} x_j, \quad i = 1, \dots, M, \quad (5.88)$$

然后得到 f_i 对矩阵中每一份量的偏导数

$$\frac{\partial f_i}{\partial A_{j,q}} = x_q. \quad (5.89)$$

将它们一行一行的组合起来，并注意一下结果的形状，我们就得到了 f_i 对矩阵 \mathbf{A} 中各行的偏导：

$$\frac{\partial f_i}{\partial A_{i,:}} = \mathbf{x}^\top \in \mathbb{R}^{1 \times 1 \times N}, \quad (5.90)$$

$$\frac{\partial f_i}{\partial A_{k \neq i,:}} = \mathbf{0}^\top \in \mathbb{R}^{1 \times 1 \times N}, \quad (5.91)$$

由于 f_i 是实值函数，矩阵 \mathbf{A} 的每一行形状为 $1 \times N$ ，我们得到的 f_i 关于矩阵每一行的偏导数张量的形状就是 $1 \times 1 \times N$ 。最后我们将 (5.91) 堆叠起来，就得到所求的梯度 (5.87) 中的每一项：

$$\frac{\partial f_i}{\partial \mathbf{A}} = \begin{bmatrix} \mathbf{0}^\top \\ \vdots \\ \mathbf{0}^\top \\ \mathbf{x}^\top \\ \mathbf{0}^\top \\ \vdots \\ \mathbf{0}^\top \end{bmatrix} \in \mathbb{R}^{1 \times (M \times N)}. \quad (5.92)$$

示例 5.13 (矩阵对矩阵的梯度) 给定矩阵 $\mathbf{R} \in \mathbb{R}^{M \times N}$ ，和矩阵值函数 $\mathbf{f} : \mathbb{R}^{M \times N} \rightarrow \mathbb{R}^{N \times N}$ ：

$$\mathbf{f}(\mathbf{R}) = \mathbf{R}^\top \mathbf{R} =: \mathbf{K} \in \mathbb{R}^{N \times N}, \quad (5.93)$$

求梯度 $\frac{d\mathbf{K}}{d\mathbf{R}}$ 。这个问题有些困难。我们先写下已知信息：梯度的维数

$$\frac{d\mathbf{K}}{d\mathbf{R}} \in \mathbb{R}^{(N \times N) \times (M \times N)}, \quad (5.94)$$

毫无疑问这是个张量。我们进一步写出 \mathbf{K} 中每个元素对矩阵 \mathbf{R} 的梯度维数：

$$\frac{dK_{p,q}}{d\mathbf{R}} \in \mathbb{R}^{1 \times M \times N}, p, q = 1, \dots, N \quad (5.95)$$

其中 $K_{p,q}$ 是 $\mathbf{K} = \mathbf{f}(\mathbf{R})$ 中处于第 p 行，第 q 列的元素。用 \mathbf{r}_i 表示 \mathbf{R} 的第 i 列，则 \mathbf{K} 中的每个元素可以写成 \mathbf{R} 中两列的点积，即

$$K_{p,q} = \mathbf{r}_p^\top \mathbf{r}_q = \sum_{m=1}^M R_{m,p} R_{m,q}. \quad (5.96)$$

接着我们计算偏导数

$$\frac{\partial K_{p,q}}{\partial R_{i,j}} = \sum_{m=1}^M \frac{\partial}{\partial R_{i,j}} R_{m,p} R_{m,q} = \partial_{p,q,i,j}, \quad (5.97)$$

其中

$$\partial_{p,q,i,j} = \begin{cases} R_{i,q}, & j = p, p \neq q \\ R_{i,p}, & j = q, p \neq q \\ 2R_{i,q}, & j = p, p = q \\ 0, & \text{其他情形} \end{cases}. \quad (5.98)$$

从 (9.94) 我们知道目标梯度的形状是 $(N \times N) \times (M \times N)$ ，它的每个分量的值由 (5.98) 给出，其中 $p, q, j = 1, \dots, N, i = 1, \dots, M$ 。

< 上一章节

下一章节 >

5.3 向量值函数的梯度

5.5 常用梯度恒等式



5.5 常用梯度恒等式

下面，我们列出了一些在机器学习环境中常用的梯度恒等式（Petersen and Pedersen, 2012）。其中 $\text{tr}(\cdot)$ 代表矩阵的迹（见定义4.4）， $\det(\cdot)$ 表示矩阵的行列式（见 4.1 节）， $\mathbf{f}(\mathbf{X})^{-1}$ 表示 $\mathbf{f}(\mathbf{X})$ 的逆（假设其存在）。

$$\frac{\partial}{\partial \mathbf{X}} \mathbf{f}(\mathbf{X})^\top = \left(\frac{\partial \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}} \right)^\top, \quad (5.99)$$

$$\frac{\partial}{\partial \mathbf{X}} \text{tr}[\mathbf{f}(\mathbf{X})] = \text{tr} \left[\frac{\partial \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}} \right], \quad (5.100)$$

$$\frac{\partial}{\partial \mathbf{X}} \det [\mathbf{f}(\mathbf{X})] = \det [\mathbf{f}(\mathbf{X})] \text{tr} \left[\mathbf{f}(\mathbf{X})^{-1} \frac{\partial \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}} \right] \quad (5.101)$$

$$\frac{\partial}{\partial \mathbf{X}} \mathbf{f}(\mathbf{X})^{-1} = -\mathbf{f}(\mathbf{X})^{-1} \left[\frac{\partial \mathbf{f}(\mathbf{X})}{\partial \mathbf{X}} \right] \mathbf{f}(\mathbf{X})^{-1} \quad (5.102)$$

$$\frac{\partial \mathbf{a}^\top \mathbf{X}^{-1} \mathbf{b}}{\partial \mathbf{X}} = -(\mathbf{X}^{-1})^\top \mathbf{a} \mathbf{b}^\top (\mathbf{X}^{-1})^\top \quad (5.103)$$

$$\frac{\partial \mathbf{x}^\top \mathbf{a}}{\partial \mathbf{x}} = \mathbf{a}^\top \quad (5.104)$$

$$\frac{\partial \mathbf{a}^\top \mathbf{x}}{\partial \mathbf{x}} = \mathbf{a}^\top \quad (5.105)$$

$$\frac{\partial \mathbf{a}^\top \mathbf{X} \mathbf{b}}{\partial \mathbf{X}} = \mathbf{a} \mathbf{b}^\top \quad (5.106)$$

$$\frac{\partial \mathbf{x}^\top \mathbf{B} \mathbf{x}}{\partial \mathbf{x}} = \mathbf{x}^\top (\mathbf{B} + \mathbf{B}^\top) \quad (5.107)$$

$$\frac{\partial}{\partial \mathbf{s}} (\mathbf{x} - \mathbf{A}\mathbf{s})^\top \mathbf{W} (\mathbf{x} - \mathbf{A}\mathbf{s}) = -2(\mathbf{x} - \mathbf{A}\mathbf{s})^\top \mathbf{W} \mathbf{A}, \quad (5.108)$$

for symmetric \mathbf{W} (1)

注：本书中我们仅讨论矩阵的迹和转置。然而，我们已看到求导的结果可能是高维张量，通常的迹和转置在其范畴中没有定义。在这些情况下，形状为 $D \times D \times E \times F$ 的张量的迹将是一个 $E \times F$ 形状的矩阵。这是 **张量缩并** (**tensor contraction**) 的一种特殊情况。类似地，当我们“转置”一个张量时，我们说的是交换前两个维度。具体而言，在 (5.99) 到 (5.102) 中，当我们计算多元函数 $\mathbf{f}(\cdot)$ 对矩阵的导数时，我们需要张量相关的计算，而像在 5.4 节中那样将其拉直成向量。



< 上一章节

下一章节 >

5.4 矩阵的梯度

5.6 反向传播与自动微分



5.6 反向传播与自动微分

在许多机器学习的应用中，我们通过计算学习目标关于模型参数的梯度，然后执行梯度下降（见 7.1 节）找好的模型参数。对于给定的目标函数，我们可以利用微积分的链式法则得到其对模型参数的梯度（见 5.2.2 节）。我们在 5.3 节已经尝试对平方损失结果关于线性回归模型参数求梯度。

考虑下面的函数：

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos[x^2 + \exp(x^2)]. \quad (5.109)$$

由链式法则，并注意到微分的线性性，我们可以得到：

$$\frac{df}{dx} = \frac{2x + 2x \exp\{x^2\}}{2\sqrt{x^2 + \exp\{x^2\}}} - \sin(x^2 + \exp\{x^2\})(2x + 2x \exp\{x^2\}) \quad (1)$$

$$= 2x \left[\frac{1}{2\sqrt{x^2 + \exp\{x^2\}}} - \sin(x^2 + \exp\{x^2\}) \right] (1 + \exp\{x^2\}) \quad (2)$$

像这样显式求解得到这样冗长的导数表达往往不切实际。在实践中这意味着^{15.10}处理，梯度的实现可能比计算函数值要昂贵得多，这增加了不必要的开销。对于神经网络模型，反向传播算法(Kelley, 1960; Bryson, 1961; Dreyfus, 1962; Rumelhart et al., 1986)是一种计算误差对模型参数梯度的有效方法。

5.6.1 深度神经网络中的梯度

深度学习领域将链式法则的功用发挥到了极致，输入 \mathbf{x} 通过多层复合的函数得到函数值 \mathbf{y} ：

$$\mathbf{y} = (f_K \circ f_{K-1} \circ \cdots \circ f_1)(\mathbf{x}) = f_K \left\{ f_{K-1} [\cdots (f_1(\mathbf{x}) \cdots)] \right\} \quad (5.111)$$

其中， \mathbf{x} 是输入（如图像）， \mathbf{y} 是观测值（如类标签），每个函数 $f_i, i = 1, \dots, K$ ，有各自的参数。

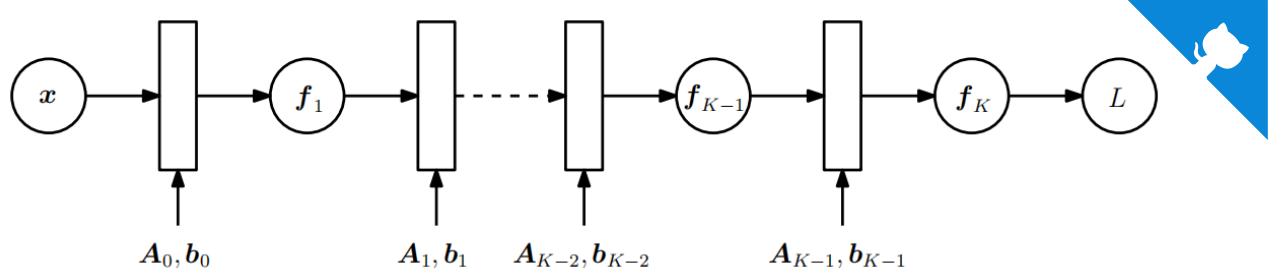


图 5.8 多层神经网络的前向传播

在一般的多层神经网络中，第 i 层中有函数 $f_i(\mathbf{x}_{i-1}) = \sigma(\mathbf{A}_{i-1}\mathbf{x}_{i-1} + \mathbf{b}_{i-1})$ 。

其中 \mathbf{x}_{i-1} 是 $i=1$ 层的输出和一个激活函数 σ ，例如 sigmoid 函数 $\frac{1}{1-e^{-x}}$ ，
 \tanh 或修正线性单元（rectified linear unit, ReLU）。训练这样的模型，我们需要一个损失函数 L ，对其值求关于所有模型参数 $\mathbf{A}_j, \mathbf{b}_j, j = 1, \dots, K$ 的梯度。这同时要求我们求其对模型中各层的输入的梯度。例如，如果我们有输入 \mathbf{x} 和观测值 \mathbf{y} 和一个网络结构（如图 5.8）：

$$\mathbf{f}_0 := \mathbf{x} \quad (5.112)$$

$$\mathbf{f}_i := \sigma_i(\mathbf{A}_{i-1}\mathbf{f}_{i-1} + \mathbf{b}_{i-1}), \quad i = 1, \dots, K, \quad (5.113)$$

我们关心找到使得下面的平方损失最小的 $\mathbf{A}_j, \mathbf{b}_j, j = 1, \dots, K$:

$$L(\boldsymbol{\theta}) = \left\| \mathbf{y} - \mathbf{f}_K(\boldsymbol{\theta}, \mathbf{x}) \right\|^2 \quad (5.114)$$

其中 $\boldsymbol{\theta} = \{\mathbf{A}_0, \mathbf{b}_0, \dots, \mathbf{A}_{K-1}, \mathbf{b}_{K-1}\}$ 。

为得到相对于参数集 $\boldsymbol{\theta}$ 的梯度，我们需要得到 L 对每一层参数 $\theta_j = \{\mathbf{A}_j, \mathbf{b}_j\}, j = 0, \dots, K-1$ 的偏导数。根据链式法则，我们得到

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-1}} = \frac{\partial L}{\partial \mathbf{f}_K} \frac{\partial \mathbf{f}_K}{\partial \boldsymbol{\theta}_{K-1}} \quad (5.115)$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-2}} = \frac{\partial L}{\partial \mathbf{f}_K} \boxed{\frac{\partial \mathbf{f}_K}{\partial \mathbf{f}_{K-1}} \frac{\partial \mathbf{f}_{K-1}}{\partial \boldsymbol{\theta}_{K-2}}} \quad (5.116)$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_{K-3}} = \frac{\partial L}{\partial \mathbf{f}_K} \boxed{\frac{\partial \mathbf{f}_K}{\partial \mathbf{f}_{K-1}} \frac{\partial \mathbf{f}_{K-1}}{\partial \mathbf{f}_{K-2}} \frac{\partial \mathbf{f}_{K-2}}{\partial \boldsymbol{\theta}_{K-3}}} \quad (5.117)$$

$$\frac{\partial L}{\partial \boldsymbol{\theta}_i} = \frac{\partial L}{\partial \mathbf{f}_K} \boxed{\frac{\partial \mathbf{f}_K}{\partial \mathbf{f}_{K-1}} \dots \frac{\partial \mathbf{f}_{i+2}}{\partial \mathbf{f}_{i+1}} \frac{\partial \mathbf{f}_{i+1}}{\partial \boldsymbol{\theta}_i}} \quad (5.118)$$

其中橙色的项是某层输出相对于其输入的偏导数，而蓝色的项是某层的输出相对于参数的偏导数。假设我们已经计算出了 $\frac{\partial L}{\partial \theta_{i+1}}$ ，那么我们可以在计算 $\frac{\partial L}{\partial \theta_i}$ 中省去大量的工作，因为我们只需计算方框中的项。图 5.9 中表示了像这样在网络中反向传递梯度的图示。

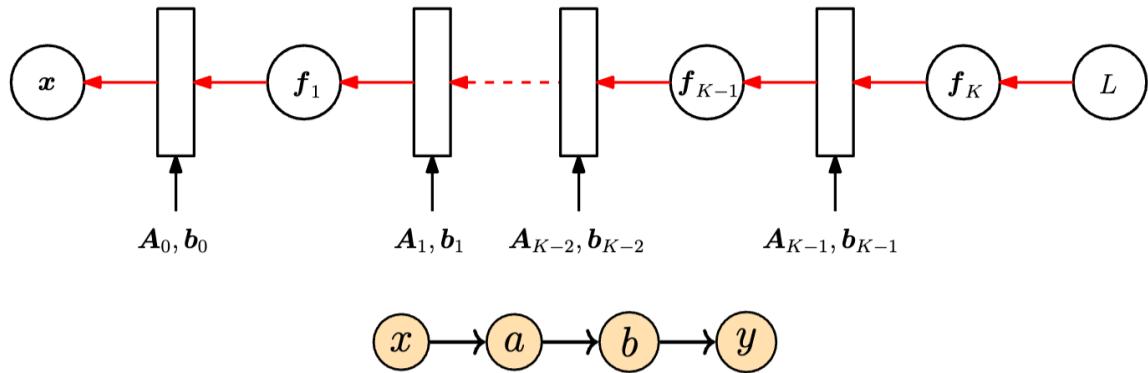


图 5.9 在多层神经网络中使用反向传播计算损失函数的梯度

对此更深入的讨论见 Justin Domke 的 [Lecture Notes](#)

5.6.2 自动微分

事实上，反向传播是数值分析中常采用采用的自动微分 (**automatic differentiation**) 的一种特殊情况。我们可将其看作是一组通过中间变量和链式法则，计算一个函数之（直到机器精度的）精确数值（而非符号）梯度。自动微分始于一系列初等算术运算（如加法、乘法）和初等函数（如 \sin 、 \cos 、 \exp 、 \log ）。通过将链式法则应用于这些操作，我们可以自动计算出相当复杂的函数的梯度。自动微分适用于一般的程序，具有正向和反向两种模式。Baydin et al. (2018) 对机器学习中的自动微分进行了很好的概述。

图5.10显示了一个简单的描述数据流动的图。数据流从输入节点 x 开始，通过中间变量 a, b 最后得到输出 y 。如果我们要计算导数 $\frac{dy}{dx}$ ，我们可以用链式法则：

$$\frac{dy}{dx} = \frac{dy}{db} \frac{db}{da} \frac{da}{dx}. \quad (5.119)$$

直观来讲，正向模式和反向模式的自动微分在处理多重嵌套梯度的乘积顺序上有所不同。由于矩阵乘法有结合律，我们可以采用下面两种不同的方法计算梯度：

$$\frac{dy}{dx} = \left(\frac{dy}{db} \frac{db}{da} \right) \frac{da}{dx}, \quad (5.120)$$

$$\frac{dy}{dx} = \frac{dy}{db} \left(\frac{db}{da} \frac{da}{dx} \right). \quad (5.121)$$

式 (5.120) 就是反向自动微分，因为梯度通过计算图向后传播（即与数据流流向相反）。式 (5.121) 是正向自动微分，其中梯度与数据的流向都是从左到右。

下面，我们将重点关注反向自动微分，即反向传播。在神经网络中，输入的维数通常比标签的维数高得多，反向自动微分在计算上比正向的计算消耗低得多。让我们从一个典型的例子开始理解它。

示例 5.14 反向自动微分 考虑函数

$$f(x) = \sqrt{x^2 + \exp\{x^2\}} + \cos\left(x^2 + \exp\{x^2\}\right) \quad (5.122)$$

这个函数就是 (5.109)。如果我们要在计算机上实现这个函数，我们将使用一些中间变量来节省一些计算：

$$a = x^2, \quad (5.123)$$

$$b = \exp\{a\}, \quad (5.124)$$

$$c = a + b, \quad (5.125)$$

$$d = \sqrt{c}, \quad (5.126)$$

$$e = \cos(c), \quad (5.127)$$

$$f = d + e. \quad (5.128)$$

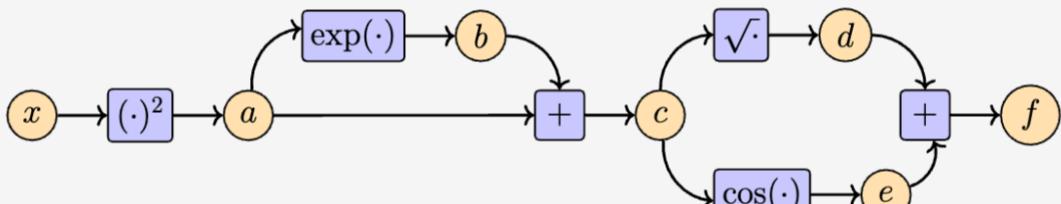


图 5.11 计算图。输入为 x ，输出为函数值 f ，并有中间变量 $a \sim e$



计算该函数的梯度和我们使用链式法则的思想类似。请注意，前面一组方程所需的操作比（5.109）中定义的函数的直接实现要少。图 5.11 中对应的计算图显示了得到函数值 f 所需的数据流和计算。包含中间变量的方程组可以被认为是一个计算图，它被广泛应用于神经网络库的实现。回顾初等函数导数的定义，我们可以直接计算中间变量与其相应输入的导数，就得到了下面这些式子：

$$\frac{\partial a}{\partial x} = 2x \quad (5.129)$$

$$\frac{\partial b}{\partial a} = \exp\{a\} \quad (5.130)$$

$$\frac{\partial c}{\partial a} = 1 = \frac{\partial c}{\partial b} \quad (5.131)$$

$$\frac{\partial d}{\partial c} = \frac{1}{2\sqrt{c}} \quad (5.132)$$

$$\frac{\partial e}{\partial c} = -\sin(c) \quad (5.133)$$

$$\frac{\partial f}{\partial d} = 1 = \frac{\partial f}{\partial e}. \quad (5.134)$$

此时我们看图 5.11 中的计算图，我们可以通过从输出逆向地计算以得到 $\frac{\partial f}{\partial x}$

:

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial d} \frac{\partial d}{\partial c} + \frac{\partial f}{\partial e} \frac{\partial e}{\partial c} \quad (5.135)$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} \frac{\partial c}{\partial b} \quad (5.136)$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} \frac{\partial b}{\partial a} + \frac{\partial f}{\partial c} \frac{\partial c}{\partial a} \quad (5.137)$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a} \frac{\partial a}{\partial x}. \quad (5.138)$$

注意，我们在上面隐式地应用了链式法则。最后我们用上前面求得的初等函数导数代入上面的式子，得到

$$\frac{\partial f}{\partial c} = 1 \cdot \frac{1}{2\sqrt{c}} + 1 \cdot [-\sin(c)] \quad (5.139)$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} \cdot 1 \quad (5.140)$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} \exp\{a\} + \frac{\partial f}{\partial c} \cdot 1 \quad (5.141)$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a} \cdot 2x. \quad (5.142)$$

如果把上面的每个偏导数看做一个变量，我们可以观察到，计算导数所需的计算量与函数值本身的计算量相似。这非常违反直觉，因为式 (5.110) 中 $\frac{\partial f}{\partial x}$ 的比式 (5.109) 中的函数 $f(x)$ 要复杂得多。

一般的自动微分是示例 5.14 的形式化。设 x_1, \dots, x_d 是函数的输入变量， x_{d+1}, \dots, x_{D-1} 是中间变量， x_D 是输出变量。则计算图可以表示为：

$$\text{For } i = d+1, \dots, D : \quad x_i = g_i[x_{\text{Pa}(x_i)}], \quad (5.143)$$

其中， $g_i(\cdot)$ 是初等函数， $x_{\text{Pa}(x_i)}$ 是图中变量 x_i 的所有父节点。给定一个以这种方式定义的函数，我们可以使用链式法则逐步计算该函数的导数。回想一下，根据定义， $f = x_D$ ，因此

$$\frac{\partial f}{\partial x_D} = 1. \quad (5.144)$$

对于其他变量 x_i ，我们应用链式法则

$$\frac{\partial f}{\partial x_i} = \sum_{x_j: x_i \in \text{Pa}(x_j)} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i} = \sum_{x_j: x_i \in \text{Pa}(x_j)} \frac{\partial f}{\partial x_j} \frac{\partial g_j}{\partial x_i}, \quad (5.145)$$

其中， $x_{\text{Pa}(x_i)}$ 是计算图中 x_j 的父节点的集合。式 (5.143) 是一个函数的正向传播，而 (5.145) 是梯度通过计算图的反向传播。在神经网络的训练中，我们将标签的预测误差反向传播。

自动微分应用于可表示为计算图，且组成计算图的基本的函数是可微时的情形。事实上，这个函数甚至可能不是一个数学意义上的函数，而是一个程序。然而并不是所有的程序都能自动微分，例如当我们找不到可微的初等函数时。程序结构中，如循环和 if 语句，在涉及自动微分的处理时需要更为小心。

< 上一章节

下一章节 >

5.5 常用梯度恒等式

5.7 高阶导数





5.7 高阶导数

到目前为止，我们讨论了梯度，即一阶导数。有时，我们关心更高阶的导数，例如当我们使用 Newton 法进行优化时，需要二阶导数（Nocedal and Wright, 2006）。在 5.1.1 节中，我们讨论了 Taylor 级数，即使用多项式近似函数。在多变量情况下，我们可以做同样的事。在接下来我们将详细讨论这一点，但在此之前，我们需要先规定一些记号。

考虑一个函数 $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ 它有两个输入变量 x, y 。我们使用以下符号表示高阶偏导数（和梯度）：

- $\frac{\partial^2 f}{\partial x^2}$ 是 f 关于 x 的二阶偏导数
- $\frac{\partial^n f}{\partial x^n}$ 是 f 关于 x 的 n 阶偏导数
- $\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial x} \right)$ 是先对 x 求偏导，然后对 y 求偏导得到的偏导数
- $\frac{\partial^2 f}{\partial x \partial y}$ 是先对 y 求偏导，然后对 x 求偏导得到的偏导数。

Hessian 矩阵是所有二阶偏导数的集合。

如果 $f(x, y)$ 是二阶（连续）可微函数，那么 $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$ ，二阶偏导和求导顺序无关。相应的 Hessian 矩阵

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \quad (5.147)$$

是对称的。Hessian 矩阵还可以表示为 $\nabla_{x,y}^2 f(x, y)$ 。一般地，对于 $x \in \mathbb{R}^n$ ，函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 的 Hessian 矩阵是一个 $n \times n$ 矩阵。Hessian 矩阵衡量了函数在 (x, y) 附近的局部曲率。

注（向量场的 Hessian 矩阵）： 如果 $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 是一个向量场，Hessian 矩阵是一个 $(m \times n \times n)$ -张量。



< 上一章节

下一章节 >

5.6 反向传播与自动微分

5.8 线性近似和多元 Taylor 级数



5.8 线性近似和多元 Taylor 级数

函数 f 的梯度 ∇f 通常被用作 f 在 x_0 附近的局部线性近似（如图 5.12 所示）：

$$f(x) \approx f(x_0) + (\nabla_x f)(x_0)(x - x_0)$$

其中 $(\nabla_x f)(x_0)$ 是 f 在 x_0 处关于 x 的梯度值。如图 5.12 所示，这种线性近似在靠近 x_0 时精度较高，但随着与 x_0 的距离增大而逐渐失效。值得注意的是，式 (5.148) 本质上是多元泰勒级数在 x_0 处仅保留前两项的特例。

对于定义在 \mathbb{R}^D 上的光滑函数 $f : \mathbb{R}^D \rightarrow \mathbb{R}$ ，设差值向量 $\delta := x - x_0$ ，其泰勒级数展开式为：

$$f(x) = \sum_{k=0}^{\infty} \frac{D_x^k f(x_0)}{k!} \delta^k$$

其中 $D_x^k f(x_0)$ 表示 f 在 x_0 处的第 k 阶全导数。

f 在 x_0 处的 n 阶泰勒多项式由级数的前 $n+1$ 项构成：

$$T_n(x) = \sum_{k=0}^n \frac{D_x^k f(x_0)}{k!} \delta^k$$

在(5.151)和(5.152)中，记号 δ^k 略显草率的，因为当向量 $x \in \mathbb{R}^D$, $D > 1$, 和 $k > 1$ 是未被定义的。事实上， $D_x^k f$ 和 δ^k 都是 k 阶张量，即 k 维数组，如图 5.13 所示。 k 阶张量 $\delta^k \in \overbrace{\mathbb{R}^{D \times D \times \cdots \times D}}^{k \text{ times}}$ 是通过向量 $\delta \in \mathbb{R}^D$ 的 k 重外积得到的，其中“外积”用 \otimes 表示。例如，

$$\boldsymbol{\delta}^2 := \boldsymbol{\delta} \otimes \boldsymbol{\delta} = \boldsymbol{\delta} \boldsymbol{\delta}^\top, \quad \boldsymbol{\delta}^2[i, j] = \delta[i]\delta[j]; \quad (5.153)$$

$$\boldsymbol{\delta}^3 := \boldsymbol{\delta} \otimes \boldsymbol{\delta} \otimes \boldsymbol{\delta}, \quad \boldsymbol{\delta}^3[i, j, k] = \delta[i]\delta[j]\delta[k]. \quad (5.154)$$

这样一来我们可以写出 Taylor 级数

$$D_{\boldsymbol{x}}^k f(\boldsymbol{x}_0) \boldsymbol{\delta}^k = \sum_{i_1=1}^D \cdots \sum_{i_k=1}^D D_{\boldsymbol{x}}^k f(\boldsymbol{x}_0)[i_1, \dots, i_k] \delta[i_1] \cdots \delta[i_k] \quad (5.155)$$

其中 $D_{\boldsymbol{x}}^k f(\boldsymbol{x}_0) \boldsymbol{\delta}^k$ 包含关于 $\boldsymbol{\delta}$ 的 k 阶多项式。

我们现在已经定义了向量场的 Taylor 级数。让我们显式的写下它的前几项 ($k = 0, \dots, 3, \boldsymbol{\delta} := \boldsymbol{x} - \boldsymbol{x}_0$) :

- $k = 0 : D_x^0 f(x_0) \boldsymbol{\delta}^0 = f(x_0) \in \mathbb{R}$
- $k = 1 : D_x^1 f(x_0) \boldsymbol{\delta}^1 = \underbrace{\nabla_x f(x_0)}_{1 \times D} \underbrace{\boldsymbol{\delta}}_{D \times 1} = \sum_{i=1}^D \nabla_x f(x_0)[i] \delta[i] \in \mathbb{R}$
- $k = 2 : D_x^2 f(x_0) \boldsymbol{\delta}^2 = \text{tr}(\underbrace{H(x_0)}_{D \times D} \underbrace{\boldsymbol{\delta}}_{D \times 1} \underbrace{\boldsymbol{\delta}^\top}_{1 \times D}) = \boldsymbol{\delta}^\top H(x_0) \boldsymbol{\delta} = \sum_{i=1}^D \sum_{j=1}^D H[i, j] \delta[i] \delta[j] \in \mathbb{R}$
- $k = 3 : D_x^3 f(x_0) \boldsymbol{\delta}^3 = \sum_{i=1}^D \sum_{j=1}^D \sum_{k=1}^D D_x^3 f(x_0)[i, j, k] \delta[i] \delta[j] \delta[k] \in \mathbb{R}$ 其中 $H(x_0)$ 是 f 在 x_0 处的 Hessian 矩阵

例5.15 (两个变量函数的 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。在开始之前，我们先讨论一下我们的预期：(5.161) 中的函数是一个 3 次多项式。而我们要找的 Taylor 级数展开，本身就是多项式的线性组合。这个 Taylor 级数不可能包含四阶或更高阶的项。这意味着计算 (5.151) 的前四项足矣。我们从常数项和一阶导开始计算：

$$f(1, 2) = 13 \quad (5.162)$$

$$\frac{\partial f}{\partial x} = 2x + 2y \Rightarrow \frac{\partial f}{\partial x}(1, 2) = 6 \quad (5.163)$$

$$\frac{\partial f}{\partial y} = 2x + 3y^2 \Rightarrow \frac{\partial f}{\partial y}(1, 2) = 14. \quad (5.164)$$

因此，我们得到

$$D_{x,y}^1 f(1,2) = \nabla_{x,y} f(1,2) = \begin{bmatrix} \frac{\partial f}{\partial x}(1,2) & \frac{\partial f}{\partial y}(1,2) \end{bmatrix} = [\begin{array}{cc} 6 & 14 \end{array}]$$

进而有

$$\frac{D_{x,y}^1 f(1,2)}{1!} \delta = [\begin{array}{cc} 6 & 14 \end{array}] \begin{bmatrix} x-1 \\ y-2 \end{bmatrix} = 6(x-1) + 14(y-2)$$

注意, $D_x^1 f(x_0) \delta^1$ 只有线性项, 换句话说, 它只含有一阶多项式。现在计算二阶偏导:

$$\frac{\partial^2 f}{\partial x^2} = 2 \implies \frac{\partial^2 f}{\partial x^2}(1,2) = 2 \quad (5.167)$$

$$\frac{\partial^2 f}{\partial y^2} = 6y \implies \frac{\partial^2 f}{\partial y^2}(1,2) = 12 \quad (5.168)$$

$$\frac{\partial^2 f}{\partial y \partial x} = 2 \implies \frac{\partial^2 f}{\partial y \partial x}(1,2) = 2 \quad (5.169)$$

$$\frac{\partial^2 f}{\partial x \partial y} = 2 \implies \frac{\partial^2 f}{\partial x \partial y}(1,2) = 2. \quad (5.170)$$

得到了所有二阶偏导数后, 我们就有了 Hessian 矩阵:

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 6y \end{bmatrix} \quad (5.171)$$

进而有

$$H(1,2) = \begin{bmatrix} 2 & 2 \\ 2 & 12 \end{bmatrix} \in \mathbb{R}^{2 \times 2}. \quad (5.172)$$

因此, Taylor 级数的下一项由下式给出:

$$\frac{D_{x,y}^2 f(1,2)}{2!} \delta^2 = \frac{1}{2} \delta^\top H(1,2) \delta \quad (5.173a)$$

$$= \frac{1}{2} \begin{bmatrix} x-1 & y-2 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 2 & 12 \end{bmatrix} \begin{bmatrix} x-1 \\ y-2 \end{bmatrix} \quad (5.173b)$$

$$= (x-1)^2 + 2(x-1)(y-2) + 6(y-2)^2. \quad (5.173c)$$

其中 $D_{x,y}^2 f(1,2) \delta^2$ 仅含有二阶项, 也就是二阶多项式 (二次型)。最后计算三阶导数:

$$D_{x,y}^3 f = \begin{bmatrix} \frac{\partial \mathbf{H}}{\partial x} & \frac{\partial \mathbf{H}}{\partial y} \end{bmatrix} \in \mathbb{R}^{2 \times 2}, \quad (5.174)$$

$$D_{x,y}^3 f[:, :, 1] = \frac{\partial \mathbf{H}}{\partial x} = \begin{bmatrix} \frac{\partial^3 f}{\partial x^3} & \frac{\partial^3 f}{\partial x^2 \partial y} \\ \frac{\partial^3 f}{\partial x \partial y \partial x} & \frac{\partial^3 f}{\partial x \partial y^2} \end{bmatrix}, \quad (5.175)$$

$$D_{x,y}^3 f[:, :, 2] = \frac{\partial \mathbf{H}}{\partial y} = \begin{bmatrix} \frac{\partial^3 f}{\partial y \partial x^2} & \frac{\partial^3 f}{\partial y \partial x \partial y} \\ \frac{\partial^3 f}{\partial y^2 \partial x} & \frac{\partial^3 f}{\partial y^3} \end{bmatrix}. \quad (5.176)$$

由于 Hessian 矩阵 (5.171) 中的大多数二阶偏导数是常数，唯一非零的三阶偏导数是

$$\frac{\partial^3 f}{\partial y^3} = 6 \Rightarrow \frac{\partial^3 f}{\partial y^3}(1, 2) = 6. \quad (5.177)$$

显然，我们可以看到该函数的更高阶导数和3阶混合偏导（例如， $\frac{\partial f^3}{\partial x^2 \partial y}$ ）都为零，因此

$$D_{x,y}^3 f[:, :, 1] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad D_{x,y}^3 f[:, :, 2] = \begin{bmatrix} 0 & 0 \\ 0 & 6 \end{bmatrix} \quad (5.178)$$

于是有

$$\frac{D_{x,y}^3 f(1, 2)}{3!} \delta^3 = (y - 2)^3. \quad (5.179)$$

现在我们已经算出了 Taylor 级数的所有非零项。因此 f 在 $(x_0, y_0) = (1, 2)$ 处的 Taylor 级数是



$$f(x) = f(1, 2) + D_{x,y}^1 f(1, 2)\delta + \frac{D_{x,y}^2 f(1, 2)}{2!}\delta^2 + \frac{D_{x,y}^3 f(1, 2)}{3!}(5.180\text{a})$$

$$= f(1, 2) + \frac{\partial f(1, 2)}{\partial x}(x - 1) + \frac{\partial f(1, 2)}{\partial y}(y - 2)$$

$$+ \frac{1}{2!} \left(\frac{\partial^2 f(1, 2)}{\partial x^2}(x - 1)^2 + \frac{\partial^2 f(1, 2)}{\partial y^2}(y - 2)^2 + 2 \frac{\partial^2 f(1, 2)}{\partial x \partial y}(x - 1)(y - 2) \right) \\ + \frac{1}{6} \frac{\partial^3 f(1, 2)}{\partial y^3}(y - 2)^3 \quad (5.180\text{b})$$

$$= 13 + 6(x - 1) + 14(y - 2)$$

$$+ (x - 1)^2 + 6(y - 2)^2 + 2(x - 1)(y - 2) + (y - 2)^3 \quad (5.180\text{c})$$

在本例中，我们得到了 (5.161) 中多项式的 Taylor 级数展开，即 (5.180c) 中的多项式。它与 (5.161) 中的原始多项式完全相同。在本例的特殊情况下，这样的结果并不令人惊讶，因为原始函数就是一个三阶多项式，我们只是在 (5.180c) 中将其写成了一阶、二阶和三阶多项式的线性组合。

< 上一章节

下一章节 >

5.7 高阶导数

5.9 拓展阅读



5.9 拓展阅读

关于矩阵微分的更多细节，以及对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，读者可以参考 Griewank and Walther (2003)、Griewank and Walther (2008)、Elliott (2009) 及其中的参考文献。

在机器学习（以及其他学科）中，我们经常需要计算期望，即我们需要求解形如

$$\mathbb{E}_x[f(x)] = \int f(x)p(x)dx \quad (5.181)$$

的积分。即使 $p(x)$ 的形式比较简单（例如 Gauss 分布），这个积分通常也没有解析解。然而使用 f 的 Taylor 级数是找到近似解的一种方法。假设 $p(x) = \mathcal{N}(\mu, \Sigma)$ 是 Gauss 分布，那么在 μ 附近的一阶 Taylor 级数展开将非线性函数 f 局部线性化。对于线性函数，如果 $p(x)$ 是 Gauss 分布，我们可以精确计算乘积均值和协方差（见6.5节）。这样的性质在 **扩展 Kalman 滤波** (**extended kalman filter, EKF**) (Maybeck, 1979) 和 **非线性系统**（也称为“状态空间模型”）的在线状态估计中被大量应用。其他用于近似 (5.181) 中积分的确定性的方法包括无需梯度计算的 **无迹变换** (**unsecnted transform**) (Julier和Uhlmann, 1997) 或者使用二阶 Taylor 展开的 (Hessian 矩阵) 以对 $p(x)$ 进行局部 Gauss 近似的 **Laplace** 近似 (MacKay, 2003; Bishop, 2006; Murphy, 2002)。

< 上一章节

下一章节 >

5.8 线性近似和多元 Taylor 级数

习题



习题

5.1

计算 $f(x) = \log(x^4) \sin(x^3)$ 的导数 $f'(x)$ 。

5.2

计算 Logistic 函数 $f(x) = \frac{1}{1 + \exp(-x)}$ 的导数 $f'(x)$ 。

5.3

计算函数 $f(x) = \exp\left[-\frac{1}{2\sigma^2}(x - \mu)^2\right]$ 的导数 $f'(x)$, 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:

$$f_1(x) = \sin(x_1) \cos(x_2), \quad x \in \mathbb{R}^2 \tag{1}$$

$$f_2(x, y) = x^\top y, \quad x, y \in \mathbb{R}^n \tag{2}$$

$$f_3(x) = xx^\top, \quad x \in \mathbb{R}^n \tag{3}$$

(a) $\frac{\partial f_i}{\partial x}$ 的维度是多少?

(b) 计算 Jacobi 矩阵。



5.6

对 f 关于 t 求导，对 g 关于 x 求导，其中 $f(t) = \sin(\log(t^\top t))$, $t \in \mathbb{R}^D$, $g(X) = \text{tr}(AXB)$, $A \in \mathbb{R}^{D \times E}$, $X \in \mathbb{R}^{E \times F}$, $B \in \mathbb{R}^{F \times D}$, 其中 $\text{tr}(\cdot)$ 表示矩阵的迹。

5.7

使用链式法则计算以下函数的导数 $\frac{df}{dx}$ 。并写出每个偏导数的维度。
(a) $f(z) = \log(1 + z)$, $z = x^\top x$, $x \in \mathbb{R}^D$

(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 $\sin(\cdot)$ 作用于 z 的每个分量。

5.8

计算以下函数的导数 $\frac{df}{dx}$ 。
(a) 使用链式法则，并给出每个一阶偏导数的维度

$$f(z) = \exp\left(-\frac{1}{2}z\right) \quad (4)$$

$$z = g(\mathbf{y}) = \mathbf{y}^\top \mathbf{S}^{-1} \mathbf{y} \quad (5)$$

$$\mathbf{y} = h(\mathbf{x}) = \mathbf{x} - \boldsymbol{\mu} \quad (6)$$

其中 $\mathbf{x}, \boldsymbol{\mu} \in \mathbb{R}^D$, $\mathbf{S} \in \mathbb{R}^{D \times D}$ 。

(b)

$$f(x) = \text{tr}(xx^\top + \sigma^2 I),$$

其中 $x \in \mathbb{R}^D$ 。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，给出每个一阶偏导数的维度（不需要显式计算偏导数的乘积）。

$$f = \tanh(z) \in \mathbb{R}^M \quad (7)$$

$$z = Ax + b, \quad x \in \mathbb{R}^N, A \in \mathbb{R}^{M \times N}, b \in \mathbb{R}^M. \quad (8)$$

这里， \tanh 应用于 z 的每个分量。



5.9

我们定义

$$g(z, \nu) := \log p(\mathbf{x}, z) - \log q(z, \nu) \quad (9)$$

$$z := t(\epsilon, \nu) \quad (10)$$

对于可微函数 p, q, t 以及 $\mathbf{x} \in \mathbb{R}^D, z \in \mathbb{R}^E, \nu \in \mathbb{R}^F, \epsilon \in \mathbb{R}^G$ 。使用链式法则计算梯度

$$\frac{d}{d\nu} g(z, \nu).$$

< 上一章节

5.9 拓展阅读



第6章 概率分布

概率，简而言之，是研究不确定性的学科。概率可以被视为某一事件发生次数的比例，或者对某一事件发生的信任程度。然后，我们希望利用这种概率来衡量实验中某事件发生的可能性。正如第1章所述，我们经常量化数据中的不确定性、机器学习模型中的不确定性以及模型预测结果中的不确定性。量化不确定性需要随机变量的概念，随机变量是一个函数，它将随机实验的结果映射到我们感兴趣的一组属性上。与随机变量相关联的是一个函数，它测量某一特定结果（或结果集）发生的概率；这被称为概率分布。

概率分布是其他概念（如概率建模（第8.4节）、图形模型（第8.5节）和模型选择（第8.6节））的基石。在下一节中，我们将介绍定义概率空间的三个概念（样本空间、事件和事件的概率）以及它们与第四个概念——随机变量的关系。本次介绍特意采用了较为直观的方式，因为过于严谨的阐述可能会掩盖这些概念背后的直觉。本章介绍的概念概要如图6.1所示。

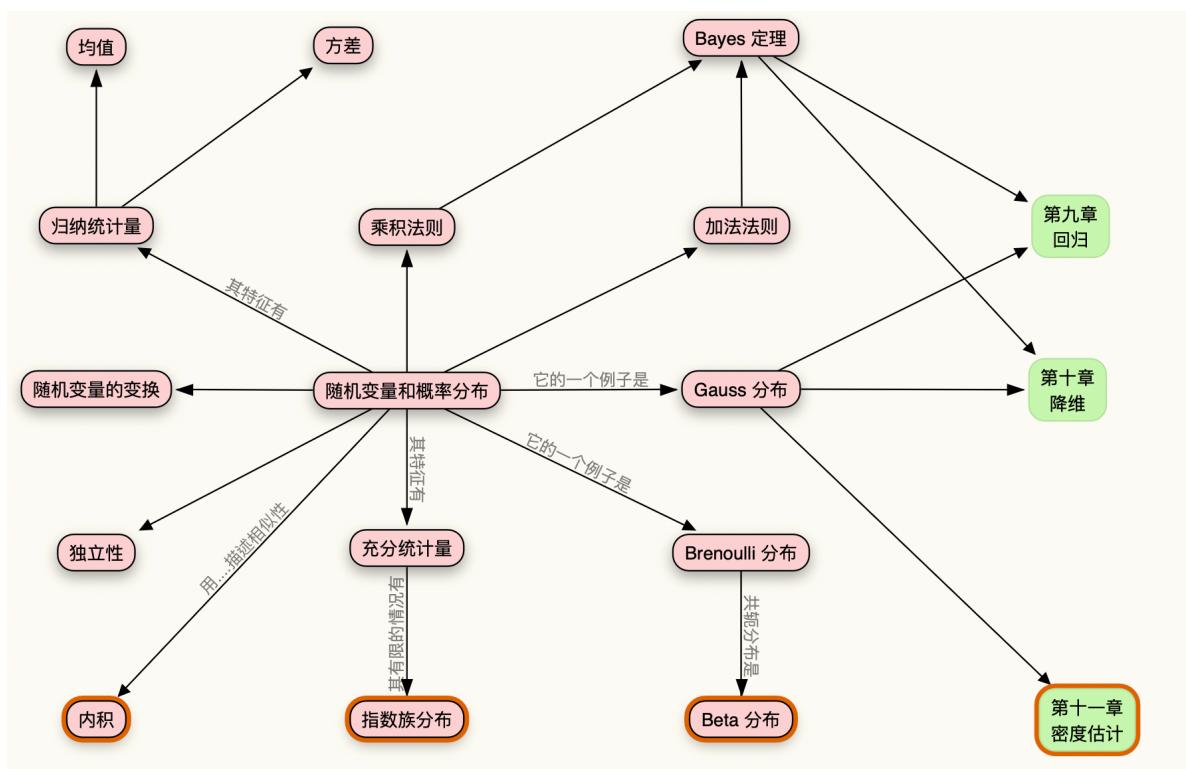


图6.1 与随机变量和概率分布相关的概念的思维导图

< 上一章节

第五章 向量微积分

下一章节 >

第七章 连续优化





6.1 概率空间的构建

概率论旨在定义一个数学结构来描述实验结果的随机性。例如，在抛掷一枚硬币时，我们无法确定结果，但通过大量抛掷硬币，我们可以观察到平均结果中的规律性。利用这种概率的数学结构，目标是进行自动化推理，从这个意义上说，概率是对逻辑推理的泛化（Jaynes, 2003）。

6.1.1 哲学问题

在构建自动化推理系统时，经典布尔逻辑不允许我们表达某些形式的合理推理。考虑以下场景：我们观察到**A**为假。我们发现**B**变得不那么可信，尽管从经典逻辑中无法得出这一结论。我们观察到**B**为真，似乎**A**又变得更为可信。我们每天都在使用这种形式的推理。比如，我们在等一位朋友，并考虑三种可能性：**H1**，她准时到达；**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，尽管逻辑上并不要求我们必须这样做。最后，我们可能会认为**H3**是可能的，但继续认为它非常不可能。那么，我们如何得出结论认为**H2**是最合理的答案？从这个角度看，“合理的概率论可以被视为布尔逻辑的一种泛化”。在机器学习的推理背景下，它经常被这样应用，以形式化并扩展自动化推理系统的设计。关于如何以真值概率论的假值为基础构建推理系统的进一步论证，可参见Pearl（1988）。

概率的哲学基础以及它应该如何以某种方式（Jaynes, 2003）与我们认为在逻辑上应该为真的事物相关联，这一问题被Cox研究过（Jaynes, 2003）。另一种思考方式是，如果我们精确地使用常识，最终会构建出概率。E. T. Jaynes（1922-1998）确定了三个数学标准，这些标准必须适用于所有可能性：

1. 可能性的程度由实数表示。
2. 这些数字必须基于常识的规则。
3. 所得推理必须是一致的，其中“一致”一词包含以下三层含义：
 - (a) 一致性或无矛盾性：当可以通过不同方式达到相同结果时，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) 可再现性：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程度的可能性。

Cox-Jaynes定理证明了这些可能性足以定义适用于可能性 p 的普遍数学规则，直到通过任意单调函数进行变换。至关重要的是，这些规则就是概率的规则。

备注。在机器学习和统计学中，概率有两种主要解释：贝叶斯解释和频率解释（Bishop, 2006; Efron and Hastie, 2016）。贝叶斯解释使用概率来指定用户对某个事件发生的不确定性程度。它有时被称为“主观概率”或“信念程度”。频率解释则考虑感兴趣事件相对于发生事件总数的相对频率。当数据无限时，某事件的概率被定义为该事件的相对频率。

一些关于概率模型的机器学习文献使用了不严谨的记号和术语，这会造成困惑。本文也不例外。多个不同的概念都被称为“概率分布”，读者往往需要从上下文中分辨其含义。一个有助于理解概率分布的技巧是检查我们是在尝试对分类事物（离散随机变量）还是连续事物（连续随机变量）进行建模。我们在机器学习中解决的问题类型与我们是考虑分类模型还是连续模型密切相关。

6.1.2 概率与随机变量

在讨论概率时，经常会混淆三个不同的概念。首先是概率空间的概念，它使我们能够量化概率的想法。然而，我们大多不直接处理这个基本的概率空间。相反，我们处理的是随机变量（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。我们将在本节中介绍前两个概念，并在6.2节中详细阐述第三个概念。

现代概率论基于Kolmogorov提出的一组公理（Grinstead and Snell, 1997; Jaynes, 2003），这些公理引入了样本空间、事件空间和概率测度这三个概念。概率空间模型用于模拟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，通常表示为 Ω 。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 \mathcal{A} . 事件空间是实验潜在结果的集合。如果实验结束时我们可以观察到某个特定结果 $\omega \in \Omega$ 是否在 \mathcal{A} 中，则样本空间 Ω 的子集 \mathcal{A} 就属于事件空间 \mathcal{A} 。事件空间 \mathcal{A} 是通过考虑 Ω 的子集集合获得的，对于离散概率分布（第6.2.1节）， \mathcal{A} 通常也是 Ω 的幂集。

概率 P . 对于每个事件 $A \in \mathcal{A}$, 我们关联一个数 $P(A)$, 它衡量了事件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位于区间 $[0,1]$ 内, 样本空间 Ω 中所有结果的总概率必须为 1, 即 $P(\Omega) = 1$ 。给定一个概率空间 (Ω, \mathcal{A}, P) , 我们希望用它来模拟一些现实世界的现象。在机器学习中, 我们通常避免明确提及概率空间, 而是指关注量的概率, 我们用 \mathcal{T} 来表示。在本书中, 我们将 \mathcal{T} 称为目标空间, 并将 \mathcal{T} 的元素称为状态。我们引入一个函数 $\bar{X} : \dot{\Omega} \rightarrow \mathcal{T}$, 它接受 Ω 的一个元素 (一个结果) 并返回一个特定的关注量 x , 即 \mathcal{T} 中的一个值。从 Ω 到 \mathcal{T} 的这种关联/映射被称为随机变量。例如, 在抛两枚硬币并计算正面朝上次数的情况下, 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, $\mathcal{T} = \{0, 1, 2\}$, 我们关注的是 \mathcal{T} 元素上的概率。对于有限的样本空间 Ω 和有限的目标空间 \mathcal{T} , 与随机变量对应的函数本质上是一个查找表。对于 \mathcal{T} 的任何子集 $S \subseteq \mathcal{T}$, 我们将 $P_X(S) \in [0, 1]$ (误解概率) 与随机变量 X 对应的特定事件相关联。示例 6.1 提供了术语的具体说明。

备注 上述样本空间 Ω 在不同的书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动态系统中的状态(Hasselblatt and Katok, 2003)。其他有时用于描述 Ω 的名称包括: “样本描述空间”、“可能性空间”和“事件空间”。

示例 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。对于更温和且包含许多例子的概率论介绍, 可以在 Walpole et al. (2011) 的第 2 章中找到。

考虑一个统计实验, 我们模拟一个游乐场游戏, 该游戏包括从一个袋子中抽取两枚硬币 (放回原袋)。袋子中有来自美国 (用 \$ 表示) 和英国 (用 £ 表示) 的硬币, 因为我们从袋子中抽取两枚硬币, 所以总共有四种结果。这个实验的状态空间或样本空间 Ω 因此是 $(S, S), (S, E), (E, S), (E, E)$ (注意: 这里的符号可能有些不一致, 通常我们会用更具体的符号如 (,), (, £), (£, \$), (£, £) 来表示, 但为了与原文保持一致, 我们保留原符号)。假设袋子中硬币的组成是这样的: 随机抽取一枚硬币得到 \$ 的概率是 0.3。

我们感兴趣的事件是重复抽取中返回 \$ 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 \mathcal{T} , 后者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，我们可以得到零次，一次，或两次，因此 $\mathcal{T} = \{0, 1, 2\}$ 。随机变量 X （一个函数或查找表）可以表示如下表：

$$\begin{aligned}X((\$, \$)) &= 2 \\X((\$, \mathcal{L})) &= 1 \\X((\mathcal{L}, \$)) &= 1 \\X((\mathcal{L}, \mathcal{L})) &= 0\end{aligned}$$

由于我们在抽取第二枚硬币之前将第一枚硬币放回袋子，这意味着两次抽取是相互独立的，我们将在第 6.4.5 节中讨论这一点。请注意，有两个实验结果映射到同一个事件，即只有其中一次抽取返回。因此， $X\$$ 的概率质量函数（第 6.2.1 节）由下式给出：

$$\begin{aligned}P(X = 2) &= P((\$, \$)) \\&= P(\$) \cdot P(\$) \\&= 0.3 \cdot 0.3 = 0.09 \\P(X = 1) &= P((\$, \mathcal{L}) \cup (\mathcal{L}, \$)) \\&= P((\$, \mathcal{L})) + P((\mathcal{L}, \$)) \\&= 0.3 \cdot (1 - 0.3) + (1 - 0.3) \cdot 0.3 = 0.42 \\P(X = 0) &= P((\mathcal{L}, \mathcal{L})) \\&= P(\mathcal{L}) \cdot P(\mathcal{L}) \\&= (1 - 0.3) \cdot (1 - 0.3) = 0.49\end{aligned}$$

在计算中，我们将两个不同的概念等同起来，即 X 的输出概率和 Ω 中样本的概率。例如，在 (6.7) 中我们说 $P(X = 0) = P((\mathcal{L}, \mathcal{L}))$ 。考虑随机变量 $X : \Omega \rightarrow \mathcal{T}$ 和一个子集 $S \subseteq \mathcal{T}$ （例如， \mathcal{T} 的一个单元素，如投掷两枚硬币时得到一个正面的结果）。令 $X^{-1}(S)$ 是 S 在 X 下的原像，即 Ω 中在 X 下映射到 S 的元素集合； $\omega \in \Omega : X(\omega) \in S$ 。理解从 Ω 中的事件通过随机变量 X 转换到概率的一种方式是将其与 S 的原像的概率相关联 (Jacod 和 Protter, 2004)。对于 $S \subseteq \mathcal{T}$ ，我们使用以下符号：

$$P_X(S) = P(X \in S) = P(X^{-1}(S)) = P(\omega \in \Omega : X(\omega) \in S).$$

(6.8)

(6.8) 的左侧是我们感兴趣的可能结果集（例如， $=$ 的数量 = 1）的概率。通过随机变量 X ，它将状态映射到结果，我们在 (6.8) 的右侧看到这是具有某种性质（例如， S 包含 ϵ, ϵ ）的状态集（在 Ω 中）的概率。我们说随机变量 X 根据特定的概率分布 P_X 分布，该分布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 或等价地 $P \circ X^{-1}$ 是随机变量 X 的分布律或分布。

备注：目标空间，即随机变量 X 的值域 \mathcal{T} ，用于指示概率空间的类型，即 T 随机变量。当 \mathcal{T} 是有限或可数无限时，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 $\mathcal{T} = \mathbb{R}$ 或 $\mathcal{T} = \mathbb{R}^D$ 。

6.1.3 统计学

概率论和统计学经常被放在一起讨论，但它们关注的是不确定性的不同方面。对比它们的一种方式是考虑所研究的问题类型。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通过随机变量来捕捉，并利用概率规则来推导出所发生的事情。在统计学中，我们观察到某件事情已经发生，并试图找出解释这些观察结果的潜在过程。从这个意义上说，机器学习的目标更接近统计学，即构建一个能够充分表示数据生成过程的模型。我们可以利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，我们关注泛化误差（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，而这些实例与我们到目前为止已经看到的实例并不相同。对未来性能的这种分析依赖于概率论和统计学，其中大部分内容超出了本章将介绍的范围。有兴趣的读者可以查阅Boucheron et al. (2013) 以及 Shalev-Shwartz 和 Ben-David (2014) 的著作。我们将在第8章中进一步了解统计学。

下一章节 >

6.2 离散概率与连续概率



6.2 离散概率与连续概率

让我们将注意力集中在如何描述6.1节中介绍的事件的概率上。根据目标空间是离散的还是连续的，描述分布的自然方式是不同的。当目标空间 \mathcal{T} 是离散的时，我们可以指定随机变量 X 取特定值 $x \in \mathcal{T}$ 的概率，表示为 $P(X = x)$ 。对于离散随机变量 X ，表达式 $P(X = x)$ 被称为概率质量函数。当目标空间 \mathcal{T} 是连续的，例如实数线 R ，则更自然地指定随机变量 X 位于某个区间内的概率，对于 $a < b$ ，表示为 $P(a \leq X \leq b)$ 。按照惯例，我们指定随机变量 X 小于特定值 x 的概率，表示为 $P(X \leq x)$ 。对于连续随机变量 X ，表达式 $P(X \leq x)$ 被称为累积分布函数。我们将在6.2.2节中讨论连续随机变量。我们将在6.2.3节中重新回顾术语，并对比离散和连续随机变量。

备注：我们将使用“单变量分布”一词来指代单个随机变量的分布（其状态用非粗体 x 表示）。我们将涉及多个随机变量的分布称为多变量分布，并通常考虑随机变量的向量（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给出了一个示例。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共同出现的条目

$$P(X = x_i, Y = y_j) = \frac{n_{ij}}{N},$$

(6.9)

其中 n_{ij} 是状态为 x_i 和 y_j 的事件数， N 是事件的总数。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展示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 ， $X = x$ 且 $Y = y$ 的概率（简略地）写为 $p(x, y)$ ，并称为联合概率。我们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 $p(x, y)$ 的原因。无论随机变量 Y 的值如何， X 取值 x 的边缘概率（简略地）写为 $p(x)$ 。我们用 $X \sim p(x)$ 来表示随机变量 X 根据 $p(x)$ 分布。如果我们只考虑 $X = x$ 的情况，那么 $Y = y$ 的实例比例（条件概率）简略地写为 $p(y | x)$ 。

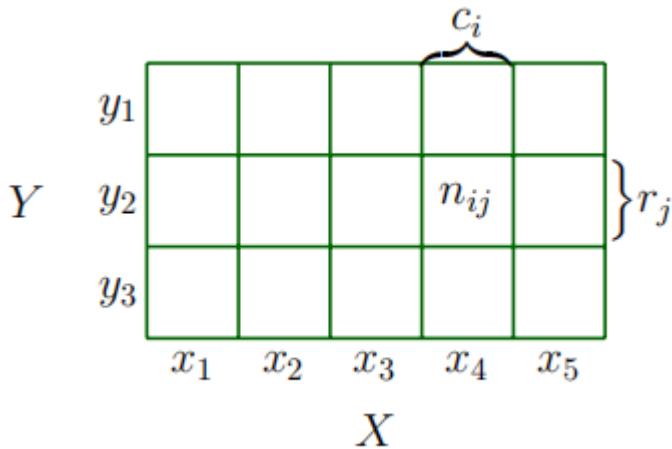


图6.2具有随机变量 X 和 Y 的离散二变量概率质量函数的可视化。此图改编自Bishop (2006)。

例6.2

考虑两个随机变量 X 和 Y , 其中 X 有五种可能的状态, 而 Y 有三种可能的状态, 如图6.2所示。我们用 n_{ij} 表示状态为 $X = x_i$ 和 $Y = y_j$ 的事件数, 用 N 表示事件的总数。值 c_i 是第 i 列各个频率的和, 即 $c_i = \sum_{j=1}^3 n_{ij}$ 。类似地, 值 r_j 是行和, 即 $r_j = \sum_{i=1}^5 n_{ij}$ 。使用这些定义, 我们可以紧凑地表示 X 和 Y 的分布。

每个随机变量的概率分布, 即边缘概率, 可以看作是某一行或列的和

(6.10)

$$P(X = x_i) = \frac{c_i}{N} = \frac{\sum_{j=1}^3 n_{ij}}{N}$$

$$P(Y = y_j) = \frac{r_j}{N} = \frac{\sum_{i=1}^5 n_{ij}}{N},$$

并且

(6.11)

其中 c_i 和 r_j 分别是概率表的第 i 列和第 j 行的值。按照惯例, 对于具有有限数量事件的离散随机变量, 我们假设概率之和为1, 即



$$\sum_{i=1}^5 P(X = x_i) = 1 \quad \text{和} \quad \sum_{j=1}^3 P(Y = y_j) = 1.$$

(6.12)

条件概率是特定单元格中某一行或列的比例。例如，给定 X 的条件下 Y 的条件概率是

$$P(Y = y_j \mid X = x_i) = \frac{n_{ij}}{c_i},$$

而给定 Y 的条件下 X 的条件概率是 (6.14)

$$P(X = x_i \mid Y = y_j) = \frac{n_{ij}}{r_j}.$$

在机器学习中，我们使用离散概率分布来模拟分类变量，即取有限个无序值的变量。它们可以是分类特征，比如用于预测一个人薪水时所用的大学学位，也可以是分类标签，比如在手写识别中使用的字母表中的字母。离散分布也常被用于构建结合了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，即目标空间是实数线 \mathbf{R} 上的区间。在本书中，我们假设可以对实值随机变量进行操作，就像我们拥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并不精确：一是当我们无限次重复某件事时；二是当我们想从某个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讨论连续分布时，如高斯分布（第6.5节）。就我们的目的而言，这种不精确性允许我们对概率进行更简洁的介绍。

备注。在连续空间中，存在两个额外的技术性问题，这两个问题都是违反直觉的。首先，所有子集的集合（用于在6.1节中定义事件空间 \mathcal{A} ）的行为不够良好。 \mathcal{A} 需要被限制在集合补集、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数元素来获得）变得棘手。集合的大小被称为其测度。例如，离散集合的基数、实数集 \mathbf{R} 中区间的长度和 \mathbb{R}^d 中区域的体积都是测度。在集合运算下表现良好且还具有拓扑结构的集合被称为Borel σ -代数。Betancourt详细介绍了从集合论中

仔细构造概率空间的方法，而没有陷入技术细节中；对于更精确的构造，我们参考 Billingsley (1995) 和 Jacod 及 Protter (2004)。see <https://tinyurl.com/yb3t6mfd>.

在这本书中，我们考虑具有相应Borel σ -代数的实值随机变量。我们认为取值在 \mathbb{R}^D 中的随机变量是实值随机变量的向量。

定义6.1（概率密度函数）。如果函数 $f : \mathbb{R}^D \rightarrow \mathbb{R}$ 满足以下条件，则称为概率密度函数 (pdf)：

$$1. \forall \mathbf{x} \in \mathbb{R}^D : f(\mathbf{x}) \geq 0$$

2. 其积分存在，且

(6.15)

$$\int_{\mathbb{R}^D} f(\mathbf{x}) d\mathbf{x} = 1.$$

对于离散随机变量的概率质量函数 (pmf)，(6.15)中的积分被替换为求和(6.12)。

请注意，概率密度函数是任何非负且积分为1的函数。我们通过以下方式将随机变量 X 与该函数 f 相关联：

$$P(a \leq X \leq b) = \int_a^b f(x) dx,$$

(6.16)

其中 $a, b \in \mathbb{R}$ 且 $x \in \mathbb{R}$ 是连续随机变量 X 的结果。通过考虑向量 $\mathbf{x} \in \mathbb{R}$ ，类似地定义 $\mathbf{x} \in \mathbb{R}^D$ 的状态。这种关联(6.16)称为随机变量 X 的概率法或分布。

备注。与离散随机变量不同，连续随机变量 X 取特定值 x 的概率 $P(X = x)$ 为零。这就像在(6.16)中尝试指定一个区间，其中 $a = b$ 。

定义6.2（累积分布函数）。具有状态 $\mathbf{x} \in \mathbb{R}^D$ 的多元实值随机变量 X 的累积分布函数 (cdf) 由下式给出：

(6.17)

$$F_X(\mathbf{x}) = P(X_1 \leq x_1, \dots, X_D \leq x_D),$$

其中 $\mathbf{X} = [X_1, \dots, X_D]^\top$, $\mathbf{x} = [x_1, \dots, x_D]^\top$, 且右侧表示随机变量 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，即

$$F_X(\mathbf{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_D} f(z_1, \dots, z_D) dz_1 \cdots dz_D.$$

(6.18)

备注。我们重申，在讨论分布时，实际上有两个不同的概念。第一个是pdf（用 $f(x)$ 表示），它是一个非负且积分为1的函数。第二个是随机变量 \bar{X} 的法则，即将随机变量 X 与pdf $f(x)$ 相关联。

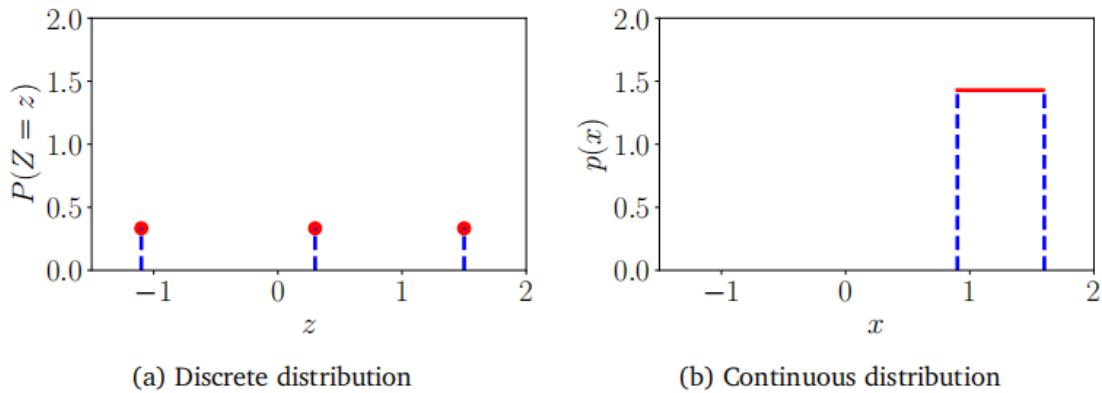


图6.3(a)离散分布和(b)连续均匀分布的例子。有关分布的详细信息，请参见示例6.3。

对于这本书的大部分内容，我们将不会使用符号 $f(x)$ 和 $F_X(x)$ ，因为我们大多不需要区分pdf和cdf。但是，我们需要小心第6.7节中的pdfs和cdfs。

6.2.3 离散分布与连续分布的对比

回顾6.1.2节，概率是正的，且所有概率之和为1。对于离散随机变量（见式(6.12)），这意味着每个状态的概率必须位于区间[0,1]内。然而，对于连续随机变量，归一化（见式(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中通过离散和连续随机变量的均匀分布来说明这一点。

例6.3

我们考虑均匀分布的两个例子，其中每个状态发生的可能性都相等。这个例子说明了离散概率分布和连续概率分布之间的一些差异。



设 Z 是一个具有三个状态 $\{z = -1.1, z = 0.3, z = 1.5\}$ 的离散均匀随机变量。其概率质量函数可以用概率值的表格来表示：

z	-1.1	0.3	1.5
$P(Z = z)$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

或者，我们可以将其视为一个图形（图6.3(a)），其中我们使用了一个事实，即状态可以位于 x 轴上，而 y 轴表示特定状态的概率。图6.3(a)中的 y 轴被故意延长，以便与图6.3(b)中的 y 轴相同。

设 X 是一个在范围 $0.9 \leq X \leq 1.6$ 内取值的连续随机变量，如图6.3(b)所示。请注意，密度的高度可以大于1。但是，它必须满足

$$\int_{0.9}^{1.6} p(x)dx = 1.$$

(6.19)

Type	“Point probability”	“Interval probability”
Discrete	$P(X = x)$ Probability mass function	Not applicable
Continuous	$p(x)$ Probability density function	$P(X \leq x)$ Cumulative distribution function

表6.1：概率分布的命名法。

备注。关于离散概率分布，还有一个微妙的细节。状态 z_1, \dots, z_d 在原则上没有任何结构，即通常没有办法比较它们，例如 z_1 = 红色， z_2 = 绿色， z_3 = 蓝色。然而，在许多机器学习应用中，离散状态会取数值，例如 $z_1 = -1.1, z_2 = 0.3, z_3 = 1.5$ ，这时我们可以说 $z_1 < z_2 < z_3$ 。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，机器学习文献中使用的符号和术语掩盖了样本空间 Ω 、目标空间 \mathcal{T} 和随机变量 X 之间的区别。对于随机变量 X 的可能结果集中的一个值 x ，即 $x \in \mathcal{T}$ ， $p(x)$ 表示随机变量 X 具有结果 x 的概率。对于离散随机变量，这被写为 $P(X = x)$ ，这被

称为概率质量函数（PMF），PMF通常被称为“分布”。对于连续变量， $p(x)$ 被称为概率密度函数（通常简称为密度）。更进一步地，累积分布函数 $P(X \leq x)$ 也经常被称为“分布”。在本章中，我们将使用符号 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中总结了这些术语。

备注。我们将使用“概率分布”这一表达，不仅指离散概率质量函数，也指连续概率密度函数，尽管这在技术上是不正确的。与大多数机器学习文献一致，我们也依赖上下文来区分“概率分布”这一短语的不同用法。

< 上一章节

下一章节 >

6.1 概率空间的构建

6.3 加法规则、乘法规则与贝叶斯公式



6.3 加法规则、乘法规则与贝叶斯公式

我们将概率论视为逻辑推理的扩展。正如我们在第6.1.1节中讨论的那样，这里提出的概率规则自然而然地满足了所需条件（Jaynes, 2003, 第2章）。概率建模（第8.4节）为设计机器学习方法提供了原则性的基础。一旦我们定义了与数据和我们问题的不确定性相对应的概率分布（第6.2节），就会发现只有两个基本规则：加法规则和乘法规则。

回顾式 (6.9)， $p(\mathbf{x}, \mathbf{y})$ 是两个随机变量 \mathbf{x}, \mathbf{y} 的联合分布。分布 $p(\mathbf{x})$ 和 $p(\mathbf{y})$ 是相应的边缘分布，而 $p(\mathbf{y} | \mathbf{x})$ 是在给定 \mathbf{x} 的条件下 \mathbf{y} 的条件分布。根据第6.2节中离散和连续随机变量的边缘和条件概率的定义，我们现在可以介绍概率论中的两个基本规则。

第一个规则，加法规则，表明

$$p(\mathbf{x}) = \begin{cases} \sum_{\mathbf{y} \in \mathcal{Y}} p(\mathbf{x}, \mathbf{y}) & \text{如果 } \mathbf{y} \text{ 是离散的} \\ \int_{\mathcal{Y}} p(\mathbf{x}, \mathbf{y}) d\mathbf{y} & \text{如果 } \mathbf{y} \text{ 是连续的} \end{cases},$$

(6.20)

其中 \mathcal{Y} 是随机变量 \mathbf{Y} 的目标空间的状态。这意味着我们对随机变量 \mathbf{Y} 的状态集 \mathbf{y} 进行求和（或积分）。加法规则也被称为边缘化属性。加法规则将联合分布与边缘分布联系起来。一般来说，当联合分布包含两个以上的随机变量时，加法规则可以应用于随机变量的任何子集，从而得到可能包含多个随机变量的边缘分布。更具体地说，如果 $\mathbf{x} = [x_1, \dots, x_D]^\top$ ，我们通过反复应用加法规则（其中我们积分/求和除了 x_i 之外的所有随机变量，用 $\setminus i$ 表示“除了 i 之外的所有”），得到边缘分布

$$p(x_i) = \int p(x_1, \dots, x_D) d\mathbf{x}_{\setminus i}$$

(6.21)

备注。概率建模中的许多计算挑战都源于加法规则的应用。当存在许多变量或具有许多状态的离散变量时，加法规则归结为执行高维求和或积分。从计算的角度来看，执

行高维求和或积分通常是困难的，因为目前没有已知的多项式时间算法可以精确计算它们。

第二条规则，称为**乘法规则**，它通过以下方式将联合分布与条件分布联系起来：

(6.22)

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y} | \mathbf{x})p(\mathbf{x}).$$

乘法规则可以理解为，任何两个随机变量的联合分布都可以分解为（写成乘积形式）另外两个分布。这两个因子分别是第一个随机变量的边缘分布 $p(\mathbf{x})$ ，以及给定第一个随机变量时第二个随机变量的条件分布 $p(\mathbf{y} | \mathbf{x})$ 。由于在 $p(x, y)$ 中随机变量的顺序是任意的，乘法规则也意味着 $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x} | \mathbf{y})p(\mathbf{y})$ 。准确地说，(6.22) 是用离散随机变量的概率质量函数来表示的。对于连续随机变量，乘法规则则是用概率密度函数来表示的（第6.2.3节）。

在机器学习和贝叶斯统计中，我们经常在观察到其他随机变量的情况下，对未观察到的（潜在的）随机变量进行推断。假设我们对一个未观察到的随机变量 \mathbf{x} 有一些先验知识 $p(\mathbf{x})$ ，以及 \mathbf{x} 与我们可以观察到的第二个随机变量 \mathbf{y} 之间的某种关系 $p(\mathbf{y} | \mathbf{x})$ 。如果我们观察到了 \mathbf{y} ，我们可以使用贝叶斯定理根据观察到的 \mathbf{y} 的值来得出关于 \mathbf{x} 的一些结论。贝叶斯定理（也称为贝叶斯规则或贝叶斯定律）

(6.23)

$$\underbrace{p(\mathbf{x} | \mathbf{y})}_{\text{后验}} = \frac{\overbrace{p(\mathbf{y} | \mathbf{x})}^{\text{似然度}} \underbrace{p(\mathbf{x})}_{\text{先验}}}{\underbrace{p(\mathbf{y})}_{\text{证据}}}$$

是 (6.22) 中乘法规则的直接结果，因为

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x} | \mathbf{y})p(\mathbf{y})$$

(6.24)

以及

(6.25)

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y} | \mathbf{x})p(\mathbf{x})$$

所以



(6.26)

在 (6.23) 中, $p(x)$ 是先验, 它包含了我们在观察到任何数据之前对未观察到的 (潜在的) 变量 x 的主观先验知识。我们可以选择任何对我们有意义的先验, 但至关重要的是要确保先验在所有可能的 x 上都有非零的概率密度函数 (或概率质量函数), 即使它们非常罕见。

似然度 $p(y | x)$ 描述了 x 和 y 之间的关系, 在离散概率分布的情况下, 它是如果我们知道潜在变量 x , 则数据 y 出现的概率。请注意, 似然度有时并不被视为 x 上的分布, 而只是 y 上的分布 (MacKay, 2003)。

后验 $p(x | y)$ 是贝叶斯统计中我们感兴趣的量, 因为它准确地表达了我们所关心的内容, 即观察到 y 之后我们对 x 的了解。

(6.27) 式中的量

$$p(y) := \int p(y | x)p(x)dx = \mathbb{E}_X [p(y | x)]$$

是边缘似然/证据。式(6.27)的右侧使用了期望算子, 我们将在第6.4.1节中定义它。根据定义, 边缘似然是对(6.23)式的分子关于隐变量 x 的积分。因此, 边缘似然与 x 无关, 并且它确保了后验 $p(x | y)$ 是归一化的。边缘似然也可以被解释为在先验 $p(x)$ 下的期望似然。除了后验的归一化外, 边缘似然在贝叶斯模型选择中也起着重要作用, 我们将在第8.6节中讨论这一点。由于(8.44)式中的积分, 证据的计算通常很困难。

贝叶斯定理(6.23)允许我们反转由似然给出的 x 和 y 之间的关系。因此, 贝叶斯定理有时被称为概率逆定理。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, 因为它包含了先验和数据中的所有可用信息。除了考虑整个后验分布外, 还可以关注后验分布的一些统计量, 如后验最大值, 这将在第8.3节中讨论。然而, 关注后验分布的一些统计量会导致信息丢失。如果我们从更大的背景来考虑, 后验分布可以在决策系统中使用, 并且拥有完整的后验分布可能非常有用, 能够做出对抗具有鲁棒性的决策。例如, 在基于模型的强化学习背景下, Deisenroth et al. (2015) 表明, 使用可能转换函数的完整后验分布会导致非常快速 (数据/样本高效) 的学习, 而关注后验最大值则会导致持续的失败。

因此，对于下游任务而言，拥有完整的后验分布可能非常有用。在第9章中，我们将在线性回归的背景下继续这一讨论。

< 上一章节

下一章节 >

6.2 离散概率与连续概率

6.4 汇总统计量与独立性





6.4 汇總統計量与独立性

我们经常对随机变量集的总结和随机变量对的比较感兴趣。随机变量的统计量是该随机变量的确定性函数。分布的汇总统计量提供了一种有用的视角，来了解随机变量的行为，并且顾名思义，它提供了能够总结和描述分布的数值。我们描述了均值和方差，这两种广为人知的汇总统计量。然后，我们讨论了比较一对随机变量的两种方法：首先，如何判断两个随机变量是独立的；其次，如何计算它们之间的内积。

6.4.1 均值与协方差

均值和（协）方差通常用于描述概率分布的性质（期望值和离散程度）。我们将在第6.6节中看到，存在一类有用的分布族（称为指数族分布），其中随机变量的统计量捕获了所有可能的信息。

期望值的概念在机器学习中至关重要，概率论本身的基础概念也可以从期望值推导出来（Whittle, 2000）。

定义 6.3（期望值）：对于单变量连续随机变量 $X \sim p(x)$ 的函数 $g : \mathbb{R} \rightarrow \mathbb{R}$ ，其期望值定义为

(6.28)

$$\mathbb{E}_X[g(x)] = \int_{\mathcal{X}} g(x)p(x)dx$$

相应地，对于离散随机变量 $X \sim p(x)$ 的函数 g ，其期望值定义为

(6.29)

$$\mathbb{E}_X[g(x)] = \sum_{x \in \mathcal{X}} g(x)p(x)$$

其中， \mathcal{X} 是随机变量 X 所有可能结果（目标空间）的集合。

在本节中，我们认为离散随机变量的结果是数值型的。这可以通过观察函数 g 以实数作为输入来看出。

备注：我们将多元随机变量 \mathbf{X} 视为单变量随机变量 $[X_1, \dots, X_D]^\top$ 的有限向量。对于多元随机变量，我们逐元素地定义期望值

随机变量函数的期望值有时被称为“无意识统计学家定律”（Casella 和 Berger, 2002, 第2.2节）

(6.30)

$$\mathbb{E}_X[g(\mathbf{x})] = \begin{bmatrix} \mathbb{E}_{X_1}[g(x_1)] \\ \vdots \\ \mathbb{E}_{X_D}[g(x_D)] \end{bmatrix} \in \mathbb{R}^D$$

其中，下标 \mathbb{E}_{X_d} 表示我们正在对向量 \mathbf{x} 的第 d 个元素取期望值。

◇

定义 6.3 定义了符号 \mathbb{E}_X 的含义，作为指示我们应对概率密度（对于连续分布）或对所有状态求和（对于离散分布）取积分的算子。均值的定义（定义6.4）是期望值的一个特例，通过选择 g 为恒等函数获得。

定义 6.4（均值）：随机变量 \mathbf{X} ，其状态 $\mathbf{x} \in \mathbb{R}^D$ ，的均值是一个平均值，定义为

(6.31)

$$\mathbf{E}_X[\mathbf{x}] = \begin{bmatrix} \mathbb{E}_{X_1}[x_1] \\ \vdots \\ \mathbb{E}_{X_D}[x_D] \end{bmatrix} \in \mathbb{R}^D$$

其中

$$\mathbb{E}_{X_d}[x_d] := \begin{cases} \int_{\mathcal{X}} x_d p(x_d) dx_d & \text{如果 } \mathbf{X} \text{ 是连续随机变量} \\ \sum_{x_i \in \mathcal{X}} x_i p(x_d = x_i) & \text{如果 } \mathbf{X} \text{ 是离散随机变量} \end{cases}$$

(6.32)

对于 $d = 1, \dots, D$ ，下标 d 表示 \mathbf{x} 的相应维度。积分和求和是针对随机变量 \mathbf{X} 的目标空间状态 \mathcal{X} 进行的。

在一维情况下，还有另外两个直观的“平均”概念，即中位数和众数。中位数是排序后位于“中间”的值，即 50% 的值大于中位数，50% 的值小于中位数。这个概念可以通过考虑累积分布函数（定义6.2）为 0.5 时的值来推广到连续值。对于不对称或有长尾分

布，中位数提供了一个比均值更接近人类直觉的典型值估计。此外，中位数比均值更稳健。中位数向更高维度的推广并非易事，因为在一个以上的维度中没有明显的“排序”方式（Hallin et al., 2010; Kong and Mizera, 2012）。众数是出现频率最高的值。对于离散随机变量，众数定义为出现频率最高的 x 值。对于连续随机变量，众数定义为密度 $p(\mathbf{x})$ 的峰值。特定的密度 $p(\mathbf{x})$ 可能有一个以上的众数，并且在高维分布中可能存在大量的众数。因此，找到分布的所有众数在计算上可能具有挑战性。

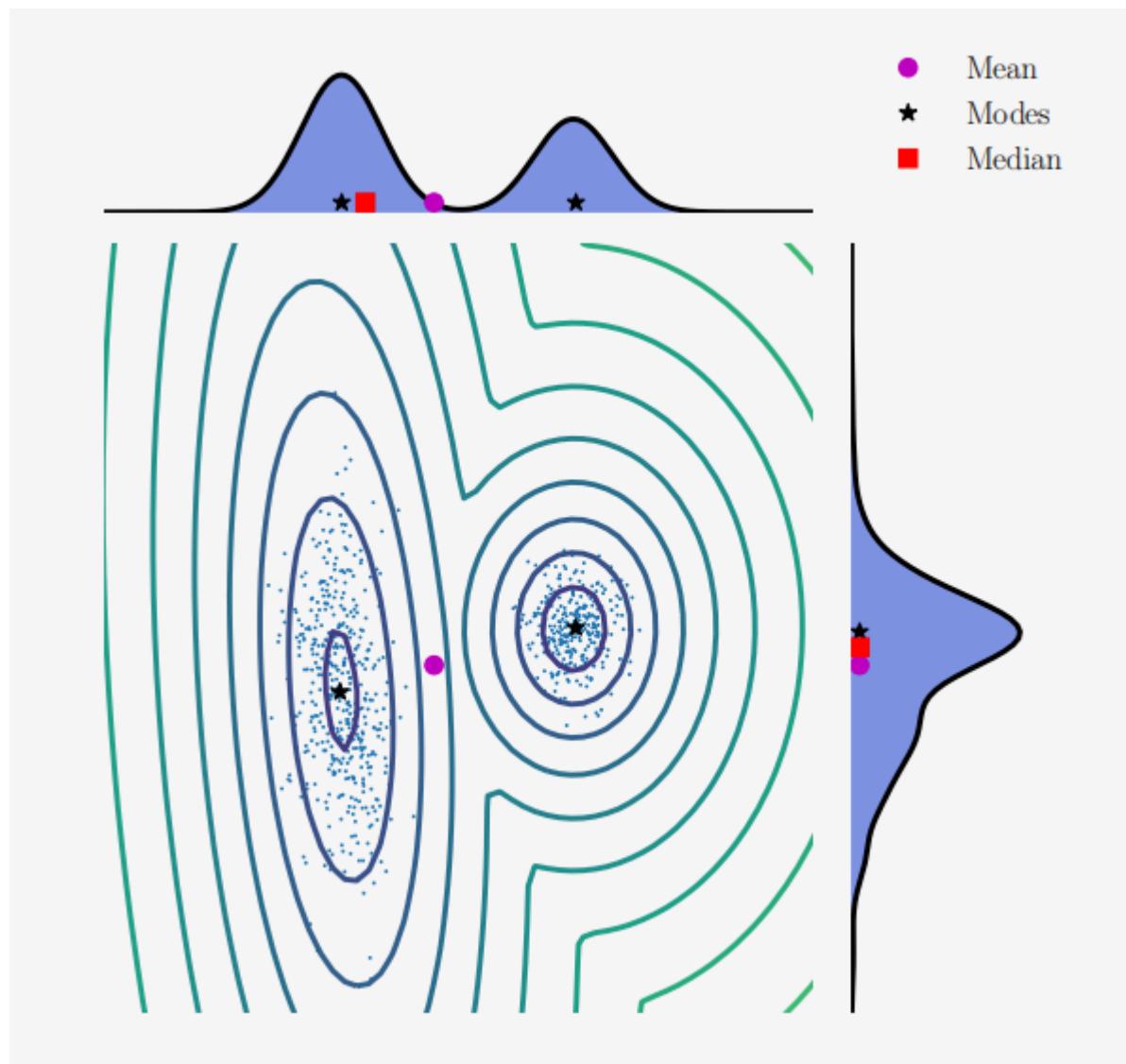
> 例 6.4

>

> 考虑图 6.4 中所示的二维分布：

>

>





>

>

图6.4一个二维数据集的平均值、模式和中位数及其边缘密度的说明。

>

>

$$p(\mathbf{x}) = 0.4\mathcal{N}\left(\mathbf{x} \mid \begin{bmatrix} 10 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) + 0.6\mathcal{N}\left(\mathbf{x} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 8.4 & 2.0 \\ 2.0 & 1.7 \end{bmatrix}\right).$$

>

> (6.33)

>

> 我们将在第 6.5 节中定义高斯分布 $\mathcal{N}(\mu, \sigma^2)$ 。同时，还展示了该分布在每个维度上的对应边缘分布。观察到该分布是双峰的（有两个众数），但其中一个边缘分布是单峰的（有一个众数）。水平方向上的双峰一元分布说明了均值和中位数可能彼此不同。尽管我们可能会想要将二维中位数定义为每个维度上中位数的串联，但由于我们无法定义二维点的顺序，这变得困难。当我们说“无法定义顺序”时，我们的意思是存在多种方式来定义关系 $<$ ，使得 $\begin{bmatrix} 3 \\ 0 \end{bmatrix} < \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ 这样的关系不是唯一的。

备注：期望值（定义 6.3）是一个线性算子。例如，给定一个实值函数 $f(\mathbf{x}) = ag(\mathbf{x}) + bh(\mathbf{x})$ ，其中 $a, b \in \mathbb{R}$ 且 $\mathbf{x} \in \mathbb{R}^D$ ，我们得到

(6.34a)

(6.34b)

$$\begin{aligned} \mathbb{E}_X[f(\mathbf{x})] &= \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \\ &= \int [ag(\mathbf{x}) + bh(\mathbf{x})]p(\mathbf{x})d\mathbf{x} \\ &= a \int g(\mathbf{x})p(\mathbf{x})d\mathbf{x} + b \int h(\mathbf{x})p(\mathbf{x})d\mathbf{x} \\ &= a \mathbb{E}_X[g(\mathbf{x})] + b \mathbb{E}_X[h(\mathbf{x})]. \end{aligned}$$

(6.34c)



(6.34d)

◇

对于两个随机变量，我们可能希望描述它们之间的对应关系。协方差直观地表示了随机变量之间依赖性的概念。

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各自偏离各自均值的乘积的期望值给出，即

(6.35)

$$\text{Cov}_{X,Y}[x, y] := \mathbb{E}_{X,Y}[(x - \mathbb{E}_X[x])(y - \mathbb{E}_Y[y])].$$

备注：当与期望值或多变量随机协方差相关的随机变量通过其参数明确时，下标通常会被省略（例如， $\mathbb{E}_X[x]$ 通常简写为 $\mathbb{E}[x]$ ）。

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

(6.36)

$$\text{Cov}[x, y] = \mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y].$$

变量与其自身的协方差 $\text{Cov}[x, x]$ 称为方差，记作 $\mathcal{V}_X[x]$ 。方差的平方根称为标准差，通常记作 $\sigma(x)$ 。协方差的概念可以推广到多变量随机变量。

定义 6.6 (协方差 (多变量))：如果我们考虑两个多变量随机变量 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

$$\text{Cov}[x, y] = \mathbb{E}[xy^\top] - \mathbb{E}[x]\mathbb{E}[y]^\top = \text{Cov}[y, x]^\top \in \mathbb{R}^{D \times E}.$$

(6.37)

定义 6.6 可以应用于两个参数中的相同多变量随机变量，这导致了一个有用的概念，它直观地捕获了随机变量的“散布”。对于多变量随机变量，方差描述了随机变量各个维度之间的关系。

定义 6.7 (方差)：随机变量 X 的方差，其状态为 $x \in \mathbb{R}^D$ ，均值向量为 $\mu \in \mathbb{R}^D$ ，定义为

(6.38a)

$$\begin{aligned}
 \mathbb{V}_X[\mathbf{x}] &= \text{Cov}_X[\mathbf{x}, \mathbf{x}] \\
 &= \mathbb{E}_X[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top] = \mathbb{E}_X[\mathbf{x}\mathbf{x}^\top] - \mathbb{E}_X[\mathbf{x}]\mathbb{E}_X[\mathbf{x}]^\top \\
 &= \begin{bmatrix} \text{Cov}[x_1, x_1] & \text{Cov}[x_1, x_2] & \dots & \text{Cov}[x_1, x_D] \\ \text{Cov}[x_2, x_1] & \text{Cov}[x_2, x_2] & \dots & \text{Cov}[x_2, x_D] \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}[x_D, x_1] & \dots & \dots & \text{Cov}[x_D, x_D] \end{bmatrix}.
 \end{aligned}$$

(6.38c)中的 $D \times D$ 矩阵被称为多元随机变量 \mathbf{X} 的协方差矩阵。协方差矩阵是对称的且是半正定的，它向我们揭示了数据的分布情况。在其对角线上，协方差矩阵包含了边缘分布的方差

(6.39)

$$p(x_i) = \int p(x_1, \dots, x_D) dx_{\setminus i},$$

其中“ $\setminus i$ ”表示“除了变量*i*之外的所有变量”。非对角线上的元素是 $i, j = 1, \dots, D, i \neq j$ 时的交叉协方差项 $\text{Cov}[x_i, x_j]$ 。

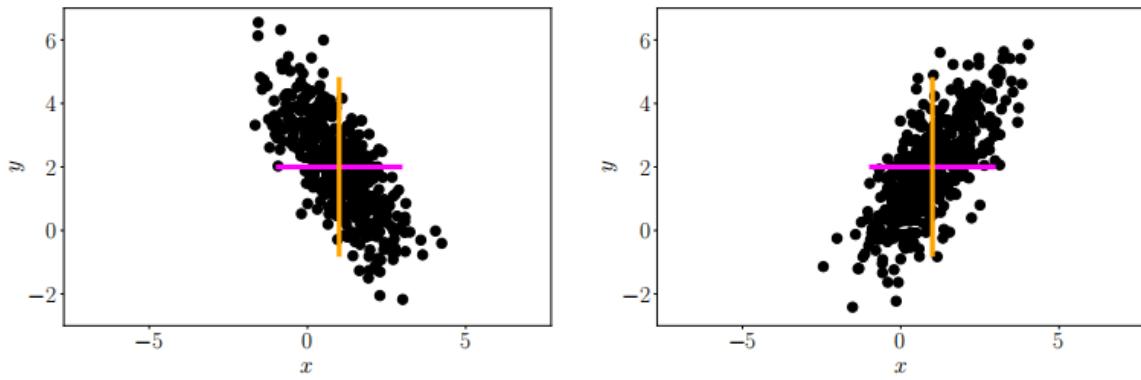


图6.5二维数据集沿每个轴（彩色线）具有相同的均值和方差，但具有不同的协方差。

备注。在本书中，我们通常假设协方差矩阵是正定的，以便更好地理解。因此，我们不讨论导致半正定（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都会影响协方差的值。协方差的归一化版本被称为相关系数。

定义6.8（相关系数） 两个随机变量 X, Y 之间的相关系数由

$$\text{corr}[x, y] = \frac{\text{Cov}[x, y]}{\sqrt{\text{V}[x]\text{V}[y]}} \in [-1, 1].$$

(6.40)

相关系数矩阵是标准化随机变量 $x/\sigma(x)$ 的协方差矩阵。换句话说，在相关系数矩阵中，每个随机变量都被其标准差（方差的平方根）除。

协方差（和相关系数）表明了两个随机变量之间的关系；见图6.5。正相关 $\text{corr}[x, y]$ 意味着当 x 增长时， y 也预期会增长。负相关则意味着当 x 增加时， y 会减小。

6.4.2 经验均值和协方差

第6.4.1节中的定义通常也被称为总体均值和总体协方差，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 。从总体统计量到经验统计量的实现，有两个概念上的步骤。首先，我们利用有限数据集（大小为 N ）来构造一个经验统计量，该统计量是有限数量相同随机变量 X_1, \dots, X_N 的函数。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），给定一个特定的数据集，我们可以获得均值的估计值，这被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值，定义为

$$\bar{\mathbf{x}} := \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n,$$

(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



$$\Sigma := \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^\top.$$

(6.42)

为了计算特定数据集的统计量，我们将使用实现（观测值） x_1, \dots, x_N ，并使用(6.41)和(6.42)。经验协方差矩阵是对称的、半正定的（见第3.2.3节）。

6.4.3 方差的三种表达式

我们现在专注于单一随机变量 X ，并使用前面的经验公式推导出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（定义6.5）的定义，是随机变量 X 与其期望值 μ 之间的平方偏差的期望值，即

$$\text{V}_X[x] := \mathbb{E}_X[(x - \mu)^2].$$

(6.43)

在(6.43)中的期望和均值 $\mu = \mathbb{E}_X(x)$ 的计算取决于 X 是离散还是连续随机变量，这通过(6.32)来完成。如(6.43)所示表达的方差是新随机变量 $Z := (X - \mu)^2$ 的均值。

在经验上估计(6.43)中的方差时，我们需要采用双遍算法：一遍遍历数据以使用(6.41)计算均值 μ ，然后第二遍使用此估计值 $\hat{\mu}$ 来计算方差。通过重新排列项，我们可以避免双遍遍历。可以将(6.43)中的公式转换为所谓的方差原始分数公式：

(6.44)

$$\text{V}_X[x] = \mathbb{E}_X[x^2] - (\mathbb{E}_X[x])^2.$$

(6.44)中的表达式可以记忆为“平方的均值减去均值的平方”。我们可以在一遍遍历数据的过程中通过同时累积 \mathbf{x}_i （以计算均值）和 \mathbf{x}_i^2 来经验地计算它，其中 \mathbf{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。当推导等权偏差-方差分解（Bishop, 2006）时，(6.44)中的原始分数版本对于机器学习可能是有用的。

理解方差的第三种方式是，它是所有观测对之间的成对差异之和。考虑随机变量 X 的实现的一个样本 x_1, \dots, x_N ，我们计算每对 x_i 和 x_j 之间的平方差。通过展开平方，我们可以证明 N^2 个成对差异的总和是观测值的经验方差：

(6.45)

$$\frac{1}{N^2} \sum_{i,j=1}^N (x_i - x_j)^2 = 2 \left[\frac{1}{N} \sum_{i=1}^N x_i^2 - \left(\frac{1}{N} \sum_{i=1}^N x_i \right)^2 \right].$$

我们看到(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总和（共有 N^2 个）表示为从均值（共有 N 个）的偏差之和。从几何角度来看，这意味着点集中心与点对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（再次是求和中的 N 项），我们可以得到一个具有 N^2 项的表达式（即(6.45)的左侧）。

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

(6.53)

$$p(x, y) = p(x)p(y).$$

直观上，如果两个随机变量 X 和 Y 是独立的，那么知道 y 的值并不会给 x 提供任何额外的信息（反之亦然）。如果 X, Y 是（统计）独立的，那么

- $p(y | x) = p(y)$

- $p(x | y) = p(x)$

$$\text{V}_{X,Y}[x + y] = \text{V}_X[x] + \text{V}_Y[y]$$

$$\cdot \text{Cov}_{X,Y}[x, y] = 0$$

最后一点可能不总是成立的逆命题，即两个随机变量可以有协方差为零但并非统计独立。为了理解这一点，需要回顾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**

> 考虑一个均值为零的随机变量 X ($\mathbb{E}_X[x] = 0$) 且

>

> $\mathbb{E}_X[x^3] = 0$ 。令 $y = x^2$ (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

>

>

$$\text{Cov}[x, y] = \mathbb{E}[xy] - \mathbb{E}[x]\mathbb{E}[y] = \mathbb{E}[x^3] = 0.$$

> (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, 即 X_1, \dots, X_N 是独立且同分布的。对于超过两个随机变量的情况, 如果所有子集都是独立的 (参见 Pollard (2002, 第4章) 和 Jacod and Protter (2004, 第3章))。“同分布”意味着所有随机变量都来自同一分布。

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 的条件下是条件独立的当且仅当

(6.55)

$$p(\mathbf{x}, \mathbf{y} \mid \mathbf{z}) = p(\mathbf{x} \mid \mathbf{z})p(\mathbf{y} \mid \mathbf{z}) \quad \text{for all } \mathbf{z} \in \mathcal{Z},$$

其中, \mathcal{Z} 是随机变量 Z 的状态集。我们用 $X \perp Y \mid Z$ 来表示给定 Z 时, X 与 Y 是条件独立的。

定义6.11 要求 (6.55) 中的关系必须对 z 的每一个值都成立。(6.55) 的解释可以理解为“在知道 z 的情况下, x 和 y 的分布是可分解的”。如果我们写 $X \perp Y \mid \emptyset$, 则独立性可以视为条件独立性的一个特例。通过使用概率的乘积规则 (6.22), 我们可以展开 (6.55) 的左侧得到

(6.56)

$$p(\mathbf{x}, \mathbf{y} \mid \mathbf{z}) = p(\mathbf{x} \mid \mathbf{y}, \mathbf{z})p(\mathbf{y} \mid \mathbf{z}).$$

通过比较 (6.55) 的右侧与 (6.56), 我们发现 $p(y \mid z)$ 同时出现在两者中, 因此

(6.57)

$$p(\mathbf{x} \mid \mathbf{y}, \mathbf{z}) = p(\mathbf{x} \mid \mathbf{z}).$$

方程 (6.57) 提供了条件独立性的另一种定义，即 $X \amalg Y \mid Z$ 。这种替代表述提供了这样的解释：“在知道 z 的情况下，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。我们可以在随机变量之间定义内积，并在本节中简要描述。如果我们有两个不相关的随机变量 X, Y ，则

多变量随机变量可以

(6.58) 此处原文似乎有误或遗漏，但基于上下文，我们可以理解为讨论的是不相关随机变量方差的可加性，即：

$$\text{V}[X + Y] = \text{V}[X] + \text{V}[Y].$$

由于方差是以平方单位衡量的，这看起来非常像直角三角形中的勾股定理 $c^2 = a^2 + b^2$ 。接下来，我们探讨是否能为 (6.58) 中不相关随机变量的方差关系找到几何解释。

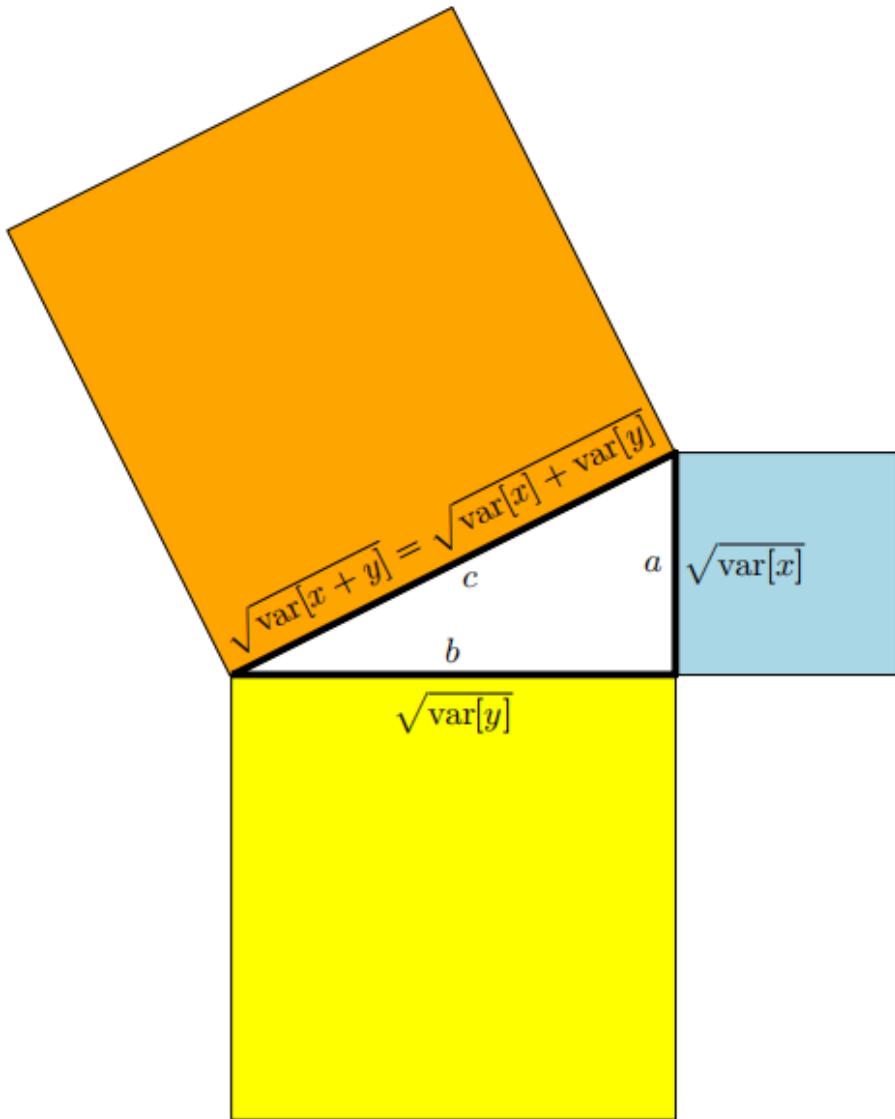


图6.6随机变量的几何形状。如果随机变量 X 和 Y 不相关，则它们是相应线性空间中的正交向量，并应用毕达哥拉斯定理。

随机变量可以视为线性空间中的向量，我们可以定义内积以获得随机变量的几何性质 (Eaton, 2007)。如果我们定义

(6.59)

$$\langle X, Y \rangle := \text{Cov}[X, Y]$$

对于均值为零的随机变量 X 和 Y ，我们得到了一个内积。可以看出，协方差是对称的、正定的，并且在任一参数上都是线性的。随机变量的“长度”是

$$\|X\| = \sqrt{\text{Cov}[X, X]} = \sqrt{\text{V}[X]} = \sigma[X],$$

(6.60)

即其标准差。随机变量“越长”，其不确定性就越大；长度为0的随机变量是确定的。

如果我们查看两个随机变量 X, Y 之间的角度 θ ，我们得到

(6.61)

$$\cos \theta = \frac{\langle X, Y \rangle}{\|X\| \|Y\|} = \frac{\text{Cov}[X, Y]}{\sqrt{\text{V}[X]\text{V}[Y]}},$$

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，可以将相关性视为两个随机变量之间角度的余弦值。根据定义3.7，我们知道 $X \perp Y \iff \langle X, Y \rangle = 0$ 。在我们的情况下，这意味着 X 和 Y 是正交的当且仅当 $\text{Cov}[X, Y] = 0$ ，即它们是不相关的。图6.6说明了这种关系。

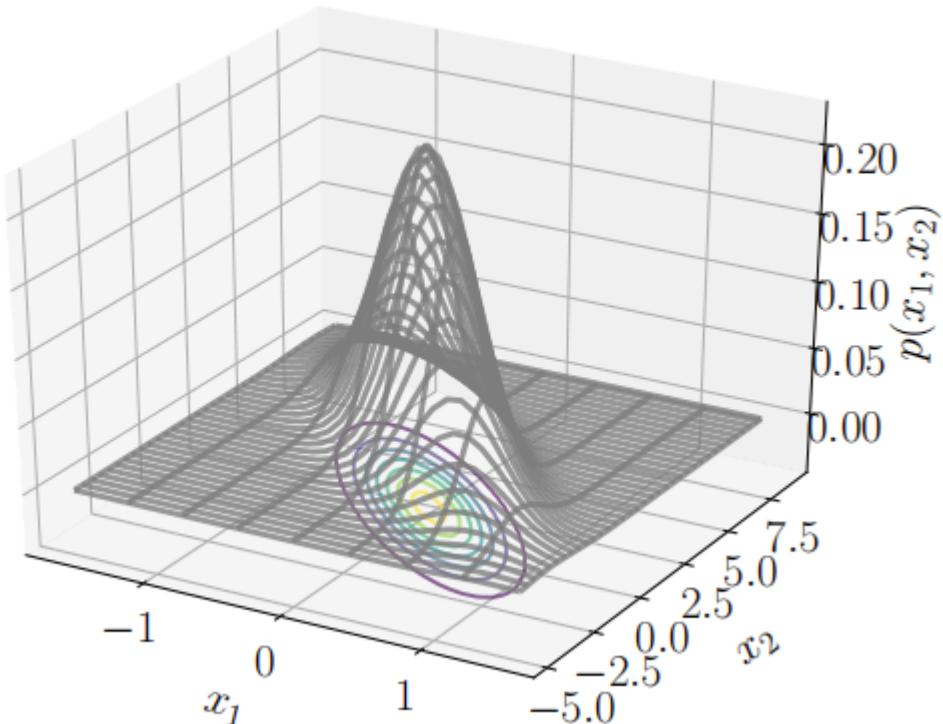


图6.7两个随机变量 x_1 和 x_2 的高斯分布。

备注：虽然使用基于前面定义的内积构造的 Euclid 距离来比较概率分布很诱人，但遗憾的是，这并不是获得分布之间距离的最佳方式。回想一下，概率质量（或密度）是正的，并且需要加起来等于1。这些约束意味着分布存在于所谓的统计流形上。对这个概率分布空间的研究称为信息几何。计算分布之间的距离通常使用Kullback-Leibler散度，它是考虑统计流形性质的距离的一种推广。就像 Euclid 距离是度量（第3.3节）的一个特例一样，Kullback-Leibler散度也是称为Bregman散度和 f -散度

的两种更一般散度类的特例。对散度的研究超出了本书的范围，我们建议查阅信息论领域创始人之一的Amari (2016) 的近期著作，以获取更多详细信息。



< 上一章节

下一章节 >

6.3 加法规则、乘法规则与贝叶斯公式

6.5 高斯分布



6.5 高斯分布

高斯分布是所有连续型随机变量的概率分布中被研究的最透彻的一种分布。它也被叫做正态分布。它的最重要性事实上来源于一些便于计算的性质，我们将在下面进行讨论。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），并考虑密度估计的高斯混合数（第11章）。

机器学习的许多其他领域也受益于使用高斯分布，例如高斯过程、变分推理和强化学习。它也被广泛应用于其他应用领域，如信号处理（如卡尔曼滤波器）、控制（如线性二次调节器）和统计（如假设检验）。

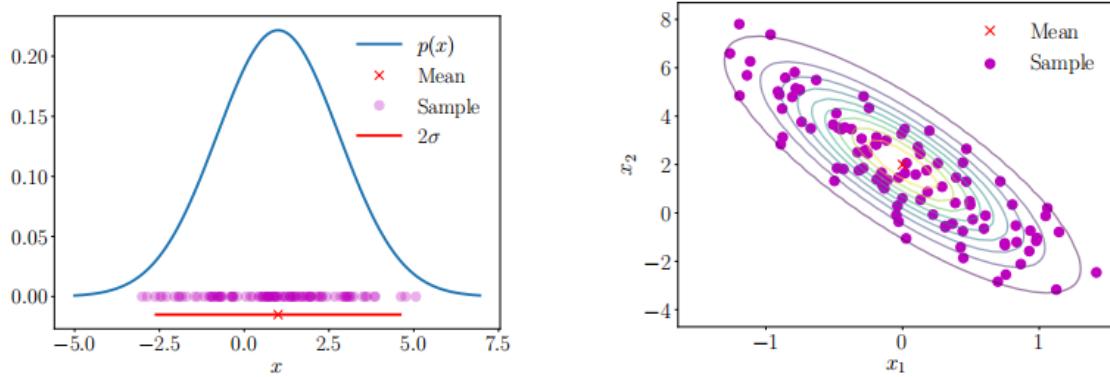


图6.8高斯分布覆盖了100个样本。(a)一维情况; (b)二维情况。

对于单变量随机变量，高斯分布（Gaussian distribution）的密度函数由下式给出：

(6.62)

$$p(x | \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right).$$

多元高斯分布（Multivariate Gaussian distribution）完全由均值向量 μ 和协方差矩阵 Σ 描述，并定义为：

(6.63)

$$p(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{D}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right),$$

其中， $\mathbf{x} \in \mathbb{R}^D$ 。我们通常写作 $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ 或 $X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ 。图6.7展示了二元高斯分布（网格图）及其对应的等高线图。图6.8展示了单变量高斯分布和二元高斯分布及其对应的样本。当高斯分布的均值为零且协方差为单位矩阵时，即 $\boldsymbol{\mu} = \mathbf{0}$ 且 $\boldsymbol{\Sigma} = \mathbf{I}$ ，这种情况被称为标准正态分布（standard normal distribution）。

高斯分布在统计估计和机器学习领域中被广泛使用，因为它们具有边际分布和条件分布的闭式表达式。在第九章中，我们将这些闭式表达式广泛用于线性回归。使用高斯随机变量进行建模的一个主要优势是通常不需要进行变量变换（第6.7节）。由于高斯分布完全由其均值和协方差指定，因此我们通常可以通过对随机变量的均值和协方差进行变换来获得变换后的分布。

6.5.1 高斯分布的边际分布和条件分布仍然是高斯分布

以下，我们介绍在多元随机变量的一般情况下的边际化和条件化。如果初次阅读时感到困惑，建议读者先考虑两个单变量随机变量的情况。设 \mathbf{X} 和 \mathbf{Y} 是两个可能具有不同维度的多元随机变量。为了考虑应用概率和规则和条件化的影响，我们明确地将高斯分布表示为连接状态 $[\mathbf{x}^\top, \mathbf{y}^\top]$ 的函数，

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}\right). \quad (6.64)$$

其中， $\boldsymbol{\Sigma}_{xx} = \text{Cov}[\mathbf{x}, \mathbf{x}]$ 和 $\boldsymbol{\Sigma}_{yy} = \text{Cov}[\mathbf{y}, \mathbf{y}]$ 分别是 \mathbf{x} 和 \mathbf{y} 的边际协方差矩阵，而 $\boldsymbol{\Sigma}_{xy} = \text{Cov}[\mathbf{x}, \mathbf{y}]$ 是 \mathbf{x} 和 \mathbf{y} 之间的互协方差矩阵。

条件分布 $p(\mathbf{x} \mid \mathbf{y})$ 也是高斯分布（如图6.9(c)所示），并由（Bishop, 2006的2.3节推导得出）

(6.65)

$$\begin{aligned} p(\mathbf{x} \mid \mathbf{y}) &= \mathcal{N}(\boldsymbol{\mu}_{x|y}, \boldsymbol{\Sigma}_{x|y}) \\ \boldsymbol{\mu}_{x|y} &= \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} (\mathbf{y} - \boldsymbol{\mu}_y) \\ \boldsymbol{\Sigma}_{x|y} &= \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\Sigma}_{yx}. \end{aligned} \quad (6.66)$$

(6.67)



注意，在计算(6.66)中的均值时， y 的值是一个观测值，不再是随机的。

备注。条件高斯分布在许多地方都会出现，特别是在我们对后验分布感兴趣的情况下：

- 卡尔曼滤波器（Kalman, 1960），是信号处理中状态估计最核心的算法之一，其本质就是计算联合分布的高斯条件分布（Deisenroth和Ohlsson, 2011; Särkkä, 2013）。
- 高斯过程（Rasmussen和Williams, 2006），是函数分布的一种实用实现。在高斯过程中，我们对随机变量的联合高斯性做出假设。通过对观测数据进行（高斯）条件化，我们可以确定函数的后验分布。
- 潜在线性高斯模型（Roweis和Ghahramani, 1999; Murphy, 2012），包括概率主成分分析（PPCA）（Tipping和Bishop, 1999）。我们将在第10.7节中更详细地讨论PPCA。

联合高斯分布 $p(\mathbf{x}, \mathbf{y})$ （见(6.64)）的边际分布 $p(\mathbf{x})$ 本身也是高斯分布，通过应用求和规则(6.20)计算得出，具体为

(6.68)

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y} = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx}).$$

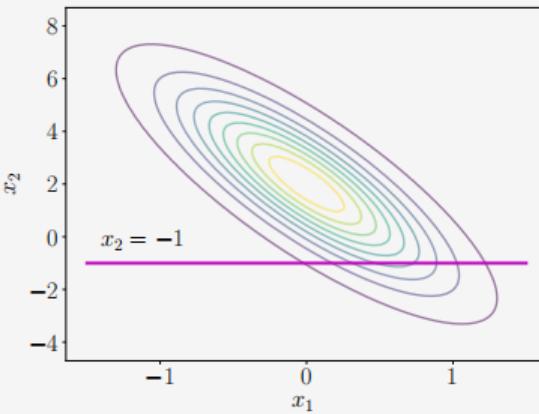
对于 $p(\mathbf{y})$ 也有相应的结果，它是通过对 \mathbf{x} 进行边际化得到的。直观上看，在观察(6.64)中的联合分布时，我们忽略了（即，积分掉）我们不感兴趣的所有内容。这如图6.9(b)所示。

> **例6.6**

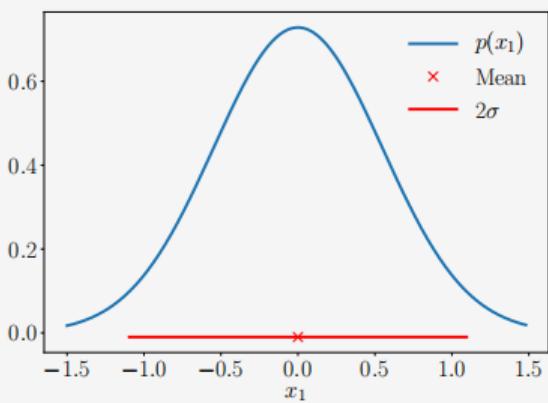
>



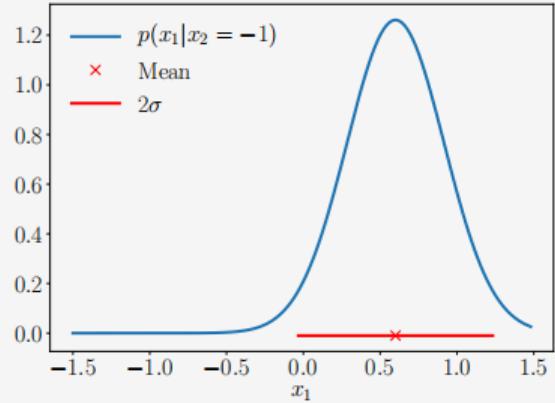
>



(a) Bivariate Gaussian.



(b) Marginal distribution.



(c) Conditional distribution.

>

>

图6.9 (a)二元高斯分布；联合高斯分布的(b)边缘是高斯分布；(c)高斯分布的条件分布也是高斯分布。

>

> 考虑二元高斯分布（如图6.9所示）：

>

> (6.69)

>

$$p(x_1, x_2) = \mathcal{N} \left(\begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 0.3 & -1 \\ -1 & 5 \end{bmatrix} \right).$$



>

> 我们可以通过应用(6.66)和(6.67)来计算在 $x_2 = -1$ 条件下，单变量高斯分布的参数，从而分别得到均值和方差。数值上，这等于

>

>

$$\mu_{x_1|x_2=-1} = 0 + (-1) \cdot 0.2 \cdot (-1 - 2) = 0.6$$

> (6.70)

>

> 以及

>

>

$$\sigma_{x_1|x_2=-1}^2 = 0.3 - (-1) \cdot 0.2 \cdot (-1) = 0.1 .$$

> (6.71)

>

> 因此，条件高斯分布由下式给出：

>

>

$$p(x_1 | x_2 = -1) = \mathcal{N}(0.6, 0.1) .$$

> (6.72)

>

> 相比之下，边际分布 $p(x_1)$ 可以通过应用(6.68)获得，这实质上就是使用随机变量 x_1 的均值和方差，得到

>

>



$$p(x_1) = \mathcal{N}(0, 0.3).$$

> (6.73)

6.5.2 高斯密度的乘积

在线性回归（第9章）中，我们需要计算高斯似然函数。此外，我们可能还希望假设一个高斯先验（第9.3节）。我们应用贝叶斯定理来计算后验分布，这涉及到似然函数和先验分布的乘积，即两个高斯密度的乘积。两个高斯分布的乘积 $\mathcal{N}(x | \mathbf{a}, \mathbf{A})\mathcal{N}(x | \mathbf{b}, \mathbf{B})$ 的推导结果是

一个由实数 $c \in \mathbb{R}$ 缩放的高斯分布，表示为 $c\mathcal{N}(x | c, C)$ ，其中相关的练习在

(6.74) (6.75)

$$\begin{aligned} C &= (A^{-1} + B^{-1})^{-1} \\ c &= C(A^{-1}\mathbf{a} + B^{-1}\mathbf{b}) \\ c &= (2\pi)^{-\frac{D}{2}} |A + B|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{a} - \mathbf{b})^\top (A + B)^{-1}(\mathbf{a} - \mathbf{b})\right). \end{aligned}$$

(6.76)

缩放常数 c 本身可以写成 \mathbf{a} 或 \mathbf{b} 的高斯密度形式，但协方差矩阵为“膨胀”的 $A + B$ ，即 $c = \mathcal{N}(\mathbf{a} | \mathbf{b}, A + B) = \mathcal{N}(\mathbf{b} | \mathbf{a}, A + B)$ 。

备注。为了方便表示，我们有时会用 $\mathcal{N}(x | \mathbf{m}, \mathbf{S})$ 来描述高斯密度的函数形式，即使 x 不是随机变量。在前面的演示中，我们就是这样做的，当时我们写了

(6.77)

$$c = \mathcal{N}(\mathbf{a} | \mathbf{b}, \mathbf{A} + \mathbf{B}) = \mathcal{N}(\mathbf{b} | \mathbf{a}, \mathbf{A} + \mathbf{B}).$$

这里， \mathbf{a} 和 \mathbf{b} 都不是随机变量。然而，将 c 写成这种形式比 (6.76) 更简洁。

6.5.3 和与线性变换



如果 X, Y 是独立的高斯随机变量（即，联合分布为 $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$ ），其中 $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ 且 $p(\mathbf{y}) = \mathcal{N}(\mathbf{y} | \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$ ，那么 $x + y$ 也是高斯分布的，并且由下式给出

$$p(\mathbf{x} + \mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_x + \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_y).$$

(6.78)

知道 $p(x + y)$ 是高斯分布的，我们可以立即使用从 (6.46) 到 (6.49) 的结果来确定均值和协方差矩阵。这一性质在我们考虑独立同分布（i.i.d.）高斯噪声作用于随机变量时非常重要，这在线性回归（第9章）中就是这样的情况。

> 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的概率分布。对于 $a\mathbf{x} + b\mathbf{y}$ ，其概率分布为

>

>

$$p(a\mathbf{x} + b\mathbf{y}) = \mathcal{N}(a\boldsymbol{\mu}_x + b\boldsymbol{\mu}_y, a^2\boldsymbol{\Sigma}_x + b^2\boldsymbol{\Sigma}_y).$$

> (6.79)

>

> 这里， a 和 b 是常数， \mathbf{x} 和 \mathbf{y} 是独立的高斯随机变量， $\boldsymbol{\mu}_x$ 和 $\boldsymbol{\mu}_y$ 分别是 \mathbf{x} 和 \mathbf{y} 的均值向量， $\boldsymbol{\Sigma}_x$ 和 $\boldsymbol{\Sigma}_y$ 分别是 \mathbf{x} 和 \mathbf{y} 的协方差矩阵。这个结果是基于高斯分布的线性变换性质得出的。

备注。在第11章中，高斯密度（Gaussian densities）的加权和将非常有用。这与高斯随机变量（Gaussian random variables）的加权和是不同的。

◆

在定理6.12中，随机变量 \mathbf{x} 来自一个由两个密度 $p_1(\mathbf{x})$ 和 $p_2(\mathbf{x})$ 按 α 加权混合而成的密度。该定理可以推广到多元随机变量的情况，因为期望的线性性质对于多元随机变量也同样成立。然而，平方随机变量的概念需要被 $\mathbf{x}\mathbf{x}^\top$ 所替代。

定理6.12. 考虑两个一元高斯密度的混合



(6.80)

$$p(x) = \alpha p_1(x) + (1 - \alpha)p_2(x),$$

其中标量 $0 < \alpha < 1$ 是混合权重, $p_1(x)$ 和 $p_2(x)$ 是具有不同参数的一元高斯密度 (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:

(6.81)

$$\mathbb{E}[x] = \alpha\mu_1 + (1 - \alpha)\mu_2.$$

混合密度 $p(x)$ 的方差由

$$\mathbb{V}[x] = [\alpha\sigma_1^2 + (1 - \alpha)\sigma_2^2] + \left([\alpha\mu_1^2 + (1 - \alpha)\mu_2^2] - [\alpha\mu_1 + (1 - \alpha)\mu_2]^2 \right)$$

给出。

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) , 并将我们的混合公式 (6.80) 代入, 得到

(6.83a)

(6.83b)

$$\begin{aligned} \mathbf{E}[x] &= \int_{-\infty}^{\infty} xp(x)dx \\ &= \int_{-\infty}^{\infty} (\alpha x p_1(x) + (1 - \alpha)x p_2(x)) dx \\ &= \alpha \int_{-\infty}^{\infty} x p_1(x) dx + (1 - \alpha) \int_{-\infty}^{\infty} x p_2(x) dx \\ &= \alpha\mu_1 + (1 - \alpha)\mu_2. \end{aligned}$$

(6.83c)

(6.83d)

为了计算方差, 我们可以使用 (6.44) 中的方差原始分数版本, 这需要平方随机变量的期望的表达式。在这里, 我们使用随机变量函数 (平方) 的期望的定义 (定义

6.3) ,



(6.84a)

$$\begin{aligned}\mathbb{E}[x^2] &= \int_{-\infty}^{\infty} x^2 p(x) dx \\ &= \int_{-\infty}^{\infty} (\alpha x^2 p_1(x) + (1 - \alpha)x^2 p_2(x)) dx\end{aligned}$$

(6.84b)

(6.84c)

$$\begin{aligned}&= \alpha \int_{-\infty}^{\infty} x^2 p_1(x) dx + (1 - \alpha) \int_{-\infty}^{\infty} x^2 p_2(x) dx \\ &= \alpha(\mu_1^2 + \sigma_1^2) + (1 - \alpha)(\mu_2^2 + \sigma_2^2),\end{aligned}$$

(6.84d)

在最后这个等式中，我们再次使用了方差的原始分数版本 (6.44)，即 $\sigma^2 = \mathbb{E}[x^2] - \mu^2$ 。这个等式被重新排列，使得平方随机变量的期望是均值的平方和方差的和。

因此，方差是通过从(6.84d)中减去(6.83d)来给出的，

(6.85a) (6.85b)

$$\begin{aligned}\mathbb{V}[x] &= \mathbb{E}[x^2] - (\mathbb{E}[x])^2 \\ &= \alpha(\mu_1^2 + \sigma_1^2) + (1 - \alpha)(\mu_2^2 + \sigma_2^2) - (\alpha\mu_1 + (1 - \alpha)\mu_2)^2 \\ &= [\alpha\sigma_1^2 + (1 - \alpha)\sigma_2^2] \\ &\quad + \left([\alpha\mu_1^2 + (1 - \alpha)\mu_2^2] - [\alpha\mu_1 + (1 - \alpha)\mu_2]^2 \right).\end{aligned}$$

(6.85c)

备注。前面的推导适用于任何密度，但由于高斯分布完全由均值和方差确定，因此混合密度可以以封闭形式确定。

◆

对于混合密度，各个组成部分可以被认为是条件分布（以组件身份为条件）。方程 (6.85c) 是条件方差公式的一个例子，也称为全方差定律，它一般地表明对于两个随

机变量 X 和 Y , 有 $\text{V}_X[x] = \mathbf{E}_Y[\text{V}_X[x|y]] + \text{V}_Y[\mathbf{E}_X[x|y]]$, 即 X 的 (总) 方差是条件方差的期望值加上条件均值的方差。

在示例6.17中, 我们考虑了一个二元标准高斯随机变量 X , 并对其进行了线性变换 Ax 。结果是一个均值为零、协方差为 AA^\top 的高斯随机变量。请注意, 添加一个常数向量会改变分布的均值, 但不会影响其方差, 即随机变量 $x + \mu$ 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 $X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ 。对于给定形状的矩阵 A 的变换, 设 Y 是一个随机变量, 使得 $y = Ax$ 是 x 的变换版本。我们可以利用期望是线性运算 (6.50) 来计算高斯 y 的均值, 如下所示:

$$\mathbf{E}[y] = \mathbf{E}[Ax] = A\mathbf{E}[x] = A\boldsymbol{\mu}.$$

(6.86)

类似地, 我们可以使用 (6.51) 来找到 y 的方差:

$$\text{V}[y] = \text{V}[Ax] = A\text{V}[x]A^\top = A\Sigma A^\top.$$

(6.87)

这意味着随机变量 y 的分布是

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} | A\boldsymbol{\mu}, A\Sigma A^\top).$$

(6.88)

现在让我们考虑反向变换: 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 $y \in \mathbb{R}^M$ 是一个均值为 Ax 的高斯随机变量, 即,

(6.89)

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} | Ax, \boldsymbol{\Sigma}).$$

那么对应的概率分布 $p(x)$ 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。但是, 一般来说 A 是不可逆的, 我们使用与伪逆 (3.57) 类似的方法。即, 我们先对两边同时左乘 A^\top , 然后求 $A^\top A$ 的逆, 它是对称且正定的, 从而得到关系



$$y = Ax \iff (A^\top A)^{-1} A^\top y = x.$$

(6.90)

因此， x 是 y 的线性变换，我们得到

(6.91)

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | (A^\top A)^{-1} A^\top \mathbf{y}, (A^\top A)^{-1} A^\top \Sigma A (A^\top A)^{-1}).$$

6.5.4 从多元高斯分布中采样

我们不会详细解释计算机上随机采样的微妙之处，感兴趣的读者可以参考Gentle (2004) 的著作。在多元高斯分布的情况下，该过程包含三个阶段：首先，我们需要一个伪随机数源，它能在区间[0,1]内提供均匀样本；其次，我们使用非线性变换，如Box-Müller变换（Devroye, 1986），从一元高斯分布中获取样本；第三，我们将这些样本向量组合起来，以获得来自多元标准正态分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 的样本。

对于一般的多元高斯分布，即均值非零且协方差不是单位矩阵的情况，我们使用高斯随机变量的线性变换的性质。假设我们想要从均值为 μ 、协方差矩阵为 Σ 的多元高斯分布中生成样本 $\mathbf{x}_i, i = 1, \dots, n$ 。我们希望从一个能为多元标准正态分布 $\mathcal{N}(\mathbf{0}, \bar{\mathbf{I}})$ 提供样本的采样器中构造样本。

为了从多元正态分布 $\mathcal{N}(\mu, \Sigma)$ 中获取样本，我们可以利用高斯随机变量的线性变换的性质：如果 $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ，那么 $\mathbf{y} = A\mathbf{x} + \mu$ ，其中 $A A^\top = \Sigma$ ，且 \mathbf{y} 是具有均值 μ 和协方差矩阵 Σ 的高斯分布。选择 A 的一个方便方法是使用协方差矩阵 $\Sigma = A A^\top$ 的Cholesky分解（第4.3节）。Cholesky分解的优点是 A 是三角矩阵，这有助于提高计算效率。

< 上一章节

下一章节 >

6.4 汇总计量与独立性

6.6 共轭性与指数族分布



6.6 共轭性与指数族分布

我们在统计学教科书中遇到的许多“有名”的概率分布都是为了模拟特定类型的现象而发现的。例如，我们在6.5节中见到了高斯分布。这些分布之间也以复杂的方式相互关联（Leemis 和 McQueston, 2008）。对于该领域的初学者来说，要弄清楚应该使用哪种分布可能会感到不知所措。此外，许多这些分布的发现时期，统计和计算还只能通过笔和纸来完成。因此，很自然地会提出这样的问题：在计算时代（Efron 和 Hastie, 2016），哪些概念是职位描述中有意义的？

在上一节中，我们看到，当分布是高斯分布时，许多用于推断的操作都可以方便地计算。此时，值得回顾在机器学习上下文中操作概率分布的期望属性：

1. 在应用概率规则时，存在一些“封闭性”，例如贝叶斯定理。这里的封闭性意味着应用特定操作会返回同类型的对象。
2. 当我们收集更多数据时，不需要更多参数来描述分布。
3. 由于我们感兴趣的是从数据中学习，因此我们希望参数估计能够表现良好。

事实证明，被称为指数族的分布类在保持一般性的同时，也保留了有利的计算和推断特性。在介绍指数族之前，让我们再来看三个“有名”的概率分布成员：伯努利分布（示例6.8）、二项分布（示例6.9）和贝塔分布（示例6.10）。

例 6.8

伯努利分布是针对单个二元随机变量 X 的分布，其状态 $x \in \{0, 1\}$ 。它由一个连续参数 $\mu \in [0, 1]$ 控制，该参数表示 $X = 1$ 的概率。伯努利分布 $\text{Ber}(\mu)$ 定义为

(6.92)

$$\begin{aligned} p(x | \mu) &= \mu^x (1 - \mu)^{1-x}, \quad x \in \{0, 1\}, \\ \mathbf{E}[x] &= \mu, \\ \mathbf{V}[x] &= \mu(1 - \mu), \end{aligned}$$

其中， $\mathbf{E}[x]$ 和 $\mathbf{V}[x]$ 分别是二元随机变量 X 的均值和方差。

可以使用伯努利分布的一个例子是，当我们对抛硬币时“头”的概率建模感兴趣时。

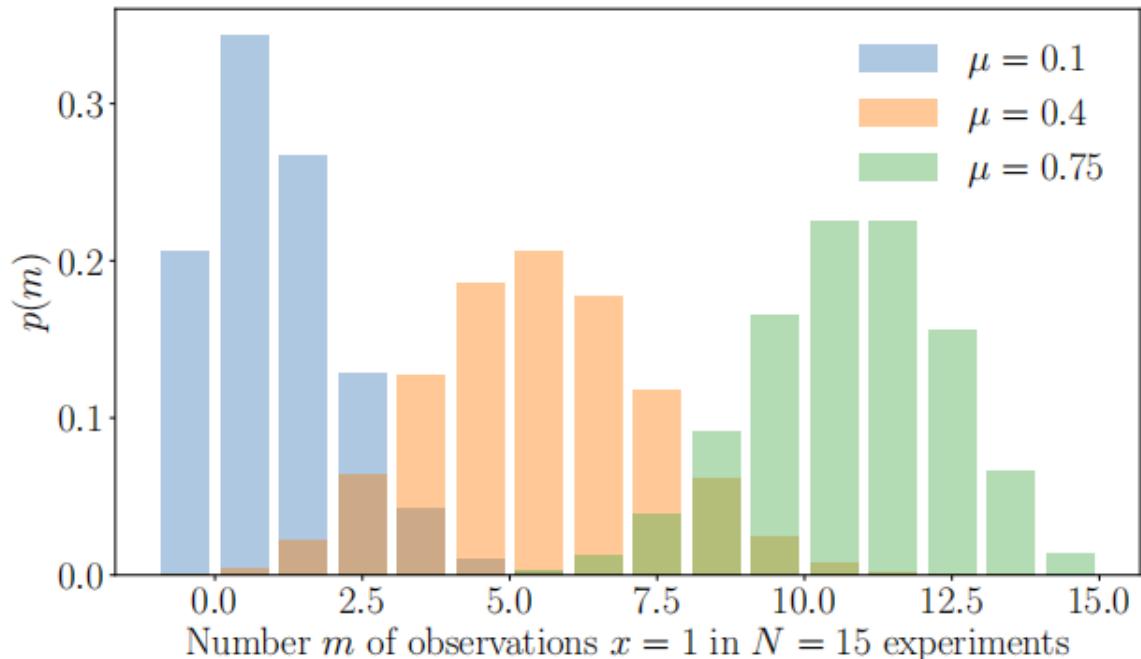


图6.10 $\mu \in \{0.1、0.4、0.75\}$ 和 $N = 15$ 的二项式分布示例。

备注。上面改写伯努利分布，我们使用布尔变量作为数值0或1，用指数表示，是机器学习教科书中经常使用的技巧。另一种情况是在表示多项分布时。

例 6.9 (二项分布)

二项分布是伯努利分布在整数上的一个推广（如图6.10所示）。特别是，二项分布可以用于描述从伯努利分布中抽取 N 个样本时，观察到 $X = 1$ 出现 m



次的概率，其中 $p(X = 1) = \mu \in [0, 1]$ 。二项分布 $\text{Bin}(N, \mu)$ 定义为

(6.95)

$$p(m | N, \mu) = \binom{N}{m} \mu^m (1 - \mu)^{N-m},$$

$$\mathbb{E}[m] = N\mu,$$

$$\mathbb{V}[m] = N\mu(1 - \mu),$$

其中， $\mathbb{E}[m]$ 和 $\mathbb{V}[m]$ 分别是 m 的均值和方差。

二项式的一个例子是，如果我们想描述在 N 个抛硬币实验中观察到 m 个“头”的概率，如果在单个实验中观察到头部的概率是 μ 。

例 6.10 (贝塔分布)

我们可能希望对有限区间上的连续随机变量进行建模。贝塔分布是一个在连续随机变量 $\mu \in [0, 1]$ 上的分布，它经常用于表示某些二元事件（例如，控制伯努利分布的参数）的概率。贝塔分布 $\text{Beta}(\alpha, \beta)$ （如图6.11所示）本身由两个参数 $\alpha > 0, \beta > 0$ 控制，并定义为

(6.98)

$$p(\mu | \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \mu^{\alpha-1} (1 - \mu)^{\beta-1}$$

$$\mathbb{E}[\mu] = \frac{\alpha}{\alpha + \beta}, \quad \mathbb{V}[\mu] = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$$

(6.99)

其中， $\Gamma(\cdot)$ 是伽马函数，定义为

$$\Gamma(t) := \int_0^\infty x^{t-1} \exp(-x) dx, \quad t > 0. \quad \Gamma(t+1) = t\Gamma(t).$$

(6.100)

(6.101)

请注意，(6.98)中的伽马函数之比用于对贝塔分布进行归一化。

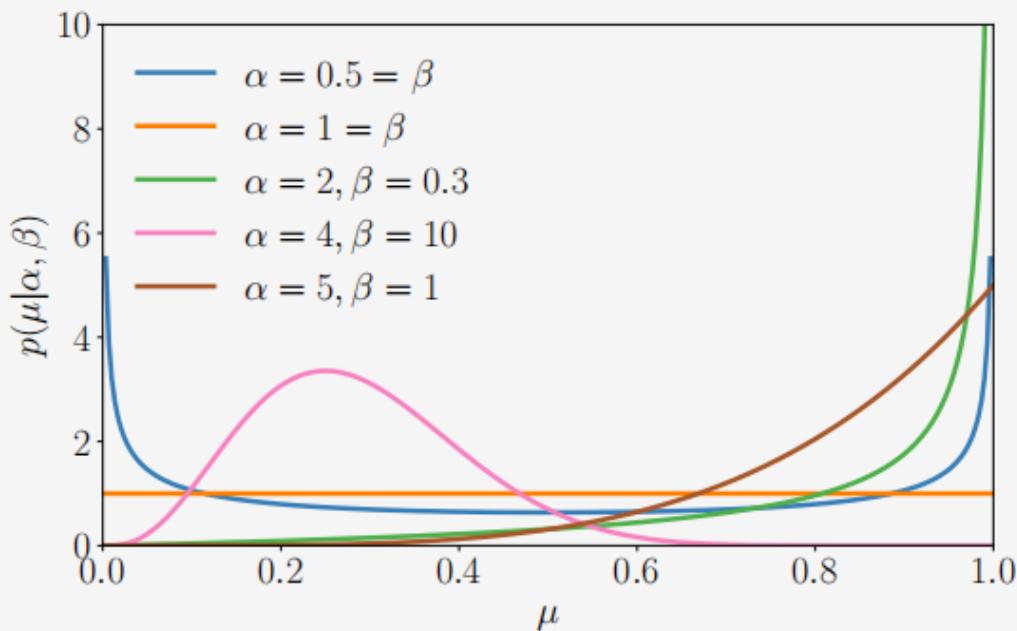


图6.11 α 和 β 不同值的Beta分布示例。

直观上， α 将概率质量向 1 移动，而 β 将概率质量向 0 移动。这里有一些特殊情况 (Murphy, 2012)：

- 当 $\alpha = 1 = \beta$ 时，我们得到均匀分布 $\mathcal{U}[0,1]$ 。
- 当 $\alpha, \beta < 1$ 时，我们得到一个在 0 和 1 处有尖峰的双峰分布。
- 当 $\alpha, \beta > 1$ 时，分布是单峰的。
- 当 $\alpha, \beta > 1$ 且 $\alpha = \beta$ 时，分布是单峰的、对称的，并且以区间 $[0,1]$ 为中心，即众数/均值为 $\frac{1}{2}$ 。

备注。存在大量具有名称的分布，它们之间以不同的方式相互关联 (Leemis 和 McQueston, 2008)。值得注意的是，每个命名的分布都是出于特定原因而创建的，但可能具有其他应用。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。我们介绍了前面的三个分布，以便能够说明共轭性（第 6.6.1 节）和指数族（第 6.6.3 节）的概念。

6.6.1 共轭性

根据贝叶斯定理 (6.23)，后验概率与先验概率和似然函数的乘积成正比。先验概率的指定可能由于两个原因而变得棘手：首先，先验概率应该包含我们在看到任何数据

之前对问题的了解，这通常很难描述。其次，通常不可能通过解析方法计算后验分布。然而，有一些计算上方便的先验概率：共轭先验。

定义 6.13（共轭先验） 如果后验概率与先验概率具有相同的形式/类型，则该先验概率是似然函数的共轭先验。

共轭性特别方便，因为我们可以更新先验分布的参数来代数地计算后验分布。

备注 在考虑概率分布的几何形状时，共轭先验保留了与似然函数相同距离结构 (Agarwal 和 Daumé III, 2010)。



为了介绍共轭先验的一个具体例子，我们在示例 6.11 中描述了二项分布（定义在离散随机变量上）和贝塔分布（定义在连续随机变量上）。

例 6.11（贝塔-二项共轭性）

考虑一个二项随机变量 $x \sim \text{Bin}(N, \mu)$ ，其中

$$p(x | N, \mu) = \binom{N}{x} \mu^x (1 - \mu)^{N-x}, \quad x = 0, 1, \dots, N,$$

(6.102)

表示在 N 次硬币抛掷中找到 x 次“正面”的概率，其中 μ 是出现“正面”的概率。我们对参数 μ 放置一个贝塔先验，即 $\mu \sim \text{Beta}(\alpha, \beta)$ ，其中

$$p(\mu | \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \mu^{\alpha-1} (1 - \mu)^{\beta-1}.$$

(6.103)

如果我们现在观察到某个结果 $x = h$ ，即在 N 次硬币抛掷中看到 h 次“正面”，我们可以计算 μ 的后验分布为

(6.104a) (6.104b)



$$\begin{aligned}
 p(\mu | x = h, N, \alpha, \beta) &\propto p(x | N, \mu)p(\mu | \alpha, \beta) \\
 &\propto \mu^h(1-\mu)^{N-h}\mu^{\alpha-1}(1-\mu)^{\beta-1} \\
 &= \mu^{h+\alpha-1}(1-\mu)^{(N-h)+\beta-1}
 \end{aligned}$$

(6.104c)

(6.104d)

$$\propto \text{Beta}(h + \alpha, N - h + \beta),$$

即，后验分布是一个贝塔分布，与先验分布相同，即贝塔先验是二项似然函数中参数 μ 的共轭先验。

表6.2常见似然函数的共轭先验的例子。

Likelihood	Conjugate prior	Posterior
Bernoulli	Beta	Beta
Binomial	Beta	Beta
Gaussian	Gaussian/inverse Gamma	Gaussian/inverse Gamma
Gaussian	Gaussian/inverse Wishart	Gaussian/inverse Wishart
Multinomial	Dirichlet	Dirichlet

在下面的例子中，我们将得到一个类似于贝塔二项共轭结果的结果。这里我们将证明贝塔分布是伯努利分布的共轭先验。

例 6.12 (贝塔-伯努利共轭性)

设 $x \in \{0, 1\}$ 根据参数为 $\theta \in [0, 1]$ 的伯努利分布进行分布，即 $p(x = 1 | \theta) = \theta$ 。这也表示为 $p(x | \theta) = \theta^x(1-\theta)^{1-x}$ 。设 θ 根据参数为 α, β 的贝塔分布进行分布，即 $p(\theta | \alpha, \beta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$ 。

将贝塔分布和伯努利分布相乘，我们得到

(6.105a)

(6.105b)



$$\begin{aligned}
 p(\theta | x, \alpha, \beta) &= p(x | \theta)p(\theta | \alpha, \beta) \\
 &\propto \theta^x(1-\theta)^{1-x}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\
 &= \theta^{\alpha+x-1}(1-\theta)^{\beta+(1-x)-1} \\
 &\propto p(\theta | \alpha + x, \beta + (1 - x)).
 \end{aligned}$$

(6.105c)

(6.105d)

最后一行是参数为 $(\alpha + x, \beta + (1 - x))$ 的贝塔分布。

表6.2列出了在概率建模中使用的一些标准似然函数参数的共轭先验示例。在任何统计文本中都可以找到如伽马先验、多项式先验、逆伽马先验、逆Wishart先验和狄利克雷先验等分布是共轭的，例如在Bishop (2006) 中就有描述。

贝塔分布是单变量高斯分布、二项分布和伯努利似然函数中参数 μ 的共轭先验。对于高斯似然函数，我们可以在均值上放置一个共轭的高斯先验。表中高斯似然函数出现两次的原因是我们需要区分单变量和多变量的情况。在单变量（标量）情况下，逆伽马是先验方差的共轭先验。在多变量情况下，我们使用共轭的逆Wishart分布作为协方差矩阵的先验。狄利克雷分布是多项式似然函数的共轭先验。更多详情，请参阅Bishop (2006)。

6.6.2 充分统计量

回顾一下，随机变量的统计量是该随机变量的一个确定性函数。例如，如果 $x = [x_1, \dots, x_N]^\top$ 是一个单变量高斯随机变量的向量，即 $x_n \sim \mathcal{N}(\mu, \sigma^2)$ ，那么样本均值 $\hat{\mu} = \frac{1}{N}(x_1 + \dots + x_N)$ 就是一个统计量。罗纳德·费希尔爵士 (Sir Ronald Fisher) 发现了充分统计量的概念：即存在统计量，它们包含了可以从与所考虑分布相对应的数据中推断出的所有可用信息。换句话说，充分统计量携带了关于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。

对于一组由 θ 参数化的分布，设 X 是一个具有分布 $p(x | \theta_0)$ 的随机变量，其中 θ_0 是未知的。如果统计量的向量 $\phi(x)$ 包含了关于 θ_0 的所有可能信息，则称 $\phi(x)$ 为 θ_0 的充分统计量。为了更正式地说明“包含所有可能信息”，这意味着在给定 θ 的情况下， x 的概率可以分解为两部分：一部分不依赖于 θ ，另一部分仅通过 $\phi(x)$ 依赖于 θ 。费希尔-奈

曼 (Fisher-Neyman) 分解定理正式化了这一概念，我们在定理6.14中不加证明地叙述了这一定理。

定理6.14 (Fisher-Neyman) 。[Lehmann和Casella (1998) 中的定理6.5]设 X 具有概率密度函数 $p(x | \theta)$ 。那么，统计量 $\phi(x)$ 是 θ 的充分统计量当且仅当 $p(x | \theta)$ 可以写成以下形式：

$$p(x | \theta) = h(x)g_{\theta}(\phi(x)),$$

(6.106)

其中 $h(x)$ 是与 θ 无关的分布，而 g_{θ} 通过充分统计量 $\phi(x)$ 捕获了所有对 θ 的依赖。

如果 $p(x | \theta)$ 不依赖于 θ ，那么对于任何函数 ϕ ， $\phi(x)$ 显然是 θ 的充分统计量。更有趣的情况是 $p(x | \theta)$ 仅依赖于 $\phi(x)$ 而不依赖于 x 本身。在这种情况下， $\phi(x)$ 是 θ 的充分统计量。

在机器学习中，我们考虑从分布中抽取的有限数量的样本。可以想象，对于简单的分布（如示例6.8中的伯努利分布），我们只需要少量样本就可以估计分布的参数。我们还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），那么哪个分布最适合这些数据？一个自然的问题是，随着我们观察更多数据，是否需要更多参数 θ 来描述分布？一般来说，答案是肯定的，这在非参数统计中有所研究

(Wasserman, 2007)。一个相反的问题是考虑哪一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，这将在下一节中介绍。

6.6.3 指数族分布

在考虑分布（无论是离散随机变量还是连续随机变量的分布）时，我们可以有三个不同层次的抽象。在第一层次（最具体的一端），我们有一个具有固定参数的特定命名分布，例如均值为0、方差为1的单变量高斯分布 $\mathcal{N}(0, 1)$ 。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（如单变量高斯分布），并从数据中推断参数。例如，我们假设一个具有未知均值 μ 和未知方差 σ^2 的单变量高斯分布 $\mathcal{N}(\mu, \sigma^2)$ ，并使用最大似然拟合来确定最佳参数 (μ, σ^2) 。在第9章考虑线性回归时，我们将看到这样的例子。第三层次的抽象是考虑分布族，而在本书中，我们关注的是指数族分布。单变量高斯分布是指数族分布的一个成员。表6.2中列出的许多广泛使用的统计模型，包括所有“命名”模型，都是指数族分布的成员。它们都可以统一到一个概念下 (Brown, 1986)。

注。一个简短的历史趣闻：像数学和科学中的许多概念一样，指数族分布也是由不同的研究者同时独立发现的。在1935-1936年间，塔斯马尼亚的埃德温·皮特曼（Edwin Pitman）、巴黎的乔治·达莫瓦（Georges Darmois）和纽约的伯纳德·库普曼（Bernard Koopman）分别证明了在重复独立抽样下，指数族分布是唯一具有有限维充分统计量的分布族（Lehmann and Casella, 1998）。



指数族分布是一个由参数 $\theta \in \mathbb{R}^D$ 参数化的概率分布族，其形式为 (6.107)

$$p(\mathbf{x} | \boldsymbol{\theta}) = h(\mathbf{x}) \exp (\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle - A(\boldsymbol{\theta})) ,$$

其中 $\phi(\mathbf{x})$ 是充分统计量的向量。一般来说，在(6.107)中可以使用任何内积（第3.2节），为了具体起见，我们将在这里使用标准点积 ($\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle = \boldsymbol{\theta}^\top \phi(\mathbf{x})$)。注意，指数族分布的形式本质上是费希尔-奈曼定理（定理6.14）中 $g_\theta(\phi(\mathbf{x}))$ 的一个特定表达式。

通过将另一个条目 ($\log h(\mathbf{x})$) 添加到充分统计量向量 $\phi(\mathbf{x})$ 中，并约束对应的参数 $\theta_0 = 1$ ，因子 $h(\mathbf{x})$ 可以被吸收到点积项中。项 $A(\boldsymbol{\theta})$ 是归一化常数，它确保分布的总和或积分为1，被称为对数配分函数。忽略这两个项，并将指数族分布视为形式如下的分布，我们可以获得对指数族分布的良好直观理解：

(6.108)

$$p(\mathbf{x} | \boldsymbol{\theta}) \propto \exp (\boldsymbol{\theta}^\top \phi(\mathbf{x})) .$$

对于这种参数化形式，参数 $\boldsymbol{\theta}$ 被称为自然参数。乍一看，指数族分布似乎只是通过在点积的结果上添加指数函数而进行的平凡变换。然而，由于我们能够在 $\phi(\mathbf{x})$ 中捕获有关数据的信息，这带来了许多便于建模和高效计算的启示。

示例 6.13 (高斯分布作为指数族分布)

考虑单变量高斯分布 $\mathcal{N}(\mu, \sigma^2)$ 。令 $\phi(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$ 。

然后，利用指数族分布的定义 (6.109) 式，我们有

$$p(x | \boldsymbol{\theta}) \propto \exp(\theta_1 x + \theta_2 x^2)$$

设定

$$\boldsymbol{\theta} = \left[\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2} \right]^\top$$

(6.110)

并将此代入 (6.109) 式，我们得到

$$p(x | \boldsymbol{\theta}) \propto \exp \left(\frac{\mu x}{\sigma^2} - \frac{x^2}{2\sigma^2} \right) \propto \exp \left(-\frac{1}{2\sigma^2}(x - \mu)^2 \right)$$

(6.111)

因此，单变量高斯分布是指数族分布的一个成员，其充分统计量为 $\phi(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$ ，自然参数由 (6.110) 式中的 $\boldsymbol{\theta}$ 给出。

示例 6.14 (伯努利分布作为指数族分布) 回顾示例 6.8 中的伯努利分布

(6.112)

$$p(x | \mu) = \mu^x (1 - \mu)^{1-x}, \quad x \in \{0, 1\}.$$

(6.113a)

这个分布可以写成指数族分布的形式：

$$\begin{aligned} p(x | \mu) &= \exp [\log (\mu^x (1 - \mu)^{1-x})] \\ &= \exp [x \log \mu + (1 - x) \log (1 - \mu)] \\ &= \exp [x \log \mu - x \log (1 - \mu) + \log (1 - \mu)] \\ &= \exp \left[x \log \frac{\mu}{1 - \mu} + \log (1 - \mu) \right]. \end{aligned}$$

(6.113c)

(6.113d)

最后一行 (6.113d) 可以识别为符合指数族分布的形式 (6.107)，通过观察可知


$$h(x) = 1$$

(6.114)

(6.115)

$$\theta = \log \frac{\mu}{1 - \mu}$$

$$\phi(x) = x$$

$$A(\theta) = -\log(1 - \mu) = \log(1 + \exp(\theta)).$$

(6.116)

(6.117)

θ 和 μ 之间的关系是可逆的，因此

(6.118)

$$\mu = \frac{1}{1 + \exp(-\theta)}.$$

关系式 (6.118) 用于获得 (6.117) 中的右侧等式。

原始伯努利参数 μ 和自然参数 θ 之间的关系被称为 **sigmoid** 函数或逻辑函数。值得注意的是， μ 的取值范围是 $(0, 1)$ ，但 θ 的取值范围是 \mathbb{R} ，因此 **sigmoid** 函数将实数值压缩到 $(0, 1)$ 范围内。这一性质在机器学习中非常有用，例如，在逻辑回归 (Bishop, 2006, 第 4.3.2 节) 中，以及在神经网络中作为非线性激活函数 (Goodfellow et al., 2016, 第 6 章) 中都会用到它。

◇ 对于如何找到某个特定分布的共轭分布的参数形式，这通常并不明显（例如，表 6.2 中的那些分布）。指数族分布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 是指数族分布 (6.107) 的一个成员：

(6.119)

$$p(\mathbf{x} | \boldsymbol{\theta}) = h(\mathbf{x}) \exp (\langle \boldsymbol{\theta}, \phi(\mathbf{x}) \rangle - A(\boldsymbol{\theta})) .$$

指数族的每一个成员都有一个共轭先验 (Brown, 1986)

(6.120)



$$p(\boldsymbol{\theta} \mid \boldsymbol{\gamma}) = h_c(\boldsymbol{\theta}) \exp \left(\left\langle \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{\theta} \\ -A(\boldsymbol{\theta}) \end{bmatrix} \right\rangle - A_c(\boldsymbol{\gamma}) \right),$$

其中 $\boldsymbol{\gamma} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}$ 的维度是 $\dim(\boldsymbol{\theta}) + 1$ 。共轭先验的充分统计量或共轭先验是 $\begin{bmatrix} \boldsymbol{\theta} \\ -A(\boldsymbol{\theta}) \end{bmatrix}$ 。利用指数族共轭先验一般形式的知识，我们可以推导出与特定分布相对应的共轭先验的函数形式。

正如上一节所述，指数族的主要动机是它们具有有限维充分统计量。此外，共轭分布很容易写出来，而且共轭分布也来自指数族。从推断的角度来看，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（回想一下高斯分布的均值和协方差）。从优化的角度来看，对数似然函数是凹函数，允许应用有效的优化方法（第7章）。

< 上一章节

下一章节 >

6.5 高斯分布

6.7 变量变换/逆变换



6.7 变量变换/逆变换

虽然已知的分布种类似乎很多，但实际上，我们有明确名称的分布集合是相当有限的。因此，了解变换后的随机变量的分布方式通常很有用。例如，假设 X 是一个服从单变量正态分布 $\mathcal{N}(0, 1)$ 的随机变量，那么 X^2 的分布是什么？另一个在机器学习中很常见的例子是，如果 X_1 和 X_2 都是标准正态分布的单变量，那么 $\frac{1}{2}(X_1 + X_2)$ 的分布是什么？

计算 $\frac{1}{2}(X_1 + X_2)$ 分布的一种方法是先计算 X_1 和 X_2 的均值和方差，然后再进行组合。正如我们在6.4.4节中看到的，当我们考虑随机变量的仿射变换时，可以计算出结果随机变量的均值和方差。然而，我们可能无法获得变换后分布的函数形式。此外，我们可能对随机变量的非线性变换感兴趣，而这些变换的闭式表达式并不容易获得。

注（符号）：在本节中，我们将明确随机变量及其取值。因此，请回忆一下，我们使用大写字母 X, Y 来表示随机变量，使用小写字母 x, y 来表示随机变量在目标空间 \mathcal{T} 中取的值。我们将离散随机变量 X 的概率质量函数（PMF）明确写为 $P(X = x)$ 。对于连续随机变量 X （第6.2.2节），概率密度函数（PDF）写为 $f(x)$ ，累积分布函数（CDF）写为 $F_X(x)$ 。

◆

我们将探讨两种方法来获得随机变量变换后的分布：一种方法是使用累积分布函数的定义进行直接计算，另一种方法是使用微积分中的链式法则（第5.2.2节）的变量变换方法。变量变换方法被广泛使用，因为它提供了一个尝试计算变换后分布的“配方”。我们将解释针对单变量随机变量的技术，并仅简要给出多变量随机变量一般情况的结果。

离散随机变量的变换可以通过直接变换来理解（第6.2.1节），并考虑一个可逆函数 $U(x)$ 。考虑变换后的随机变量 $Y := U(X)$ ，其PMF为 $P(Y = y)$ 。那么

$$P(Y = y) = P(U(X) = y) \quad (\text{感兴趣的变换}, \text{ 6.125a})$$

$$= P(X = U^{-1}(y)) \quad (\text{逆变换}, \text{ 6.125b})$$

其中我们可以观察到 $x = U^{-1}(y)$ 。因此，对于离散随机变量，变换直接改变了各个事件（同时适当地变换了概率）。

6.7.1 分布函数技术

分布函数技术回归到基本原理，利用累积分布函数（CDF） $F_X(x) = \bar{P}(X \leq x)$ 的定义，以及其微分为概率密度函数（PDF） $f(x)$ 的事实（Wasserman, 2004, 第2章）。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：

$$F_Y(y) = P(Y \leq y)$$

(6.126)

2. 对CDF $F_Y(y)$ 求导，得到PDF $f(y)$ 。

(6.127)

$$f(y) = \frac{d}{dy} F_Y(y)$$

我们还需要记住，由于 U 的变换，随机变量的定义域可能已经改变。

例6.16 设 \dot{X} 是一个连续随机变量，其概率密度函数为

在 $0 \leq x \leq 1$ 上

$$f(x) = 3x^2$$

(6.128)

我们感兴趣的是找到 $Y = X^2$ 的概率密度函数（PDF）。

函数 f 是 x 的增函数，因此得到的 y 值位于区间 $[0,1]$ 内。我们得到 (6.129a)
(6.129b) (6.129c) (6.129d)



$$\begin{aligned}
 F_Y(y) &= P(Y \leq y) && \text{CDF的定义} \\
 &= P(X^2 \leq y) && \text{感兴趣的变换} \\
 &= P(X \leq y^{\frac{1}{2}}) && \text{逆变换} \\
 &= F_X(y^{\frac{1}{2}}) && \text{CDF的定义} \\
 &= \int_0^{y^{\frac{1}{2}}} 3t^2 dt && \text{CDF作为定积分} \\
 &= [t^3]_{t=0}^{t=y^{\frac{1}{2}}} && \text{积分结果} \\
 &= y^{\frac{3}{2}}, \quad 0 \leq y \leq 1.
 \end{aligned}$$

(6.129e) (6.129f) (6.129g) 因此, Y 的CDF为

$$F_Y(y) = y^{\frac{3}{2}}$$

(6.130)

对于 $0 \leq y \leq 1$ 。为了得到PDF, 我们对CDF求导

(6.131)

$$f(y) = \frac{d}{dy} F_Y(y) = \frac{3}{2} y^{\frac{1}{2}}$$

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的函数 $f(x) = 3x^2$ 。这意味着我们可以计算其反函数。一般来说, 我们要求感兴趣的函数 $y = U(x)$ 具有反函数 $x = U^{-1}(y)$ 。通过考虑随机变量 X 的累积分布函数 $F_X(x)$, 并将其用作变换 $U(x)$, 我们可以得到一个有用的结果。这导致了以下定理。

定理6.15. [Casella和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 $F_X(x)$ 的连续随机变量。那么定义为

(6.132)

$$Y := F_X(X)$$

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，它用于通过转换来自均匀随机变量的抽样结果来推导从分布中抽样的算法（Bishop, 2006）。该算法的工作原理是，首先从一个均匀分布中生成一个样本，然后通过逆cdf（假设这是可用的）转换它，以从期望的分布中获得一个样本。概率积分变换也用于假设检验一个样本是否来自一个特定的分布（Lehmann和Romano, 2005）。cdf的输出给出均匀分布的想法也形成了连接的基础（Nelsen, 2006）。

6.7.2 变量替换

第6.7.1节中的分布函数技术是基于第一性原理推导出来的，它基于累积分布函数（CDF）的定义，并利用反函数、微分和积分的性质。这种从第一性原理出发的论证依赖于两个事实：

1. 我们可以将 Y 的CDF转换为 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过程，以理解定理6.16中更一般的变量替换方法。

注记：“变量替换”的名称来源于在面对复杂积分时改变积分变量的想法。对于单变量函数的变量替换，我们使用积分的换元法，

$$\int f(g(x))g'(x)dx = \int f(u)du, \quad \text{where } u = g(x).$$

(6.133)

这个规则的推导基于微积分的链式法则（5.32）并应用微积分基本定理两次。微积分基本定理将积分和微分在某种程度上形式化为“逆”运算。通过（大致上）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，我们可以直观地理解这个规则。通过将 $u = g(x)$ 代入，积分右侧括号内的自变量变为 $f(g(x))$ 。通过假设 du 可以近似为 $du \approx \Delta u = g'(x)\Delta x$ ，且 $dx \approx \Delta x$ ，我们得到(6.133)。

◇

考虑一个单变量随机变量 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 的状态 x 位于区间 $[a, b]$ 。根据CDF的定义，我们有

(6.134)

$$F_Y(y) = P(Y \leq y).$$



我们感兴趣的是随机变量的函数 U

(6.135)

$$P(Y \leq y) = P(U(X) \leq y),$$

其中我们假设函数 U 是可逆的。在区间上的可逆函数要么是严格递增的，要么是严格递减的。在 U 严格递增的情况下，其反函数 U^{-1} 也是严格递增的。通过对 $P(U(X) \leq y)$ 的自变量应用反函数 U^{-1} ，我们得到

$$P(U(X) \leq y) = P(U^{-1}(U(X)) \leq U^{-1}(y)) = P(X \leq U^{-1}(y))$$

(6.136) (6.136) 中的最右边项是 X 的CDF的表达式。回想一下根据PDF定义的CDF

$$P(X \leq U^{-1}(y)) = \int_a^{U^{-1}(y)} f(x)dx.$$

(6.137)

现在我们有了关于 x 的 Y 的CDF的表达式：

$$F_Y(y) = \int_a^{U^{-1}(y)} f(x)dx.$$

(6.138)

为了得到PDF，我们对(6.138)关于 y 求导：

$$f(y) = \frac{d}{dy} F_y(y) = \frac{d}{dy} \int_a^{U^{-1}(y)} f(x)dx.$$

(6.139)

请注意，右侧的积分是关于 x 的，但我们需要一个关于 y 的积分，因为我们要对 y 求导。特别地，我们使用(6.133)进行替换

$$\int f(U^{-1}(y))U^{-1'}(y)dy = \int f(x)dx \quad \text{where } x = U^{-1}(y).$$

将(6.140)应用于(6.139)的右侧，我们得到



$$f(y) = \frac{d}{dy} \int_a^{U^{-1}(y)} f_x(U^{-1}(y)) U^{-1}'(y) dy .$$

(6.141)

然后，我们回忆到微分是一个线性算子，并且我们使用下标 x 来提醒自己 $f_x(U^{-1}(y))$ 是 x 的函数而不是 y 的函数。再次调用微积分基本定理，我们得到

$$f(y) = f_x(U^{-1}(y)) \cdot \left(\frac{d}{dy} U^{-1}(y) \right) .$$

(6.142)

回忆一下，我们假设 U 是一个严格增函数。对于减函数，按照相同的推导，我们会发现前面有一个负号。为了对增函数和减函数都使用相同的表达式，我们引入微分的绝对值：

$$f(y) = f_x(U^{-1}(y)) \cdot \left| \frac{d}{dy} U^{-1}(y) \right| .$$

(6.143)

这被称为变量变换技术。在变量变换技术中，术语 $\left| \frac{d}{dy} U^{-1}(y) \right|$

(6.143)用于衡量在应用 U 时单位体积的变化量（也请参见第5.3节中Jacobian行列式的定义）。

备注。与(6.125b)中的离散情况相比，我们有一个额外的因子 $\left| \frac{d}{dy} U^{-1}(y) \right|$ 。连续情况需要更加小心，因为对于所有 y , $P(Y = y) = 0$ 。概率密度函数 $f(y)$ 不能描述为涉及 y 的事件的概率。

◇

到目前为止，在本节中，我们一直在研究单变量变量的变换。多变量随机变量的情况类似，但由于绝对值不能用于多变量函数，因此情况更为复杂。相反，我们使用Jacobian矩阵的行列式。从(5.58)中回忆，Jacobian是一个偏导数矩阵，非零行列式的存在表明我们可以对Jacobian进行求逆。回想第4.1节中的讨论，行列式的出现是因为我们的微分（体积的立方体）通过Jacobian转换成平行六面体。让我们在以下定理中总结前面的讨论，该定理为我们提供了多变量变量变换的方法。

定理6.16。[比林斯利 (1995) 中的定理17.2]设 $f(\mathbf{x})$ 是多元连续随机变量 \mathbf{X} 的概率密度的值。如果向量值函数 $\mathbf{y} = \mathbf{U}(\mathbf{x})$ 在其定义域内的所有值上都是可微且可逆的，则对于对应的 \mathbf{y} 值， $\mathbf{Y} = \mathbf{U}(\mathbf{X})$ 的概率密度由下式给出：

$$f(\mathbf{y}) = f_{\mathbf{x}}(\mathbf{U}^{-1}(\mathbf{y})) \cdot \left| \det \left(\frac{\partial}{\partial \mathbf{y}} \mathbf{U}^{-1}(\mathbf{y}) \right) \right|.$$

(6.144)

这个定理初看起来有些吓人，但关键点是多元随机变量的变量变换遵循单变量变量变换的程序。首先，我们需要求出逆变换，并将其代入 \mathbf{x} 的密度函数中。然后，我们计算Jacobian矩阵的行列式并乘以结果。以下示例说明了双变量随机变量的情况。

例6.17 考虑一个双变量随机变量 \mathbf{X} ，其状态为 $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ ，且概率密度函数为

$$f \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) = \frac{1}{2\pi} \exp \left(-\frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right).$$

(6.145)

我们使用定理6.16中的变量变换技术来推导随机变量的线性变换（第2.7节）的效果。考虑一个矩阵 $\mathbf{A} \in \mathbb{R}^{2 \times 2}$ ，定义为

(6.146)

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

我们感兴趣的是找到变换后的双变量随机变量 \mathbf{Y} （其状态为 $\mathbf{y} = \mathbf{Ax}$ ）的概率密度函数。回忆一下，对于变量变换，我们需要将 \mathbf{x} 作为 \mathbf{y} 的函数进行逆变换。由于我们考虑的是线性变换，逆变换由矩阵的逆给出（参见第2.2.2节）。对于 2×2 矩阵，我们可以明确写出其公式，即

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}.$$

(6.147)

请注意， $ad - bc$ 是矩阵 A 的行列式（第4.1节）。

相应的概率密度函数由下式给出：

$$f(\mathbf{x}) = f(A^{-1}\mathbf{y}) = \frac{1}{2\pi} \exp\left(-\frac{1}{2}\mathbf{y}^\top A^{-\top} A^{-1}\mathbf{y}\right). \quad (6.148)$$

矩阵乘以向量关于向量的偏导数就是矩阵本身（第5.5节），因此

$$\frac{\partial}{\partial \mathbf{y}} A^{-1}\mathbf{y} = A^{-1}.$$

(6.149)

回忆第4.1节，逆矩阵的行列式是行列式的倒数，所以 Jacobian 矩阵的行列式为

(6.150)

$$\det\left(\frac{\partial}{\partial \mathbf{y}} A^{-1}\mathbf{y}\right) = \frac{1}{ad - bc}.$$

现在我们可以应用定理6.16中的变量变换公式，将(6.148)与(6.150)相乘，得到

(6.151a)

$$\begin{aligned} f(\mathbf{y}) &= f(\mathbf{x}) \left| \det\left(\frac{\partial}{\partial \mathbf{y}} A^{-1}\mathbf{y}\right) \right| \\ &= \frac{1}{2\pi} \exp\left(-\frac{1}{2}\mathbf{y}^\top A^{-\top} A^{-1}\mathbf{y}\right) |ad - bc|^{-1}. \end{aligned}$$

(6.151b)

尽管示例6.17是基于双变量随机变量的，这使得我们可以很容易地计算矩阵的逆，但前面的关系在高维情况下也是成立的。

备注。我们在第6.5节中看到，(6.148)中的密度函数 $f(\mathbf{x})$ 实际上是标准高斯分布，而变换后的密度函数 $f(\mathbf{y})$ 是具有协方差 $\Sigma = AA^\top$ 的双变量高斯分布。



我们将在本章中使用这些思想来描述第8.4节中的概率建模，并在第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。

[< 上一章节](#)

[下一章节 >](#)

6.6 共轭性与指数族分布

6.8 拓展阅读



6.8 拓展阅读

本章内容有时较为简洁。Grinstead和Snell（1997）以及Walpole et al.（2011）提供了更为宽松且适合自学的讲解。对概率的哲学层面感兴趣的读者可以阅读Hacking（2001），而与软件工程更相关的方法则由Downey（2014）介绍。Barndorff-Nielsen（2014）提供了指数族分布的概述。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。具有讽刺意味的是，最近神经网络兴趣的激增使得人们对概率模型有了更广泛的认识。例如，标准化流（normalizing flows）的概念（Jimenez Rezende和Mohamed, 2015）依赖于变量变换来转换随机变量。Goodfellow et al.（2016）所著书籍的第16至20章概述了将变分推断方法应用于神经网络的方法。

我们通过避免测度理论问题（Billingsley, 1995; Pollard, 2002）并假设我们已有实数以及实数上集合的定义方式及其适当的出现频率（而无需构建这些），从而绕过了连续随机变量中的大部分困难。这些细节在某些情况下很重要，例如在为连续随机变量 x, y 指定条件概率 $p(y | x)$ 时（Proschan和Presnell, 1998）。这种简略的记法隐藏了我们想要指定 $X = x$ （这是一个测度为零的集合）的事实。此外，我们还对 y 的概率密度函数感兴趣。更精确的记法应该是 $\mathbb{E}_y[f(y) | \sigma(x)]$ ，其中我们对测试函数 f 关于 x 的 σ -代数取 y 的期望。对概率论细节感兴趣的更专业的读者有很多选择（Jaynes, 2003; MacKay, 2003; Jacod和Protter, 2004; Grimmett和Welsh, 2014），包括一些非常技术性的讨论（Shiryayev, 1984; Lehmann和Casella, 1998; Dudley, 2002; Bickel和Doksum, 2006; Çinlar, 2011）。另一种研究概率的方法是从期望的概念出发，“逆向工作”以推导出概率空间所需的性质（Whittle, 2000）。随着机器学习使我们能够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将不得不理解这些更技术性的方面。以概率建模为重点的机器学习著作包括MacKay（2003）、Bishop（2006）、Rasmussen和Williams（2006）以及Barber（2012）以及Murphy（2012）的书籍。



< 上一章节

下一章节 >

6.7 变量变换/逆变换

习题



练习

6.1 考虑以下两个离散随机变量 X 和 Y 的双变量分布 $p(x, y)$ 。

计算：

- 边缘分布 $p(x)$ 和 $p(y)$ 。
- 条件分布 $p(x|Y = y_1)$ 和 $p(y|X = x_3)$ 。

6.2 考虑两个高斯分布的混合（如图6.4所示），

$$0.4\mathcal{N}\left(\begin{bmatrix} 10 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) + 0.6\mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 8.4 & 2.0 \\ 2.0 & 1.7 \end{bmatrix}\right).$$

- 计算每个维度的边缘分布。
- 计算每个边缘分布的均值、众数和中位数。
- 计算二维分布的均值和众数。

6.3 你编写了一个计算机程序，该程序有时会编译成功，有时会编译失败（代码没有改变）。你决定使用参数为 μ 的伯努利分布来模拟编译器的这种随机性（成功与不成功） x ：

$$p(x|\mu) = \mu^x(1 - \mu)^{1-x}, \quad x \in \{0, 1\}.$$

为伯努利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。第一个袋子里有四个芒果和两个苹果；第二个袋子里有四个芒果和四个苹果。

我们还有一个有偏的硬币，正面朝上的概率为0.6，反面朝上的概率为0.4。如果硬币正面朝上，我们从袋子1中随机挑选一个水果；否则，从袋子2中随机挑选一个水果。

你的朋友抛了硬币（你看不到结果），从对应的袋子中随机挑选了一个水果，并给了你一个芒果。

从袋子2中挑选出这个芒果的概率是多少?



提示: 使用贝叶斯定理。

6.5 考虑时间序列模型

$$\begin{aligned}x_{t+1} &= Ax_t + w, \quad w \sim \mathcal{N}(0, Q) \\y_t &= Cx_t + v, \quad v \sim \mathcal{N}(0, R),\end{aligned}$$

其中 w, v 是独立同分布的高斯噪声变量。进一步假设 $p(x_0) = \mathcal{N}(\mu_0, \Sigma_0)$ 。

a. $p(x_0, x_1, \dots, x_T)$ 的形式是什么? 证明你的答案 (你不需要显式地计算联合分布)。

b. 假设 $p(x_t | y_1, \dots, y_t) = \mathcal{N}(\mu_t, \Sigma_t)$ 。

1. 计算 $p(x_{t+1} | y_1, \dots, y_t)$ 。

2. 计算 $p(x_{t+1}, y_{t+1} | y_1, \dots, y_t)$ 。

3. 在时间 $t + 1$, 我们观察到值 $y_{t+1} = \hat{y}$ 。计算条件分布 $p(x_{t+1} | y_1, \dots, y_{t+1})$ 。

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, 该关系将数据集中示例之间的成对差异与方差的原始分数表达式联系起来。

6.8 将伯努利分布表示为指数族分布的自然参数形式, 参见 (6.107)。

6.9 将二项式分布表示为指数族分布。同样, 将贝塔分布表示为指数族分布。证明贝塔分布和二项式分布的乘积也是指数族分布的成员。

6.10 以两种方式推导出第6.5.2节中的关系:

a. 通过完成平方

b. 通过将高斯分布表达为其指数族形式

两个高斯分布 $\mathcal{N}(x | a, A)\mathcal{N}(x | b, B)$ 的乘积是一个未归一化的高斯分布 $c\mathcal{N}(x | c, C)$, 其中

\$\$\begin{aligned}&C = (A^{-1} + B^{-1})^{-1} \\&c = A^{-1}c + B^{-1}c\end{aligned}

\$\$\$\\begin{aligned}&C = (A^{-1} + B^{-1})^{-1} \\&c = A^{-1}c + B^{-1}c\end{aligned}



$$\&c=C(A^{-1}a+B^{-1}b)\backslash$$

$$\&c=(2\pi)^{-\frac{D}{2}}|A+B|^{\frac{1}{2}}\exp\left(-\frac{1}{2}(a-b)^T(A+B)(a-b)\right).$$

\end{aligned}\$\$

注意，归一化常数 c 本身可以视为在 a 或 b 上的（归一化）高斯分布，具有“膨胀”的协方差矩阵 $A + B$ ，即 $c = \mathcal{N}(a | b, A + B) = \mathcal{N}(b | a, A + B)$ 。

6.11 迭代期望

考虑两个具有联合分布 $p(x, y)$ 的随机变量 x, y 。证明

$$\mathbb{E}_X[x] = \mathbb{E}_Y[\mathbb{E}_X[x | y]].$$

这里， $\mathbb{E}_X[x | y]$ 表示在条件分布 $p(x | y)$ 下 x 的期望值。

6.12 高斯随机变量的操作

考虑一个高斯随机变量 $x \sim \mathcal{N}(x | \mu_x, \Sigma_x)$ ，其中 $x \in \mathbb{R}^D$ 。

此外，我们有

$$y = Ax + b + w,$$

其中 $y \in \mathbb{R}^E, A \in \mathbb{R}^{E \times D}, b \in \mathbb{R}^E$ ，且 $w \sim \mathcal{N}(w | 0, Q)$ 是独立的高斯噪声。

“独立”意味着 x 和 w 是独立的随机变量，且 Q 是对角的。

a. 写下似然 $p(y | x)$ 。

b. 分布 $p(y) = \int p(y | x)p(x)d\mathbf{x}$ 是高斯的。计算均值 μ_y 和协方差 Σ_y 。详细推导你的结果。

c. 随机变量 y 根据测量映射进行变换

$$z = Cy + v,$$

其中 $z \in \mathbb{R}^F, C \in \mathbb{R}^{F \times E}$ ，且 $v \sim \mathcal{N}(v | \mathbf{0}, R)$ 是独立的高斯（测量）噪声。

- 写下 $p(z | y)$ 。

- 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，测量了一个值 \hat{y} 。计算后验分布 $p(\mathbf{x} \mid \hat{\mathbf{y}})$ 。

解题提示：这个后验也是高斯的，即我们只需要确定其均值和协方差矩阵。首先明确计算联合高斯 $p(\mathbf{x}, \mathbf{y})$ 。这也需要我们计算交叉协方差 $\text{Cov}_{x,y}[\mathbf{x}, \mathbf{y}]^{**}$ 和 $\text{Cov}_{y,x}[\mathbf{y}, \mathbf{x}]^{**}$ 。然后应用高斯条件规则。

6.13 概率积分变换

给定一个连续随机变量 X ，其累积分布函数为 $F_X(x)$ ，证明随机变量 $Y := F_X(X)$ 是均匀分布的（定理6.15）。

< 上一章节

6.8 拓展阅读



第七章 连续优化

机器学习算法跑在计算机上，因此一切优化相关的数学设定需要被翻译为数值优化的方法。本章讲解了用于训练机器学习模型的简单数值方法。要训练一个机器学习模型，往往需要寻找一个最佳的参数集合，何谓“最佳”由目标函数或概率模型所确定（见本书的后半部分）。给定一个目标函数，我们会用优化算法找到它的最值（以及对应的最优参数集合）。

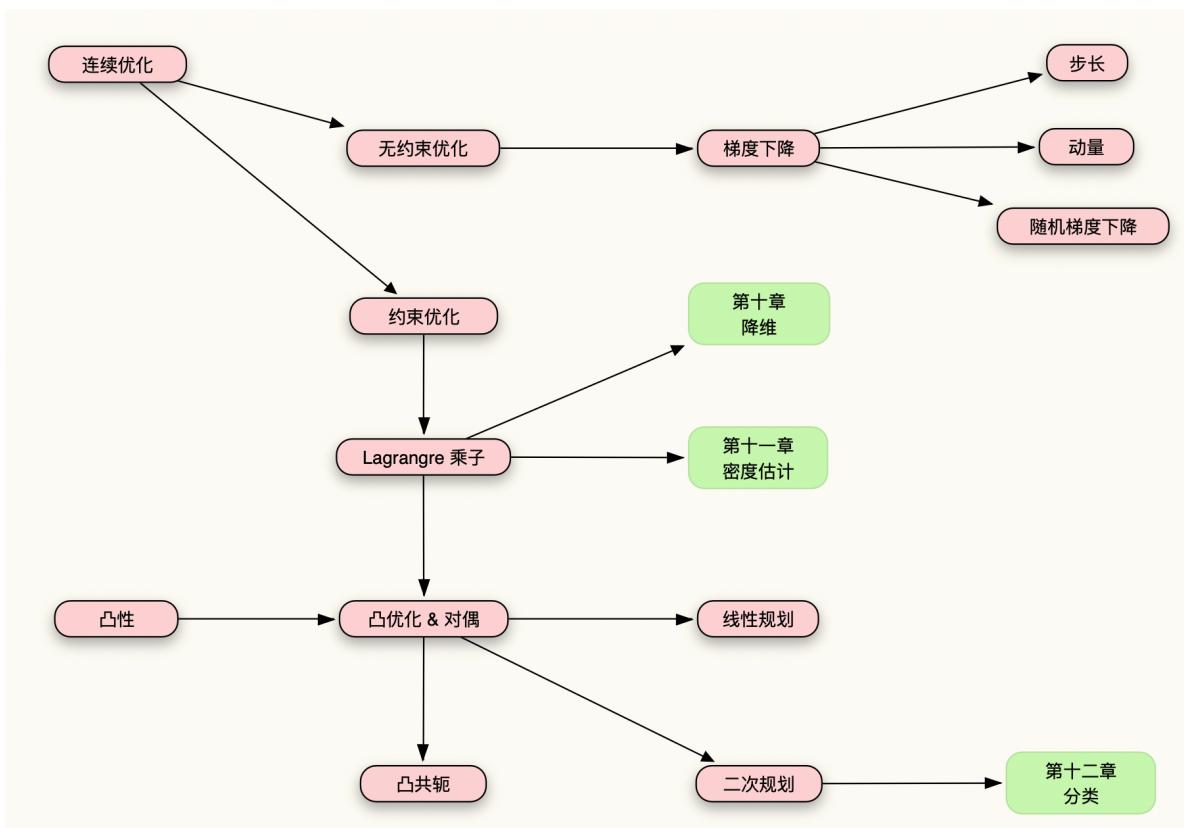


图 7.1 本章的概念地图

本章包括连续优化中的两个主要分支——无约束优化和约束优化，并总是假定目标函数是可微的（见第五章），这让我们能得到目标函数在空间中任意点出的梯度——这可以帮助我们找到最值点。一般地，大多数机器学习算法中我们要找对应目标函数的最小值（以及最小值点），所以直观的来说，我们要找目标函数的“谷底”，而目标函数的梯度总是指向目标函数的“高处”。所以我们的想法是逆着梯度的方向往“下”走，并祈祷我们能找到全局最小值点。对无约束优化而言，上述内容加上一些其他东西也就几乎是全部了（7.1 节）。但对于约束优化，我们要引进更多的概念来处理其中的“约束”（7.2 节）。除此之外，我们在 7.3 节中还将引入一类特殊的优化问题（凸优化），它们有非常好的性质，可以以一些方式达到全局最优。



注释

我们考虑的数据和模型工作在 \mathbb{R}^d 上，我们处理的优化问题称为 **连续优化 (continuous optimization)**；在另一边，也就是离散的世界，对应的优化问题称为 **组合优化 (combinatorial optimization)**。

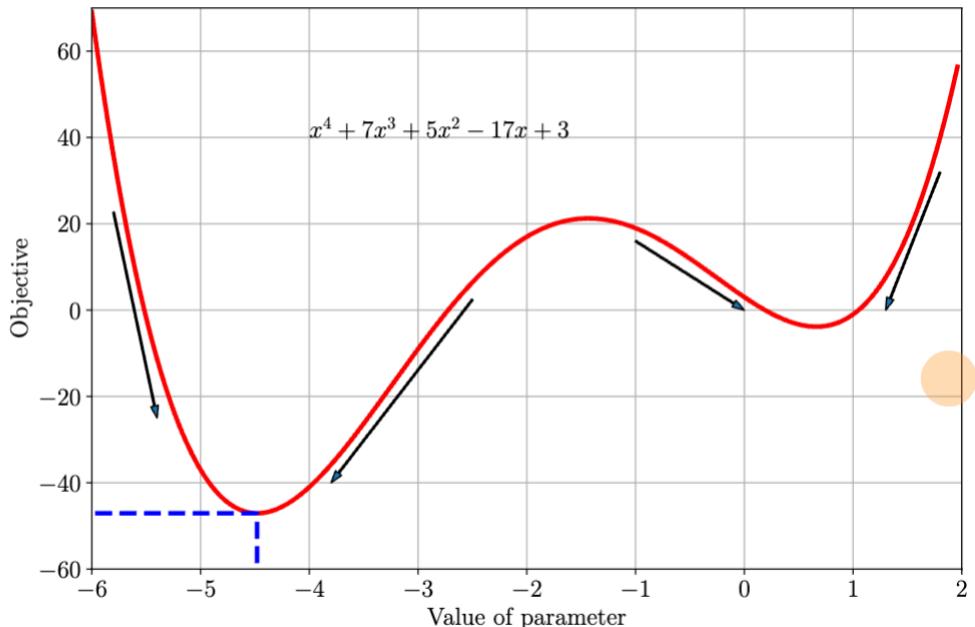


图 7.2 一个目标函数示例。负梯度的方向用箭头表示，全局最小值点用蓝色虚线表示

考虑图 7.2 中的函数，它在 $x = -4.5$ 附近有一个 **全局最小值点**，对应的函数值大概是 -47 。这是一个“光滑”的函数，我们可以用当前位置的梯度——它告诉我们应该向左走还是向右走——来找最小值。这样做其实假设我们处在一个盆地的结构：我们能看到函数在 $x = 0.7$ 附近有一个 **局部最小值点**。回忆一下，我们可以令导数为零得到函数的稳定点。

注释 稳定点是导函数的实根，对应所有梯度为零的点。

对于函数

$$\ell(x) = x^4 + 7x^3 + 5x^2 - 17x + 3, \quad (7.1)$$

它对 x 的导数是

$$\frac{d\ell(x)}{dx} = 4x^3 + 21x^2 + 10x - 17 \quad (7.2)$$

这是个三次方程，一般有三个零点。在本例中，两个对应着局部最小值点，一个对应着局部最大值点。我们要再求一次导，来看一阶导的零点（也就是函数的稳定点）处的二阶导数值的符号：

$$\frac{d^2\ell(x)}{dx^2} = 12x^2 + 42x + 10 \quad (7.3)$$

我们代入目测的三个极值点 $x = -4.5, -1.4, 0.7$ ，最终得到中间的那个极值点是一个局部最大值点 $\left(\frac{d^2\ell(x)}{dx^2} < 0\right)$ ，而其他两个稳定点是局部最小值点。

读者也许发现我们在上面的例子中刻意避免了求导数等于零这个方程的解析解——虽然对于阶数较低多项式我们可以这么做——但在一般情况下我们几乎不可能接触目标函数一阶导为零的解析解。因此我们转向数值解法：找一个初始点，例如 $x_0 = -6$ ，然后跟着负梯度的方向走。在图中，负梯度的箭头告诉我们我们应该向右走，但我们不知道应该向右走多远（这叫做 **步长**）。进一步地，如果我们的初始值取在右边（例如 $x_0 = 0$ ），负梯度方向将把我们带到一个“错误的”最小值点。我们在图 7.2 中可以看到，右侧的负梯度方向指向的是局部最优解，它对应的函数值大于函数的最小值（左侧）。

注释 根据 Abel-Ruffni 定理，五次及以上的多项式方程没有代数解。(Abel, 1826)

在 7.3 节中，我们将会见到一族称为“凸函数”的函数，初始点的选取不会对优化算法的结果造成像前文那样的影响。对于凸函数来说，它的局部最小值点总是全局最小值点。事实上，很多机器学习算法的目标函数都被设计为凸函数，我们将会在第十二章见到一个例子。

到此为止，本章的讨论内容仅仅局限于一元函数，这是为了更方便地展示梯度、下降方向和最优点。在接下来的内容，我们将在高位空间中使用类似的想法。不幸的是，一些概念不能简单推广到高维空间，所以在遇到似曾相识的概念时需要多加小心。



< 上一章节

下一章节 >

第六章 概率与统计

第八章 当模型遇上数据



7.1 基于梯度下降的优化

现在考虑求解一个实值函数最小值的问题：

$$\min_{\mathbf{x}} f(\mathbf{x}), \quad (7.4)$$

其中 $f : \mathbb{R}^d \rightarrow \mathbb{R}$ 是一个函数，它刻画了我们手中的机器学习问题。我们假设函数 f 是可微的，并且我们无法找到上述问题的解析解。

梯度下降是一个一阶优化算法。它的每次迭代都将估计点做一个正比于函数在该点处的负梯度向量的移动，以逐步找到一个局部最小值点。回顾第 5.1 节，梯度方向是函数值增长最快的方向。另一个有用的直观理解是考虑函数处于某个特定值处的那组线（即 $f(\mathbf{x}) = c$ ，其中某个值 $c \in \mathbb{R}$ ），这些线被称为等高线。梯度方向与我们希望优化的函数的等高线方向正交。

让我们考虑多变量函数。想象一个曲面（由函数 $f(\mathbf{x})$ 描述），并设想一个球从某个特定位置 \mathbf{x}_0 开始。当球被释放时，它会沿着最陡峭的下坡方向向下滚动。梯度下降利用了这样一个事实：从 \mathbf{x}_0 出发，若朝着函数 f 在 \mathbf{x}_0 处负的梯度方向 $-((\nabla f)(\mathbf{x}_0))^{\top}$ 移动， $f(\mathbf{x}_0)$ 的值将最快地减小。本书假设所涉及的函数都是可微的，并引导读者参考第 7.4 节中更一般的设置。于是假如我们考虑下面的更新：

$$\mathbf{x}_1 = \mathbf{x}_0 - \gamma [(\nabla f)(\mathbf{x}_0)]^{\top} \quad (7.5)$$

若 $\gamma \geq 0$ 是一个很小的步长，就有 $f(\mathbf{x}_1) \leq f(\mathbf{x}_0)$ 。注意我们在梯度的部分使用了转置记号，这是因为我们在本书中默认梯度时行向量——如果不转置的话维度对不上。

有了这个发现，我们就能提出一个简单的梯度下降算法：我们想要找到一个函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}, \mathbf{x} \mapsto f(\mathbf{x})$ 的局部最优解 $f(\mathbf{x}_*)$ ，我们从一个初始估计 \mathbf{x}_0 开始，然后按照下面的更新规则不断迭代

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \gamma_i [(\nabla f)(\mathbf{x}_i)]^{\top} \quad (7.6)$$

假设我们每次迭代选择的步长足够合适，我们得到的序列就是一个下降的“链”：
 $f(\mathbf{x}_0) \geq f(\mathbf{x}_1) \geq \dots$ 它最终会趋于函数的局部最小值。



示例 7.1 考虑下面的二维二次函数

$$f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^\top \begin{bmatrix} 2 & 1 \\ 1 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 5 \\ 3 \end{bmatrix}^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (7.7)$$

它对 \mathbf{x} 的梯度是

$$\nabla f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^\top \begin{bmatrix} 2 & 1 \\ 1 & 20 \end{bmatrix} - \begin{bmatrix} 5 \\ 3 \end{bmatrix}^\top \quad (7.8)$$

如图 7.3 所示，我们从初始估计 $\mathbf{x}_0 = [-3, -1]^\top$ 开始用公式 (7.6) 不断迭代，以得到一个收敛于函数最小值的估计值序列。可见 \mathbf{x}_0 处的负梯度指向右上方，从而得到第二个估计 $\mathbf{x}_1 = [-1.98, 1.21]^\top$ （令 $\gamma = 0.085$ ，并将 \mathbf{x}_0 代入 (7.8)）。再迭代一次，我们得到 $\mathbf{x}_2 = [-1.32, -0.42]^\top$ ，以此类推。

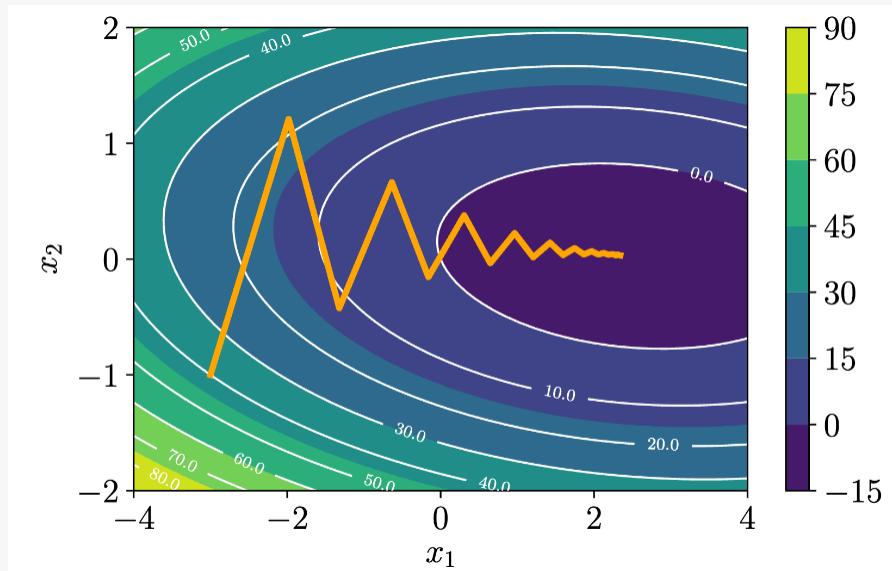


图 7.3 梯度下降算法的示例

注释 梯度下降算法趋近局部最小值的速度可以很慢，它的渐近收敛速度弱于很多其他算法。在面临一些性质不甚好的凸函数时，我们可以想象一个从很长但很窄的斜坡滚下的球：梯度下降的更新轨迹将会是像图 7.3 那样的锯齿形，每次更新的方向甚至会与该点与局部最小值点的直接连线几乎垂直。



7.1.1 步长（学习率）

前文提到，步长大小在梯度下降算法中十分重要：如果步长太小，梯度下降的速度会很慢；如果步长太大，梯度下降算法有可能射出原本的“峡谷”区域，难以收敛，甚至发散。解决方法之一——动量法是通过平滑不稳定的更新行为并抑制更新中的震荡现象的方法，我们将在下一节介绍它。

另一种解决方法是所谓 **自适应梯度法**。它们在每次梯度更新时都会根据函数在局部的行为对梯度进行缩放。下面是两个简单的启发方法 (Toussaint, 2012)

- 梯度更新后函数值变大了，这说明步长太大走得太远。回退这一步然后选一个更小的步长
- 梯度更新后函数值变小了，说明还可以走更远，因此可以尝试更大的步长 虽然“回退”这个做法看起来浪费资源，但这可以保证每次更新都会降低目标函数值。

示例 7.2（解线性方程） 假如我们用的范数是 Euclidean 范数，当我们解像 $\mathbf{Ax} = \mathbf{b}$ 这样的方程时，我们其实是通过找最小化

$$\|\mathbf{Ax} - \mathbf{b}\|^2 = (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) \quad (7.9)$$

以找到 $\mathbf{Ax} - \mathbf{b} = \mathbf{0}$ 的近似解 \mathbf{x}_* 来完成的。公式 (7.9) 对 \mathbf{x} 的梯度是

$$\nabla_{\mathbf{x}} = 2(\mathbf{Ax} - \mathbf{b})^\top \mathbf{A}, \quad (7.10)$$

我们可以用它直接导出梯度下降算法。但对于这个例子本身，我们有一个解析解——令梯度为零就可得到。我们将在第九章介绍更多求解平方损失的内容。

注释 用上述方法解 $\mathbf{Ax} = \mathbf{b}$ 形式的方程在某些情况下并不高效。梯度下降算法的收敛速度取决于矩阵的 **条件数** $\kappa = \frac{\sigma(\mathbf{A})_{\max}}{\sigma(\mathbf{A})_{\min}}$ ，它的值为矩阵 \mathbf{A} 的最大奇异值（见 4.5 节）和最小奇异值之比。换句话说，条件数刻画了目标函数最陡峭的方向和最平缓方向的“差距”。这和我们之前提到的情形相似：窄且长的“峡谷”对应着高的条件数：沿着峡谷行进的方向坡度平缓，而垂直于它的方向坡度陡峭。实际操作中我们不会直接求解 $\mathbf{Ax} = \mathbf{b}$ ，而是转而求



解 $\mathbf{P}^{-1}(\mathbf{A}\mathbf{x} - \mathbf{b}) = 0$ ，其中 \mathbf{P} 称为 **预条件子**，它可以降低新得到的线性方程系数矩阵的条件数，且 \mathbf{P} 本身需要容易得到。更多信息请参见 Boyd and Vandenberghe (2004, 第九章)。

7.1.2 动量梯度下降

如图 7.3 所示，如果优化曲面的曲率使得某些区域的性质不好，梯度下降的收敛速度可能会非常慢。曲率使得梯度下降更新在“峡谷”两侧跳跃，只能一小步一小步地接近最优值。为提高收敛性，我们可以赋予梯度下降一些“记忆”。

注释 Goh (2017) 撰写了一篇关于动量梯度下降的直观博客文章。

动量梯度下降 (Rumelhart et al., 1986) 是一种引入与上一次迭代的相关项的方法。这种记忆可以抑制振荡并使得梯度更新更加平滑。我们像之前一样考虑一个很重的滚动的球，动量项就模拟了它的惯性——很难轻易改变运动方向。这个方法也同时通过记忆梯度的更新以实现移动平均。具体而言，基于动量的方法会储存第 i 次迭代的更新 $\Delta\mathbf{x}_i$ ，然后加在第 $i + 1$ 次的梯度更新上；这相当于将第 i 次迭代和第 $i + 1$ 次迭代中得到的梯度做线性组合：

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \gamma_i [(\nabla f)(\mathbf{x}_i)]^\top + \alpha \Delta\mathbf{x}_i \quad (7.11)$$

$$\Delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_{i-1} = \alpha \Delta\mathbf{x}_{i-1} - \gamma_{i-1} [(\nabla f)(\mathbf{x}_{i-1})]^\top, \quad (7.12)$$

其中 $\alpha \in [0, 1]$ 。有时我们只知道梯度的一个估计值，此时上面的动量项作为移动平均会帮我们抹除梯度估计中的噪声，因此十分有用。下面介绍的随机梯度下降就是一个动量法大展身手的例子。

7.1.3 随机梯度下降

精确地计算梯度十分费时费力，但我们往往可以找到更快速地计算梯度估计值的方法——只要我们估计的梯度和真实的梯度方向大致相同。**随机梯度下降 (SGD)** 是一种用于最小化可被写成一系列可微函数的目标函数，并给出梯度的随机估计的梯度下降算法。“随机”一词指的是我们每次更新不知道梯度的真实值，而只有一个带噪声的

梯度估计值。如果限制梯度估计值的分布，在理论上我们依然可以保证 SGD 的收敛性。

在机器学习中，给定 $n = 1, \dots, N$ 个数据点，我们通常将每个数据的损失 L_n 的求和作为目标函数：

$$L(\boldsymbol{\theta}) = \sum_{n=1}^N L_n(\boldsymbol{\theta}) \quad (7.13)$$

其中 $\boldsymbol{\theta}$ 是我们关心的参数向量——我们要找出最小化 L 的参数 $\boldsymbol{\theta}$ 。第九章中我们将见到来自回归问题的 **负对数似然函数**，它是每个数据的负对数似然函数的求和：

$$L(\boldsymbol{\theta}) = -\sum_{n=1}^N \log p(y_n | \mathbf{x}_n, \boldsymbol{\theta}) \quad (7.14)$$

其中 $\mathbf{x}_n \in \mathbb{R}^D$ 是训练中的输入数据， y_n 是训练中的目标数据， $\boldsymbol{\theta}$ 是回归模型的参数。

前文提到，经典的梯度下降是一个“整批”的优化方法，这是说每次我们都要选一个合适的 γ_i ，并用**所有的**训练集来完成下面的迭代：

$$\boldsymbol{\theta}_{i+1} + \boldsymbol{\theta}_i = \gamma_i [\nabla L(\boldsymbol{\theta}_i)]^\top = \boldsymbol{\theta}_i - \gamma_i \sum_{n=1}^N [\nabla L_n(\boldsymbol{\theta}_i)]^\top \quad (7.15)$$

计算上面对所有 L_n 的梯度之和是个大工程。当训练集很大，或是没有显式的梯度可以求解的时候，这么做显然是极其昂贵的。

考虑 (7.15) 中的一项 $\sum_{n=1}^N [\nabla L_n(\boldsymbol{\theta})]$ ，我们可以通过只算一小部分 L_n 的梯度之和来降低计算成本。相较于用上全部 $L_n, n = 1, \dots, N$ 的经典梯度下降算法，我们只选择小部分 L_n ，这样我们就得到了**小批次梯度下降**；该算法最极端的情况是每次只考虑一个 L_n 。我们这么做是有道理的：我们只需要拿到一个对真实梯度的**无偏估**

计，而公式 (7.15) 中的 $\sum_{n=1}^N [\nabla L_n(\boldsymbol{\theta})]$ 事实上就是对梯度期望值(见 6.4.1) 的经验估计，因此任何对梯度的无偏估计都可以拿来用。不论我们的小批次中的数据量是多少，它都是对梯度的无偏估计，SGD 也总会收敛。



注释 在相对较弱的假设下，如果学习率以适当的幅度逐步降低，SGD 几乎必然 (**almost surely**) 收敛到局部最优解。 (Bottu, 1998)

译者注 几乎必然是一个专有名词，它属于概率论，指的是事件发生的概率为 1，或 Lebesgue 测度为 1；有时也简记为 a.s.

我们为什么要估计梯度的值呢？主要的原因是实践中的 CPU 和 GPU 的存储空间或是计算时间有限。我们可以考虑不同大小的批次。较大的批次不但可以利用高效的矩阵算法快速计算结果，还会给出梯度更加精确的估计，降低了参数更新的方差，算法的收敛也会更稳定。相比之下较小的批次可以更快的算出，但牺牲了估计的精确性，这可能会让我们陷入更差的局部最优而无法脱离。

机器学习中，我们用优化算法解决我们的短期目标：训练集上的目标函数，以期完成增强模型泛化性能的长期目标（第八章）。机器学习实践中也不需要对目标函数的最小值有多么精确地估计，因此类似上文中的小批量算法被大量使用，且在大规模机器学习问题 (Bottou et al., 2018) 例如训练神经网络为几十万张图片进行分类 (Dean et al., 2012)、主题模型 (Hoffman et al., 2013)、强化学习 (Mnih et al., 2015) 或是训练大规模 Gauss 过程模型 (Hensman et al, 2013; Gal et al, 2014) 中效果拔群。

下一章节 >

7.2 约束优化和 Lagrange 乘子



7.2 约束优化和 Lagrange 乘子

在前一节中，我们讨论了如何求解函数的最小化问题：

$$\min_{\mathbf{x}} f(\mathbf{x}), \quad (7.16)$$

其中 $f : \mathbb{R}^D \rightarrow \mathbb{R}$ 。但在本节中，我们得面对额外的“约束条件”，具体来说，对于实值函数 $g_i : \mathbb{R}^D \rightarrow \mathbb{R}$ ($i = 1, \dots, m$)，我们考虑如下的约束优化问题（如图 7.4）：

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) \\ & \text{subject to } g_i(\mathbf{x}) \leq 0 \quad \text{for all } i = 1, \dots, m \end{aligned} \quad (7.17)$$

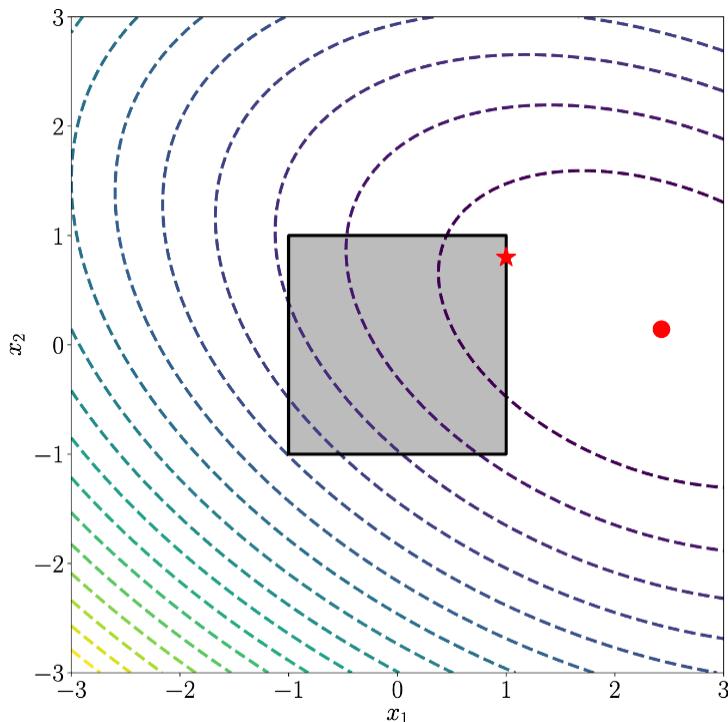


图7.4 约束优化图示

这里有个值得注意的细节：函数 f 和 g_i 在一般情况下可能非凸（non-convex），不过别急，我们将在下一节讨论凸优化这个“乖孩子”。

一种直观但不太实用的方法是使用 **示性函数**（**indicator function**）将约束问题 (7.17) 转化为无约束形式：

$$J(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^m \mathbf{1}[g_i(\mathbf{x})] \quad (7.18)$$

其中

$$\mathbf{1}(z) = \begin{cases} 0 & z \leq 0 \\ \infty & \text{otherwise} \end{cases}. \quad (7.19)$$

这招儿就像给违反约束的行为判了“无期徒刑”，理论上能给出相同解，但实际优化起来十分困难。我们可以用Lagrange 乘数法（Lagrange multipliers）解决这个问题：它的妙招是把阶跃函数松弛为线性函数。

我们为问题 (7.17) 引入 **Lagrange 函数 (Lagrangian)**，通过Lagrange 乘数 $\lambda_i \geq 0$ 将每个不等式约束松弛化（Boyd and Vandenberghe, 2004, 第四章）：

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i g_i(\mathbf{x}) \quad (7.20a)$$

$$= f(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{g}(\mathbf{x}) \quad (7.20b)$$

这里，我们把所有约束 $g_i(\mathbf{x})$ 打包成一个向量 $\mathbf{g}(\mathbf{x})$ ，所有乘数也塞进向量，得到 $\boldsymbol{\lambda} \in \mathbb{R}^m$ 。

现在我们引入 **Lagrange 对偶性 (Lagrangian duality)**。优化中的对偶思想，本质是把原变量（primal variables） \mathbf{x} 的问题，转换成另一组对偶变量（dual variables） $\boldsymbol{\lambda}$ 的问题。本节我们聚焦Lagrange 对偶，除此之外我们将在 7.3.3 节介绍 Legendre-Fenchel 对偶。

定义 7.1 我们称 (7.17) 中的问题

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) \\ & \text{subject to } g_i(\mathbf{x}) \leq 0 \quad \text{for all } i = 1, \dots, m \end{aligned} \quad (7.21)$$

为原问题（primal problem），对应原变量 \mathbf{x} 。其关联的**Lagrange 对偶问题 (Lagrangian dual problem)** 是

$$\begin{aligned} & \min_{\boldsymbol{\lambda} \in \mathbb{R}^m} \mathfrak{D}(\boldsymbol{\lambda}) \\ & \text{subject to } \boldsymbol{\lambda} \geq 0. \end{aligned} \quad (7.22)$$

其中 λ 是对偶变量, $\mathcal{D}(\lambda) = \min_{x \in \mathbb{R}^d} \mathcal{L}(x, \lambda)$ 。

注释

在定义 7.1 的讨论中, 我们用到两个独立有趣的概念 (Boyd and Vandenberghe, 2004)

第一个概念叫做 极小极大不等式 (**minimax inequality**) : 对任意双变量函数 $\varphi(x, y)$, 有

$$\max_y \min_x \varphi(x, y) \leq \min_x \max_y \varphi(x, y). \quad (7.23)$$

可以考虑下面的不等式来证明

$$\forall x, y \quad \min_x \varphi(x, y) \leq \max_y \varphi(x, y). \quad (7.24)$$

显然, 左边的式子对 y 取 \max 就对应 (7.23) 的左边; 类似地操作我们也能得到右边。

第二个概念是 弱对偶性 (**weak duality**), 这是说我们在 (7.23) 证明了了的 "原问题值总大于等于对偶值", 更多细节见 (7.27)。

回忆一下, (7.18) 中的 $J(x)$ 与 Lagrange 函数的关键区别, 是我们把指示函数松弛成了线性函数。因此, 当 $\lambda \geq 0$ 时, Lagrange $\mathcal{L}(x, \lambda)$ 是 $J(x)$ 的下界。于是, $\mathcal{L}(x, \lambda)$ 对 λ 的最大化给出

$$J(x) = \max_{\lambda \geq 0} \mathcal{L}(x, \lambda). \quad (7.25)$$

同时原问题是最小化 $J(x)$

$$\min_{x \in \mathbb{R}^d} \max_{\lambda \geq 0} \mathcal{L}(x, \lambda). \quad (7.26)$$

由极小极大不等式 (7.23), 交换最小和最大顺序会得到更小值, 也就是所谓的弱对偶性:

$$\min_{x \in \mathbb{R}^d} \max_{\lambda \geq 0} \mathcal{L}(x, \lambda) \geq \max_{\lambda \geq 0} \min_{x \in \mathbb{R}^d} \mathcal{L}(x, \lambda). \quad (7.27)$$

其中右侧里面正是对偶目标函数 $\mathfrak{D}(\boldsymbol{\lambda})$ 。

与原优化问题（带约束）相比， $\min_{\mathbf{x} \in \mathbb{R}^d} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ 对给定 $\boldsymbol{\lambda}$ 是无约束问题。如果这个子问题容易求解，那整体问题就变简单了！观察 (7.20b)， $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ 关于 $\boldsymbol{\lambda}$ 是仿射 (affine) 的，因此 $\min_{\mathbf{x} \in \mathbb{R}^d} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ 是 $\boldsymbol{\lambda}$ 的仿射函数的逐点最小值，故 $\mathfrak{D}(\boldsymbol{\lambda})$ 是凹函数——即使 $f(\cdot)$ 和 $g_i(\cdot)$ 非凸。外部最大化问题（对 $\boldsymbol{\lambda}$ ）是凹函数的最大化，可高效求解。

假设 $f(\cdot)$ 和 $g_i(\cdot)$ 可微，我们通过微分 Lagrange 函数求解对偶问题：对 \mathbf{x} 求导、设导数为零、解最优值。第 7.3.1 和 7.3.2 节将讨论两个具体例子 (f 和 g_i 为凸时)。

注释（等式约束）考虑 (7.17) 添加等式约束

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) \\ \text{subject to } & g_i(\mathbf{x}) \leq 0 \quad \forall i = 1, \dots, m \\ & h_j(\mathbf{x}) = 0 \quad \forall j = 1, \dots, n. \end{aligned} \tag{7.28}$$

我们可以用两个不等式约束模拟等式约束：对每个 $h_j(\mathbf{x}) = 0$ ，等价替换为 $h_j(\mathbf{x}) \leq 0$ 和 $h_j(\mathbf{x}) \geq 0$ 。结果 Lagrange 乘数将无约束。因此，在 (7.28) 中，我们仅约束不等式乘数为非负，而等式乘数则没有约束。

< 上一章节

下一章节 >

7.1 基于梯度下降的优化

7.3 凸优化



7.3 凸优化

我们将目光聚焦于一类能保证全局最优解的特殊优化问题。当目标函数 $f(\cdot)$ 是凸函数，且约束函数 $g(\cdot)$ 和 $h(\cdot)$ 定义的集合为凸集时，这类问题称为**凸优化问题**。凸优化问题具有**强对偶性**：对偶问题的最优解与原问题完全一致。虽然机器学习文献常模糊凸函数与凸集的界限，但上下文通常能提供明确指引。

定义 7.2 (凸集) 若集合 \mathcal{C} 满足：对任意 $x, y \in \mathcal{C}$ 和标量 $\theta \in [0, 1]$ ，有

$$\theta x + (1 - \theta)y \in \mathcal{C}. \quad (7.29)$$

则称 \mathcal{C} 为凸集。

凸集中两点之间的线段总是位于凸集中。下图给出了凸集的一个典型例子和反例。

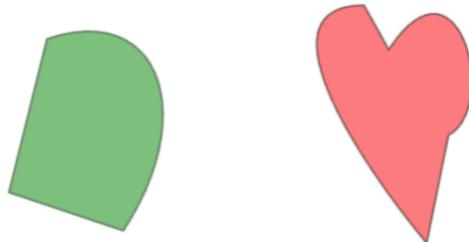


图 7.5, 图 7.6 凸集（左）和非凸集合（右）

凸函数定义和凸集很像，它的定义是函数上两点的连线一定位于函数曲线的上方。图 7.2 画了一个非凸函数，图 7.3 画的是凸函数。图 7.7 中也是一个凸函数。

定义 7.3 (凸函数) 考虑函数 $f : \mathbb{R}^D \rightarrow \mathbb{R}$ ，且 f 的定义域为凸集。则它被称为是**凸函数**如果对定义域中的所有 \mathbf{x}, \mathbf{y} 和任意标量 $0 \leq \theta \leq 1$ ，都有

$$f(\theta \mathbf{x} + (1 - \theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1 - \theta)f(\mathbf{y}). \quad (7.30)$$



注释 一个 凹函数 一定是某个 凸函数 的负数

公式 (7.28) 中的约束通过限制约束函数 $f(\cdot)$ 和 $g(\cdot)$ 的标量函数值，最终得到一个集合——可行域。凸函数和凸集之间的另一种关系是考虑通过“填充”凸函数得到的集合。凸函数是一个“碗状物体”，我们想象往里面倒水来填满它。这个填满的集合被称为凸函数的**上镜图 (epigraph)**，它也是一个凸集。

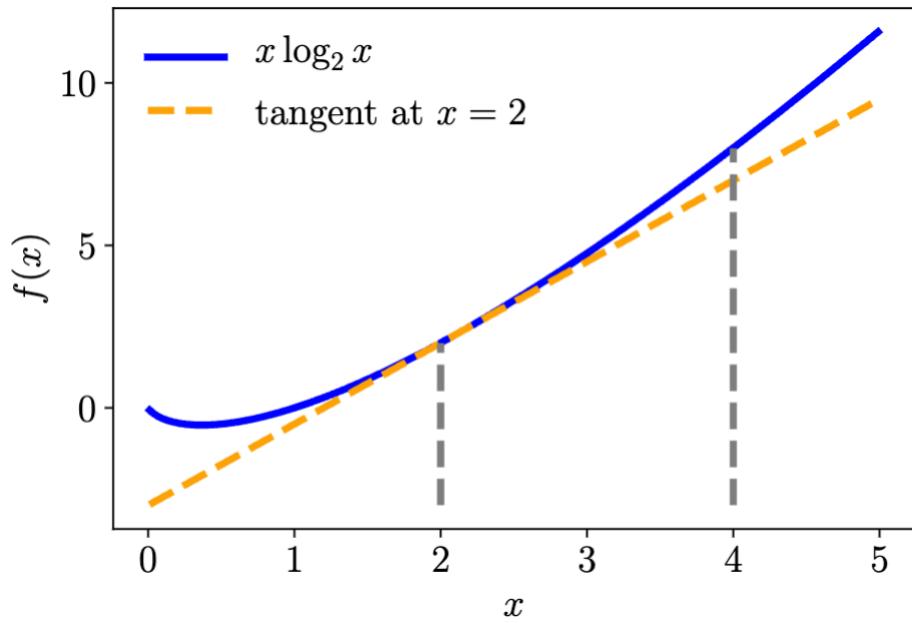
如果函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 是可微的，我们还可以根据其梯度 $\nabla_{\mathbf{x}} f(\mathbf{x})$ (见 5.2) 来判断其凸性。这样的函数是凸的，当且仅当对任意定义域中的 \mathbf{x} 和 \mathbf{y} ，都有

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla_{\mathbf{x}} f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}). \quad (7.31)$$

进一步地，如果我们知道 f 是二阶可微的，也就是在定义域中的每一点都存在 Hesse 矩阵 (5.147)，则该函数是凸的当且仅当 $\nabla_{\mathbf{x}}^2 f(\mathbf{x})$ 是半正定的 (Boyd and Vandenberghe, 2004)。

示例 7.3 (熵)

负熵函数 $f(x) = x \log_2 x$ 在 $x > 0$ 上是凸函数，如图 7.8 所示。为了说明先前提到的凸函数的定义，我们选择 $x = 2$ 和 $x = 4$ 两个位置检查。需要注意的是，要证明该函数的凸性，只选择两个点不够，我们要检查所有的 $x \in \mathbb{R}$ 。



让我们回忆定义 7.3，考虑它们的中间位置 ($\theta = 0.5$)，那么公式 (7.30) 的左边是 $f(0.5 \cdot 2 + 0.5 \cdot 4) = 3 \log_2 3 \approx 4.75$ ，右边是 $0.5(2 \log_2 2) + 0.5(4 \log_2 4) = 1 + 4 = 5$ ，这符合凸函数的定义。由于 $f(x)$ 是可微的，我们也可以使用公式 (7.31) 对其图形进行判定。首先我们计算 $f(x)$ 的导数：

$$\nabla_x \log(x \log_2 x) = 1 \cdot \log_2 x + x \cdot \frac{1}{x \log_e 2} = \log_2 x + \frac{1}{\log_e 2} \quad (7.32)$$

我们同样使用 $x = 2$ 和 $x = 4$ 两点，公式 (7.31) 的左侧是 $f(4) = 8$ ，右侧是

$$f(\mathbf{x}) + \nabla_{\mathbf{x}}^\top (\mathbf{y} - \mathbf{x}) = f(2) + \nabla f(2) \cdot (4 - 2) \quad (7.33a)$$

$$= 2 + \left(1 + \frac{1}{\log_e 2}\right) \cdot 2 \approx \frac{6}{9} \quad (7.33b)$$

我们可以通过多种方法检查一个函数是否是凸函数。实际操作中我们通常通过保持凸性的变换来检查某个函数或集合是不是凸的。尽管细节有很大不同，但这仍然是我们在第二章中为线性空间引入的闭包思想。



示例 7.4 (凸函数的非负线性组合) 若干凸函数的非负线性组合还是凸函数。我们首先观察到, 如果 f 是凸函数, 那么对于任意非负实数 α , 函数 αf 也是凸的。这个证明很简单, 只需要将公式 (7.3) 的左右两侧都乘上 α 即可。考虑两个凸函数 f_1 和 f_2 , 根据凸函数定义我们有

$$f_1(\theta \mathbf{x} + (1 - \theta)\mathbf{y}) \leq \theta f_1(\mathbf{x}) + (1 - \theta)f_1(\mathbf{y}) \quad (7.34)$$

$$f_2(\theta \mathbf{x} + (1 - \theta)\mathbf{y}) \leq \theta f_2(\mathbf{x}) + (1 - \theta)f_2(\mathbf{y}). \quad (7.35)$$

两式相加, 有

$$\begin{aligned} & f_1(\theta \mathbf{x} + (1 - \theta)\mathbf{y}) + f_2(\theta \mathbf{x} + (1 - \theta)\mathbf{y}) \\ & \leq \theta f_1(\mathbf{x}) + (1 - \theta)f_1(\mathbf{y}) + \theta f_2(\mathbf{x}) + (1 - \theta)f_2(\mathbf{y}) \end{aligned} \quad (7.36)$$

其中不等式右边还可以进一步整理为

$$\theta [f_1(\mathbf{x}) + f_2(\mathbf{x})] + (1 - \theta) [f_1(\mathbf{y}) + f_2(\mathbf{y})], \quad (7.37)$$

这样我们就证明了 $f_1 + f_2$ 是凸的。结合这两个事实, 我们有对于任意的 $\alpha, \beta \geq 0$, $\alpha f_1 + \beta f_2$ 是凸函数。对于三个及以上函数的非负线性组合, 证明方法类似。

注释 公式 (7.30) 中的不等式又称为 **Jensen 不等式**。事实上, 这一整类用于求凸函数非负加权和的不等式都称为 Jensen 不等式。

总的来说, 被称为 **凸优化** 的约束优化问题的长相如下:

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) \\ & \text{subject to } g_i(\mathbf{x}) \leq 0 \quad \forall i = 1, \dots, m \\ & \quad h_j(\mathbf{x}) = 0 \quad \forall j = 1, \dots, n, \end{aligned} \quad (7.38)$$

其中 $f(\mathbf{x})$ 和所有的 $g_i(\mathbf{x})$ 都是凸函数, 所有的 $h_j(\mathbf{x}) = 0$ 都对应着凸集。下面的内容我们将讨论两个常用并以研究透了的凸优化问题。

7.3.1 线性规划



我们首先考虑所有函数都是线性函数这一特殊情况：

$$\begin{aligned} & \min_{\boldsymbol{x} \in \mathbb{R}^d} \boldsymbol{c}^\top \boldsymbol{x} \\ & \text{subject to } \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \end{aligned} \tag{7.39}$$

其中 $\boldsymbol{A} \in \mathbb{R}^{m \times d}$, $\boldsymbol{b} \in \mathbb{R}^m$ 。这样的问题称为 **线性规划**。

注释 线性规划是工业中最常用的一类方法

它有 d 个变量和 m 个线性约束，它的 Lagrange 函数是

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\lambda}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}), \tag{7.40}$$

其中 $\boldsymbol{\lambda} \in \mathbb{R}^m$ 是非负的 Lagrange 乘子组成的向量，稍微整理一下，得到

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = (\boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\lambda})^\top \boldsymbol{x} - \boldsymbol{\lambda}^\top \boldsymbol{b}. \tag{7.41}$$

求 $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ 对 \boldsymbol{x} 的导数，并令其为零，我们得到

$$\boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\lambda} = \mathbf{0}. \tag{7.42}$$

因此我们得到对偶 Lagrange 函数 $\mathfrak{D}(\boldsymbol{\lambda}) = -\boldsymbol{\lambda}^\top \boldsymbol{b}$ ，我们需要最大化 $\mathfrak{D}(\boldsymbol{\lambda})$ 。除了前文中 $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda})$ 要为零，我们还需要保持 $\boldsymbol{\lambda} \geq \mathbf{0}$ ，这就得到下面的对偶优化问题

$$\begin{aligned} & \max_{\boldsymbol{\lambda} \in \mathbb{R}^m} -\boldsymbol{b}^\top \boldsymbol{\lambda} \\ & \text{subject to } \boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\lambda} = \mathbf{0} \\ & \quad \boldsymbol{\lambda} \geq \mathbf{0} \end{aligned} \tag{7.43}$$

注解 一般主问题是一个最小化的问题，对偶问题则是一个最大化的问题。

这还是一个线性优化问题，但变元的数量是 m 。我们可以依据实际情况选择是解原问题 (7.39) 还是解对偶问题 (7.43)，就看是原问题中的变元数量 d 更小还是原问题

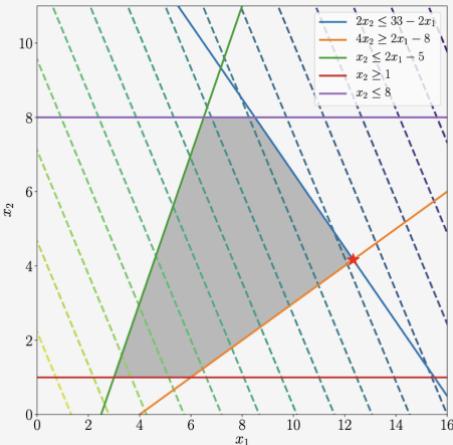
中约束数量 m 更小，哪个小选哪个。



示例 7.5 (线性规划) 考虑下面的二变元线性规划问题

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^2} \quad & \begin{bmatrix} 5 \\ 3 \end{bmatrix}^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ \text{subject to} \quad & \begin{bmatrix} 2 & 2 \\ 2 & -4 \\ -2 & 1 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 33 \\ 8 \\ 5 \\ -1 \\ 8 \end{bmatrix} \end{aligned} \quad (7.44)$$

如图 7.9。



由图可知目标函数是线性的——它的等高线是直线。问题的约束集合在图中由不同颜色的实直线表示，可行域由灰色阴影表示，这意味着最优解（红色五角星）必须在灰色阴影区域（在此例中，也包括其边缘）。

7.3.2 二次规划

现在考虑目标函数是凸的二次函数，而约束是仿射函数的情形：

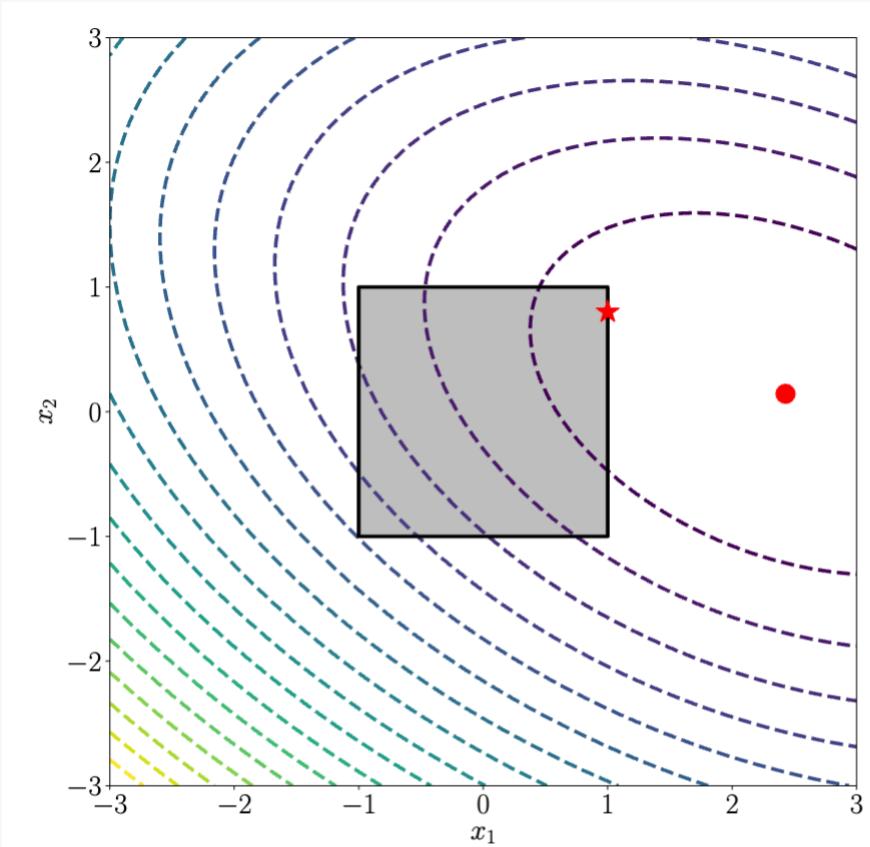
$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \quad & \frac{1}{2} \boldsymbol{x}^\top \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{A} \boldsymbol{x} \leq \boldsymbol{b}, \end{aligned} \quad (7.45)$$

其中 $\mathbf{A} \in \mathbb{R}^{m \times d}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^d$ 。目标函数中的矩阵 $\mathbf{Q} \in \mathbb{R}^{d \times d}$ 是正定的，因此目标函数是凸的。这样的问题叫做 **二次规划**。它有 d 个变量， m 个线性约束。

示例 7.6 (二次规划) 考虑下面的二变元二次规划问题

$$\min_{x \in \mathbb{R}^2} \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^\top \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 5 \\ 3 \end{bmatrix}^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (7.46)$$

$$\text{subject to } \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leqslant \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (7.47)$$



由图可知，目标函数是二次的，矩阵 \mathbf{Q} 是半正定的，因此我们看到的目标函数等高线是一系列椭圆。可行域是灰色区域，最优解由红色五角星表示。

二次规划的 Lagrange 函数整理一下之后是

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) &= \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{c}^\top \mathbf{x} + \boldsymbol{\lambda}^\top (\mathbf{A} \mathbf{x} - \mathbf{b}) \\ &= \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + (\mathbf{c} + \mathbf{A}^\top \boldsymbol{\lambda})^\top \mathbf{x} - \boldsymbol{\lambda}^\top \mathbf{b},\end{aligned}$$

(7.48a), (7.48b)

求它对 \mathbf{x} 的导数并令其为零，我们有

$$\mathbf{Q} \mathbf{x} + (\mathbf{c} + \mathbf{A}^\top \boldsymbol{\lambda}) = \mathbf{0}. \quad (7.49)$$

假设 \mathbf{Q} 是可逆的，得到

$$\mathbf{x} = -\mathbf{Q}^{-1}(\mathbf{c} + \mathbf{A}^\top \boldsymbol{\lambda}). \quad (7.50)$$

把 (7.50) 代入最初的 Lagrange 函数 $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ ，我们得到 Lagrange 对偶函数

$$\mathfrak{D}(\boldsymbol{\lambda}) = \frac{1}{2} (\mathbf{c} + \mathbf{A}^\top \boldsymbol{\lambda})^\top \mathbf{Q}^{-1} (\mathbf{c} + \mathbf{A}^\top \boldsymbol{\lambda}) - \boldsymbol{\lambda}^\top \mathbf{b}. \quad (7.51)$$

于是二次规划的对偶优化问题就是

$$\begin{aligned}\max_{\boldsymbol{\lambda} \in \mathbb{R}^m} \quad & \frac{1}{2} (\mathbf{c} + \mathbf{A}^\top \boldsymbol{\lambda})^\top \mathbf{Q}^{-1} (\mathbf{c} + \mathbf{A}^\top \boldsymbol{\lambda}) - \boldsymbol{\lambda}^\top \mathbf{b} \\ \text{subject to} \quad & \boldsymbol{\lambda} \geq \mathbf{0}.\end{aligned} \quad (7.52)$$

我们将在第十二章的机器学习内容中再次见到二次规划。

7.3.3 Legendre-Fenchel 变换和凸共轭

让我们不考虑约束，重新回顾 7.2 节中的对偶概念。关于凸集的一个有用事实是，它可以用它的支撑超平面等价地描述。如果一个超平面与凸集相交，并且凸集只包含在它的一侧，则该超平面称为凸集的支撑超平面。回想一下，我们可以“填充”凸函数来获得上镜图，它是一个凸集。因此，我们也可以用它们的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，实际上是该函数在该点的切线。回想一下，函数 $f(\mathbf{x})$ 在给定点 \mathbf{x}_0 的切线是该函数在该点的梯度的求值

$$\left. \frac{df(\mathbf{x})}{d\mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_0} \quad \text{。总而言之，由于凸集可以用其支撑超平面等效地描述，因此凸函数}$$

也可以用其梯度的函数等效地描述。**Legendre 变换**形式化地表达了这一概念。



注解 物理系学生常常在学习经典力学中的 Lagrange 量和 Hamilton 量的时候接触 Legendre 变换的

我们从最一般的定义开始，但它的形式有些违反直觉。我们先来看一些特殊情况，以便将定义与上一段描述的直觉联系起来。Legendre-Fenchel 变换是从凸可微函数 $f(\mathbf{x})$ 到依赖于切线 $s(\mathbf{x}) = \nabla_{\mathbf{x}} f(\mathbf{x})$ 的函数的变换（在傅里叶变换的意义上）。值得强调的是，这是函数 $f(\cdot)$ 的变换，而不是变量 \mathbf{x} 或在 \mathbf{x} 处求值的函数的变换。Legendre-Fenchel 变换也称为凸共轭（关于凸共轭的原因，我们很快就会看到），并且与对偶性密切相关（Hiriart-Urruty and Lemaréchal, 2001, 第五章）。

定义 7.4 (凸共轭) 函数 $f : \mathbb{R}^D \rightarrow \mathbb{R}$ 的 **凸共轭** 是

$$f^*(\mathbf{s}) = \sup_{\mathbf{x} \in \mathbb{R}^D} [\langle \mathbf{s}, \mathbf{x} \rangle - f(\mathbf{x})]. \quad (7.53)$$

注意下文中提到的凸共轭并不需要函数 f 是凸的或是可微的。定义 7.4 中，我们用的是抽象的内积记号（见 3.2），但下文中我们将继续使用有限维向量之间的标准内积 ($\langle \mathbf{s}, \mathbf{x} \rangle = \mathbf{s}^\top \mathbf{x}$)，以避免一些不必要的麻烦

注解 画图能帮我们更好理解凸共轭的定义

为了从几何角度理解定义 7.4 的内容，考虑一个简单的一元可微的凸函数，例如 $f(x) = x^2$ 。注意我们考虑的是一元函数，超平面就是一条直线。考虑直线 $y = sx + c$ ——我们可以用支撑超平面描述凸函数，因此让我们尝试用支撑超平面来描述函数 $f(x)$ 。固定直线的梯度 $s \in \mathbb{R}$ ，对于 f 的图上的每个点 $(x_0, f(x_0))$ ，找到 c 的最小值，使直线仍然经过 $(x_0, f(x_0))$ 相交。请注意， c 的最小值是斜率为 s 的直线刚好和函数 $f(x) = x^2$ 相切的位置。通过 $(x_0, f(x_0))$ 且梯度为 s 的直线由

$$y - f(x_0) = s(x - x_0). \quad (7.54)$$

给出。这条直线的 y 轴截距为 $-sx_0 + f(x_0)$ 。因此，当 $y = sx + c$ 与 f 的图像相交时， c 的最小值为

$$\inf_{x_0} \left[-sx_0 + f(x_0) \right]. \quad (7.55)$$

按照惯例，前述凸共轭定义为其负值。本段的推理并不依赖于我们选择一维凸可微函数这一事实，并且对于 $f : \mathbb{R}^D \rightarrow \mathbb{R}$ 成立，它们是非凸且不可微的。

注解 像 $f(x) = x^2$ 这样的可微凸函数是一个很好的特殊情况，我们不需要求上确界，且每个可微的凸函数和它的 Legendre 变换一一对应。让我们一步一步导出这个结果。考虑可微凸函数 f ，和 $(x_0, f(x_0))$ 处的切线

$$f(x_0) = sx_0 + c. \quad (7.56)$$

回忆可微凸函数 f 和其梯度 $\nabla_x f(x)$ 的性质，我们有 $x = \nabla_x f(x_0)$ ，整理上式，得到

$$-c = sx_0 - f(x_0). \quad (7.57)$$

注意 c 随着 x_0 （也即 s ）的变化而变化，我们可以将其写为

$$f^*(s) := sx_0 - f(x_0). \quad (7.58)$$

将 (7.58) 与定义 7.4 对比，容易发现前者是一个不带上确界的特殊情况。

凸共轭函数有不少良好的性质。例如对于凸函数，它的共轭的共轭是它本身。同样地， $f(x)$ 处的切线斜率是 s 而 $f^*(s)$ 处的斜率是 x 。下面的两个例子给出凸共轭在机器学习中的常见应用。

示例 7.7 (凸共轭) 为了展示凸共轭的应用，考虑下面的二次规划问题

$$f(\mathbf{y}) = \frac{\lambda}{2} \mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y} \quad (7.59)$$

其中 $\mathbf{K} \in \mathbb{R}^{n \times n}$ 是一个正定矩阵。我们定义主变量是 $\mathbf{y} \in \mathbb{R}^n$ ，对偶变量是 $\boldsymbol{\alpha} \in \mathbb{R}^n$ 。根据定义 7.4，我们有



$$f^*(\boldsymbol{\alpha}) = \sup_{\mathbf{y} \in \mathbb{R}^n} \left[\langle \mathbf{y}, \boldsymbol{\alpha} \rangle - \frac{\lambda}{2} \mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y} \right]. \quad (7.60)$$

由于该上确界中的函数是可微的，我们可以通过令其对 \mathbf{y} 的梯度

$$\frac{\partial [\langle \mathbf{y}, \boldsymbol{\alpha} \rangle - \frac{\lambda}{2} \mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}]}{\partial \mathbf{y}} = (\boldsymbol{\alpha} - \lambda \mathbf{K}^{-1} \mathbf{y})^\top \quad (7.61)$$

为零，也即当 $\mathbf{y} = \frac{1}{\lambda} \mathbf{K} \boldsymbol{\alpha}$ ，得到其最大值，也就是

$$f^*(\boldsymbol{\alpha}) = \frac{1}{\lambda} \boldsymbol{\alpha}^\top \mathbf{K}^{-1} \boldsymbol{\alpha} - \frac{\lambda}{2} \left(\frac{1}{\lambda} \mathbf{K} \boldsymbol{\alpha} \right)^\top \mathbf{K}^{-1} \left(\frac{1}{\lambda} \mathbf{K} \boldsymbol{\alpha} \right) = \frac{1}{2\lambda} \boldsymbol{\alpha}^\top (\mathbf{K} \boldsymbol{\alpha})$$

示例 7.8 机器学习中，我们常用一系列函数（例如每条训练数据的损失函数 $\ell : \mathbb{R} \rightarrow \mathbb{R}$ ）的和作为目标。下面我们推导损失函数 $\ell(\mathbf{t})$ 之和的凸共轭，这同时展示了凸共轭在向量变元函数情况下的应用。令 $\mathcal{L}(\mathbf{t}) = \sum_{i=1}^n \ell_i(t_i)$ ，于是

$$\mathcal{L}^*(\mathbf{z}) = \sup_{\mathbf{t} \in \mathbb{R}^n} \left[\langle \mathbf{z}, \mathbf{t} \rangle - \sum_{i=1}^n \ell_i(t_i) \right] \quad (7.63a)$$

$$= \sup_{\mathbf{t} \in \mathbb{R}^n} \sum_{i=1}^n [z_i t_i - \ell_i(t_i)] \quad \text{内积定义} \quad (7.63b)$$

$$= \sum_{i=1}^n \sup_{\mathbf{t} \in \mathbb{R}^n} [z_i t_i - \ell_i(t_i)] \quad (7.63c)$$

$$= \sum_{i=1}^n \ell_i^*(z_i) \quad \text{共轭定义} \quad (7.63d)$$

回忆在 7.2 节中，我们使用 Lagrange 乘子导出原问题的对偶优化问题。进一步地，凸优化问题具有强对偶性：对偶问题的解就是原问题的解。本节中介绍的 Legendre-Fenchel 变换也可以用来求对偶优化问题，特别地，当目标函数是可微且凸时，Legendre-Fenchel 变换中的上确界是唯一的。为了进一步说明这两个方法之间的联系，考虑下面带线性等式约束的凸优化问题。



示例 7.9 考虑凸函数 $f(\mathbf{x})$, $g(\mathbf{x})$, 实矩阵 \mathbf{A} , 并假设方程 $\mathbf{Ax} = \mathbf{y}$ 中的向量和矩阵形状匹配。于是

$$\min_{\mathbf{x}} f(\mathbf{Ax}) + g(\mathbf{x}) = \min_{\mathbf{Ax}=\mathbf{y}} f(\mathbf{y}) + g(\mathbf{x}). \quad (7.64)$$

引入约束 $\mathbf{Ax} = \mathbf{y}$ 和 Lagrange 乘子 \mathbf{u} , 有

$$\min_{\mathbf{Ax}=\mathbf{y}} f(\mathbf{y}) + g(\mathbf{x}) = \min_{\mathbf{x}, \mathbf{y}} \max_{\mathbf{u}} f(\mathbf{y}) + g(\mathbf{x}) + (\mathbf{Ax} - \mathbf{y})^\top \mathbf{u} \quad (7.65a)$$

$$= \max_{\mathbf{u}} \min_{\mathbf{x}, \mathbf{y}} f(\mathbf{y}) + g(\mathbf{x}) + (\mathbf{Ax} - \mathbf{y})^\top \mathbf{u} \quad (7.65b)$$

其中最后一步可以交换 \max 和 \min 是因为 $f(\mathbf{y})$ 和 $g(\mathbf{x})$ 是凸函数。展开点积这一项, 然后分开 \mathbf{x} 和 \mathbf{y} 的项, 得到

$$\max_{\mathbf{u}} \min_{\mathbf{x}, \mathbf{y}} f(\mathbf{y}) + g(\mathbf{x}) + (\mathbf{Ax} - \mathbf{y})^\top \mathbf{u} \quad (7.66a)$$

$$= \max_{\mathbf{u}} \left[\min_{\mathbf{y}} -\mathbf{y}^\top \mathbf{u} + f(\mathbf{y}) \right] + \left[\min_{\mathbf{x}} (\mathbf{Ax})^\top \mathbf{u} + g(\mathbf{x}) \right] \quad (7.66b)$$

$$= \max_{\mathbf{u}} \left[\min_{\mathbf{y}} -\mathbf{y}^\top \mathbf{u} + f(\mathbf{y}) \right] + \left[\min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{u} + g(\mathbf{x}) \right] \quad (7.66c)$$

回忆凸共轭的定义 (定义 7.4) 以及 (实) 点积的对称性, 我们有

$$\max_{\mathbf{u}} \left[\min_{\mathbf{y}} -\mathbf{y}^\top \mathbf{u} + f(\mathbf{y}) \right] + \left[\min_{\mathbf{x}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{u} + g(\mathbf{x}) \right] \quad (7.67a)$$

$$= \max_{\mathbf{u}} -f^*(\mathbf{y}) - g^*(-\mathbf{A}^\top \mathbf{u}). \quad (7.67b)$$

于是我们就证明了

$$\min_{\mathbf{x}} f(\mathbf{Ax}) + g(\mathbf{x}) = \max_{\mathbf{u}} -f^*(\mathbf{u}) - g^*(-\mathbf{A}^\top \mathbf{u}). \quad (7.68)$$

事实上, Legendre-Fenchel 共轭在可表示为凸优化的机器学习中非常有用。特别地, 对于独立作用于每个数据的损失函数, 共轭损失函数是推导对偶问题的便捷方法。

< 上一章节

下一章节 >

7.2 约束优化和 Lagrange 乘子

7.4 拓展阅读





7.4 拓展阅读

连续优化是一个活跃的研究领域，我们并不试图对近期进展进行全面的介绍。

从梯度下降的角度来看，它有两个主要弱点，每个弱点都有相应的文献。第一个挑战是梯度下降是一种一阶算法，它不使用有关表面曲率的信息。当存在“狭长的山谷”时，梯度垂直于感兴趣的方向。动量的概念可以推广到一类加速方法（Nesterov, 2018）。共轭梯度法通过考虑先前的方向来避免梯度下降面临的问题（Shewchuk, 1994）。二阶方法（如 Newton 法）使用 Hessian 矩阵来提供有关曲率的信息。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）。拟 Newton 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指标产生了兴趣，从而产生了诸如镜像下降（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。当函数中有扭结时，梯度方法定义不明确。在这些情况下，可以使用次梯度法（Shor, 1985）。有关优化不可微函数的更多信息和算法，请参阅 Bertsekas (1999) 的书。有大量关于数值求解连续优化问题的不同方法的文献，包括约束优化问题的算法。理解这类文献的良好起点是阅读 Luenberger (1969) 和 Bonnans et al. (2006) 的著作。Bubeck (2015) 提供了关于连续优化的最新综述。

现代机器学习的应用通常意味着数据集的大小限制了批量梯度下降的使用，因此随机梯度下降是当前大规模机器学习方法的主力。最近的文献综述包括 Hazan (2015) 和 Bottou et al. (2018)。

关于对偶和凸优化，Boyd and Vandenberghe (2004) 的著作包含在线讲座和幻灯片。Bertsekas (2009) 提供了更数学化的处理，而优化领域一位关键研究人员最近出版的著作是 Nesterov (2018)。凸优化基于凸分析，对凸函数更基础结果感兴趣的读者可以参考 Rockafellar (1970)、Hiriart-Urruty 和 Lemaréchal (2001) 以及 Borwein 和 Lewis (2006)。上述关于凸分析的书籍也涵盖了 Legendre–Fenchel 变换，但 Zia et al. (2009) 的著作中提供了更适合初学者的介绍。Polyak (2016) 的著作概述了 Legendre–Fenchel 变换在凸优化算法分析中的作用。



< 上一章节

下一章节 >

7.3 凸优化

习题



习题

1

考虑一元函数

$$f(x) = x^3 + 6x^2 - 3x - 5$$

求它的稳定点，并分析它们是极小值、极大值还是鞍点

2

考虑公式 (7.15) 中的 SGD 更新规则，写出批量大小为 1 时的更新公式

3

判断正误

- 任意两个凸集的交还是凸集
- 任意两个凸集的并还是凸集
- 两个凸集 A 和 B ，差集 $A - B$ 还是凸集

4

判断正误

- 两个凸函数的和还是凸函数
- 两个凸函数的差还是凸函数



- 两个凸函数的乘积还是凸函数
- 两个凸函数 f 和 g , 则 $\max\{f, g\}$ 还是凸函数

5

将下面的优化问题转化为矩阵形式的线性优化问题

$$\max_{\mathbf{x} \in \mathbb{R}^2, \xi \in \mathbb{R}} \mathbf{p}^\top \mathbf{x} + \xi$$

其中约束是 $\xi \geq 0, x_0 \leq 0, x_1 \leq 3$ 。

6

考虑图 7.9 中所示的线性规划问题:

$$\min_{\mathbf{x} \in \mathbb{R}^2} - \begin{bmatrix} 5 \\ 3 \end{bmatrix}^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (1)$$

$$\text{subject to } \begin{bmatrix} 2 & 2 \\ 2 & -4 \\ -2 & 1 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 33 \\ 8 \\ 5 \\ -1 \\ 8 \end{bmatrix} \quad (2)$$

使用 Lagrange 对偶求该问题的对偶线性规划问题

7

考虑图 7.4 中所示的二次规划问题:



$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^2} \quad & \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^\top \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 5 \\ 3 \end{bmatrix}^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ \text{subject to} \quad & \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \end{aligned} \tag{4}$$

使用 Lagrange 对偶求该问题的对偶二次规划问题

8

考虑下面的凸优化问题

$$\min_{\boldsymbol{w} \in \mathbb{R}^D} \quad \boldsymbol{w}^\top \boldsymbol{w} \tag{5}$$

$$\text{subject to } \boldsymbol{w}^\top \boldsymbol{x} \geq 1. \tag{6}$$

引入 Lagrange 乘子 λ , 求该问题的 Lagrange 对偶

9

考虑向量 $\boldsymbol{x} \in \mathbb{R}^D$ 的负熵

$$f(\boldsymbol{x}) = \sum_{d=1}^D x_d \log x_d.$$

假设我们使用的内积是标准内积, 求它的凸共轭函数 $f^*(s)$

提示 考虑某个函数, 并令其梯度为零

10

考慮下面的函数



$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x} + \mathbf{b}^\top \mathbf{x} + c,$$

其中 \mathbf{A} 是严格正定矩阵（可逆）。求 $f(\mathbf{x})$ 的凸共轭

11

常用于支持向量机（SVM）的铰链损失的形式如下：

$$L(\alpha) = \max \{0, 1 - \alpha\},$$

如果我们想要用梯度方法（例如 L-BGFS）求其最小值，并避免用到次梯度，我们需要对其不可微点“光滑”化。计算铰链损失的凸共轭 $L^*(\beta)$ （其中 β 是对偶变量），加上一个 ℓ_2 邻近项，然后再计算下面函数的凸共轭

$$L^*(\beta) + \frac{\gamma}{2} \beta^2,$$

其中 γ 是超参数。

◀ 上一章节

7.4 拓展阅读



第8章 当模型遇上数据

在本书的第一部分，我们介绍了构成许多机器学习方法基础的数学知识。我们希望读者能够从第一部分学习到数学语言的基础知识，接下来我们将用这些知识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的是说明如何将第一部分介绍的数学概念用于设计机器学习算法，以解决这四个支柱范围内的任务。我们并不打算引入高级机器学习概念，而是提供一系列实用的方法，使读者能够应用他们从本书第一部分获得的知识。同时，对于已经熟悉数学的读者来说，这也为他们通往更广泛的机器学习文献提供了门户。

< 上一章节

第七章 连续优化

下一章节 >

第九章 线性回归



8.1 数据、模型与学习

此时，值得停下来思考一下机器学习算法旨在解决的问题。正如第一章所讨论的，机器学习系统主要由三个部分组成：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探讨它。同时，如何客观地定义“好”这个词也并非完全显而易见。机器学习的一个指导原则是，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，如准确率或与真实情况的距离，并找出在这些性能指标下表现良好的方法。本章将介绍一些常用于讨论机器学习模型的数学和统计语言的基本要素。通过这样做，我们简要概述了训练模型的最佳实践，以便得到的预测器在尚未见过的数据上表现良好。

Name	Gender	Degree	Postcode	Age	Annual salary
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

表8.1来自非数字格式的虚构人力资源数据库的示例数据。

如第一章所述，“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，以及在不同模型之间进行选择的想法。我们将在第8.2节中介绍经验风险最小化的框架，在第8.3节中介绍最大似然估计的原理，在第8.4节中介绍概率模型的思想。我们将在第8.5节中简要概述一种用于指定概率模型的图形语言，并在第8.6节中讨论模型选择。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

我们假设数据可以被计算机读取，并以数值格式进行充分表示。数据被假定为表格形式（如图8.1所示），其中每一行代表一个特定的实例或示例，每一列代表一个特定的特征。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、网页的文本和图像内容以及社交媒体图表。我们不在

此讨论识别良好特征的重要性和挑战性。这些方面的许多内容都取决于领域专业知识，需要仔细设计，并且在近年来，它们已被纳入数据科学的范畴（Stray, 2016; Adhikari and DeNero, 2018）。

即使我们拥有表格格式的数据，也仍然需要做出选择以获得数值表示。例如，在表8.1中，性别列（一个分类变量）可以转换为数字0表示“男性”，1表示“女性”。或者，性别也可以用数字-1, +1分别表示（如表8.2所示）。此外，在构建表示时经常使用领域知识也很重要，例如知道大学学位从学士到硕士再到博士的进展，或者意识到提供的邮政编码不仅仅是一串字符，而实际上是对伦敦某个地区的编码。在表8.2中，我们将表8.1中的数据转换为数值格式，每个邮政编码都用两个数字表示，即纬度和经度。即使是可能直接读入机器学习算法的数值数据，也应该仔细考虑其单位、缩放和约束。在没有其他信息的情况下，应该对数据集的所有列进行平移和缩放，使其经验均值为0，经验方差为1。为了本书的目的，我们假设领域专家已经对数据进行了适当的转换，即每个输入 x_n 是一个 D 维实数向量，这些实数被称为特征、属性或协变量。我们认为数据集的特征形式如图8.2所示。请注意，在新的数值表示中，我们省略了表8.1中的“姓名”列。这样做有两个主要原因：（1）我们不期望标识符（即姓名）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保护员工的隐私。

Gender	ID	Degree	Latitude (in degrees)	Longitude (in degrees)	Age	Annual Salary (in thousands)
-1		2	51.5073	0.1290	36	89.563
-1		3	51.5074	0.1275	47	123.543
+1		1	51.5071	0.1278	26	23.989
-1		1	51.5075	0.1281	68	138.769
+1		2	51.5074	0.1278	33	113.888

表8.2来自一个虚构的人力资源数据库的示例数据（见表8.1），被转换为数字格式。

在本书的这一部分，我们将使用 N 来表示数据集中的示例数量，并用小写字母 $n = 1, \dots, N$ 对示例进行索引。我们假设给定了一组数值数据，表示为一个向量数组（表8.2）。每一行都是一个特定的个体 x_n ，在机器学习中通常被称为示例或数据点。下标 n 表示这是数据集中总共 N 个示例中的第 n 个示例。每一列代表示例的一个特定特征，我们用 $d = 1, \dots, D$ 对特征进行索引。请记住，数据以向量的形式表示，这意味着每个示例（每个数据点）都是一个 D 维向量。表格的方向源自数据库社区，但对于某些机器学习算法（例如，在第10章中），将示例表示为列向量更为方便。

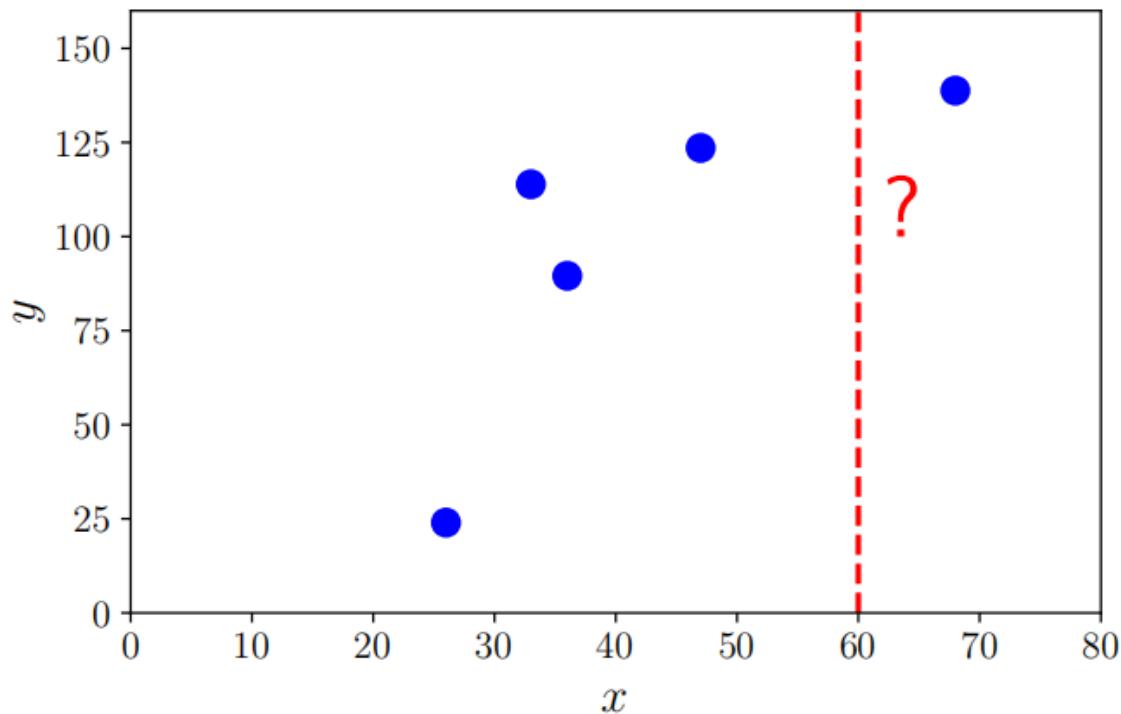


图8.1用于线性回归的玩具数据。来自表8.2最右边两列的训练数据 (x_n, y_n) 对。我们感兴趣的是一个60岁（岁的 $x=60$ ）的工资，用垂直虚线表示，这不是训练数据的一部分。

让我们考虑基于表8.2中的数据，根据年龄预测年薪的问题。这被称为监督学习问题，其中每个示例 x_n （即年龄）都与一个标签

$$y_n$$

（即薪资）相关联。标签

$$y_n$$

还有其他各种名称，包括目标、响应变量和注释。数据集被写为一组示例-标签对 $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n), \dots, (\mathbf{x}_N, y_N)\}$ 。示例表 $\{x_1, \dots, x_N\}$ 经常被串联起来，并写为 $\mathbf{X} \in \mathbb{R}^{N \times D}$ 。图8.1展示了由表8.2中最右两列组成的数据集，其中 $x =$ 年龄， $y =$ 薪资。

我们使用本书第一部分介绍的概念来形式化机器学习问题，如前面段落中的问题。将数据表示为向量 \mathbf{x}_n 使我们能够使用线性代数中的概念（在第2章中介绍）。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章中看到的那样，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一种直觉，即具有

相似特征的示例应该具有相似的标签。比较两个向量需要我们构造一个几何结构（在第3章中解释），并允许我们使用第7章中的技术来优化所得的学习问题。

由于我们有了数据的向量表示，我们可以对数据进行操作以找到其潜在的更好表示。我们将通过两种方式讨论如何找到好的表示：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，我们将看到一个通过找到主成分来找到原始数据空间低维近似的例子。找到主成分与第4章中介绍的特征值和奇异值分解的概念密切相关。对于高维表示，我们将看到一个显式的特征映射 $\phi(\cdot)$ ，它允许我们使用更高维的表示 $\phi(x_n)$ 来表示输入 x_n 。高维表示的主要动机是我们可以将新特征构建为原始特征的非线性组合，这反过来可能使学习问题变得更容易。我们将在第9.2节中讨论特征映射，并在第12.4节中展示这个特征映射如何导致核的出现。近年来，深度学习方法（Goodfellow et al., 2016）已显示出使用数据本身来学习新的良好特征的潜力，并在计算机视觉、语音识别和自然语言处理等领域取得了巨大成功。本书这一部分不会涵盖神经网络，但读者可参考第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。

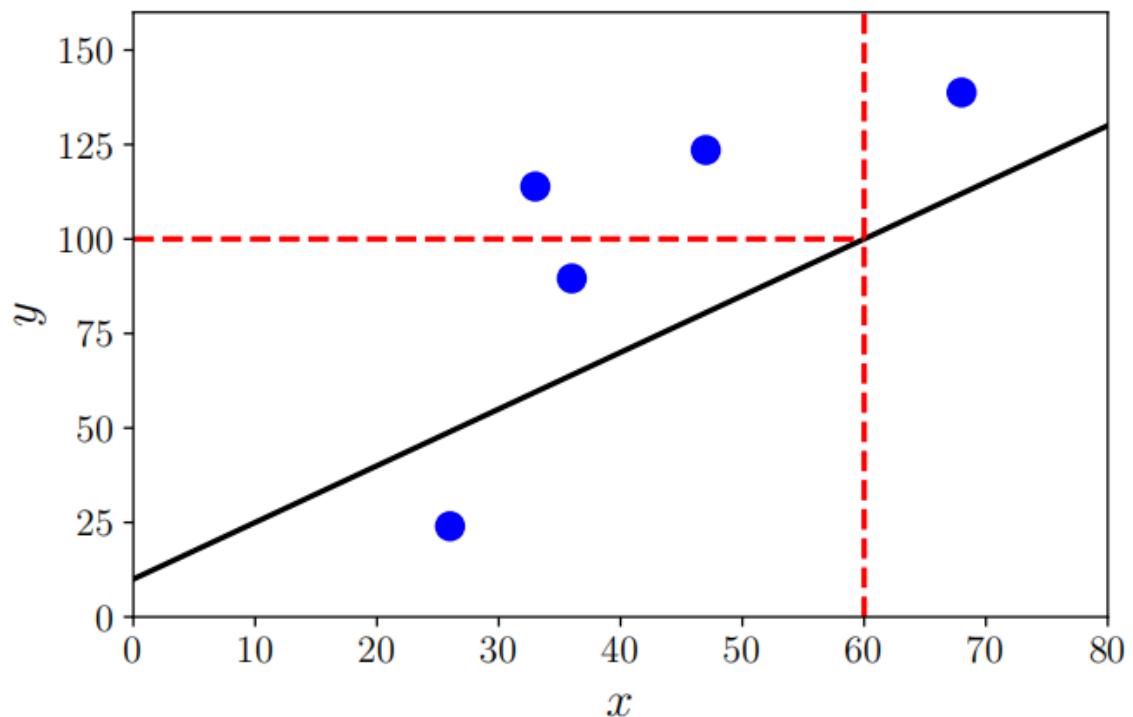


图8.2示例函数（黑色实对角线）及其在 $x = 60$ 处的预测，即 $f(60) = 100$ 。

8.1.2 模型作为函数

一旦我们将数据以适当的向量形式表示，我们就可以着手构建预测函数（称为预测器）。在第1章中，我们还没有精确描述模型的语言。现在，使用本书第一部分的概念，我们可以介绍“模型”的含义。本书提出了两种主要方法：将预测器视为函数，以及将预测器视为概率模型。我们在这里描述前者，并在下一小节中描述后者。

预测器是一个函数，当给定一个特定的输入示例（在我们的情况下，是一个特征向量）时，会产生一个输出。目前，我们将输出视为一个单一的数字，即一个实值标量输出。这可以写为

$$f : \mathbb{R}^D \rightarrow \mathbb{R},$$

(8.1)

其中输入向量 x 是 D 维的（具有 D 个特征），然后函数 f 应用于它（写为 $f(x)$ ）并返回一个实数。图 8.2 展示了一个可能的函数，该函数可用于计算输入值 x 的预测值。

在本书中，我们不考虑所有函数的一般情况，因为这会涉及泛函分析。相反，我们考虑线性函数的特殊情况

$$f(x) = \boldsymbol{\theta}^\top \mathbf{x} + \theta_0$$

(8.2)

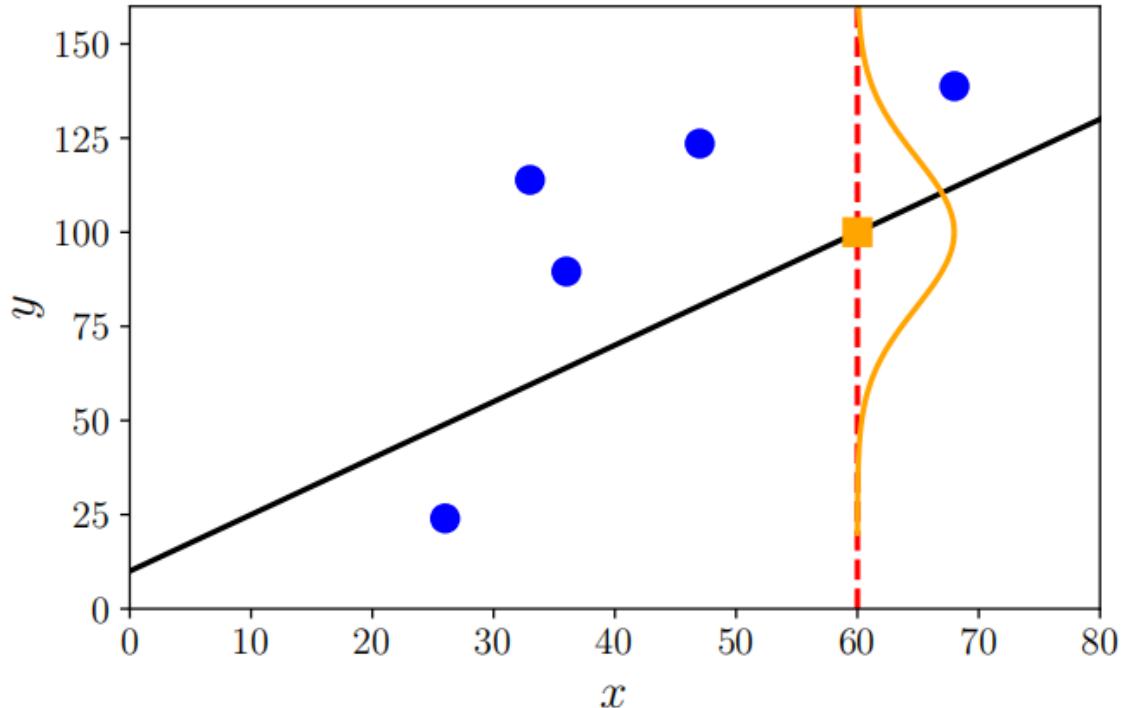


图8.3示例函数（黑色实对角线）及其在 $x = 60$ 处的预测不确定性（绘制为高斯曲线）。

其中 θ 和 θ_0 是未知的。这一限制意味着第2章和第3章的内容足以精确阐述非概率（与接下来描述的概率观点相比）机器学习观点下的预测器概念。线性函数在可解决问题的一般性和所需背景数学知识的数量之间取得了良好的平衡。

8.1.3 模型作为概率分布

我们通常认为数据是对某些真实潜在效应的有噪声观测，并希望通过应用机器学习从噪声中识别出信号。这要求我们有一种量化噪声效应的语言。我们还经常希望预测器能够表达某种不确定性，例如，量化我们对特定测试数据点的预测值所具有的信心。正如我们在第6章中所见，概率论提供了一种量化不确定性的语言。图8.3展示了函数预测不确定性的高斯分布图示。

我们不必将预测器视为单个函数，而可以将其视为概率模型，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的特殊情况，这使我们能够描述概率模型而无需涉及随机过程和随机测度。对于这个特殊情况，我们可以将概率模型视为多元概率分布，这已经允许了一个丰富的模型类别。

我们将在第8.4节中介绍如何使用概率（第6章）中的概念来定义机器学习模型，并在第8.5节中介绍一种图形语言，以便以紧凑的方式描述概率模型。

8.1.4 学习是寻找参数

学习的目标是找到一个模型及其对应的参数，使得得到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，从概念上讲，有三个不同的算法阶段：

1. 预测或推断
2. 训练或参数估计
3. 超参数调整或模型选择

预测阶段是我们在之前未见过的测试数据上使用已训练的预测器的过程。换句话说，参数和模型选择已经固定，预测器被应用于代表新输入数据点的新向量。如第1章和前一小节所述，本书将考虑两种机器学习流派，分别对应于预测器是函数还是概率模型。当我们有概率模型（在第8.4节中进一步讨论）时，预测阶段被称为推断。

备注：不幸的是，对于不同的算法阶段并没有统一的命名。单词“推断”有时也用于表示概率模型的参数估计，而较少用于表示非概率模型的预测。



训练或参数估计阶段是我们根据训练数据调整预测模型的过程。我们希望在给定训练数据的情况下找到好的预测器，并有两种主要策略来实现这一点：基于某种质量度量找到最佳预测器（有时称为找到点估计），或使用贝叶斯推断。找到点估计可以应用于两种类型的预测器，但贝叶斯推断需要概率模型。

对于非概率模型，我们遵循经验风险最小化的原则，这将在第8.2节中描述。经验风险最小化直接为寻找良好参数提供了一个优化问题。对于统计模型，我们使用最大似然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来进一步模拟参数的不确定性，这将在第8.4节中更详细地讨论。

我们使用数值方法来找到适合数据的良好参数，大多数训练方法都可以视为寻找目标最大值的爬山方法，例如似然函数的最大值。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，我们感兴趣的是基于数据学习模型，以便它在未来的数据上表现良好。仅使模型很好地拟合训练数据是不够的，预测器还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）来模拟预测器在未来未见数据上的行为。正如我们将看到的，在哲学上，这既不是归纳也不是演绎，而被称为溯因。根据斯坦福哲学百科全书，溯因是推断最佳解释的过程（Douven, 2017）。

我们通常需要就预测器的结构做出高级建模决策，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。在不同模型之间做出选择的问题被称为模型选择，我们将在第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，这将在第8.6.1节中描述。我们还使用模型选择来选择我们模型的超参数。

备注：参数和超参数之间的区别有些任意，主要是由可以数值优化与需要使用搜索技术的区别所驱动的。考虑这种区别的另一种方式是，将参数视为概率模型的显式参数，而将超参数（高级参数）视为控制这些显式参数分布的参数。



在以下部分中，我们将探讨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）和概率建模（第8.4节）。



[下一章节 >](#)

8.2 经验风险最小化



8.2 经验风险最小化

在我们掌握了所有相关的数学知识之后，我们现在可以介绍学习的含义了。机器学习中的“学习”部分实质上就是基于训练数据来估计参数。在本节中，我们考虑预测器是一个函数的情况，而概率模型的情况将在第8.3节中讨论。我们将描述经验风险最小化的概念，这一概念最初是由支持向量机（第12章描述）的提出而普及的。然而，其一般原则具有广泛的适用性，使我们能够在不显式构建概率模型的情况下，探讨学习的本质。以下是四个主要的设计选择，我们将在以下小节中详细讨论：

8.2.1 我们允许预测器采用哪些函数集？

8.2.2 我们如何衡量预测器在训练数据上的表现好坏？

8.2.3 我们如何仅从训练数据中构建预测器，使其在未见过的测试数据上表现良好？

8.2.4 在模型空间中搜索的程序是什么？

8.2.1 函数假设类

假设我们得到 N 个样本 $\mathbf{x}_n \in \mathbb{R}^D$ 和对应的标量标签 $y_n \in \mathbb{R}$ 。我们考虑监督学习的设置，其中我们获得样本对 $(x_1, y_1), \dots, (x_N, y_N)$ 。基于这些数据，我们希望估计一个预测器 $f(\cdot, \theta) : \mathbb{R}^D \rightarrow \mathbb{R}$ ，它通过参数 θ 进行参数化。我们希望能够找到一个好的参数 θ^* ，以便很好地拟合数据，即

$$f(\mathbf{x}_n, \theta^*) \approx y_n \quad \text{对于所有 } n = 1, \dots, N.$$

(8.3)

在本节中，我们使用符号 $\hat{y}_n = f(x_n, \theta^*)$ 来表示预测器的输出。

备注：为了便于阐述，我们将以监督学习（即我们有标签）的角度来描述经验风险最小化。这简化了假设类和损失函数的定义。在机器学习中，选择一类参数化函数也很常见，例如仿射函数。

例8.1



我们引入普通最小二乘回归问题来说明经验风险最小化。关于回归的更全面介绍将在第9章中给出。当标签 y_n 是实数值时，预测器函数类的一个流行选择是仿射函数集。我们通过向 x_n 添加一个额外的单位特征 $x^{(0)} = 1$ ，即 $x_n = [1, x_n^{(1)}, x_n^{(2)}, \dots, x_n^{(D)}]^\top$ ，来更简洁地表示仿射函数。相应地，机器学习参数向量是 $\theta = [\theta_0, \theta_1, \theta_2, \dots, \theta_D]^\top$ ，这使得我们可以将预测器写为线性函数

$$f(x_n, \theta) = \theta^\top x_n.$$

(8.4)

这个线性预测器等价于仿射模型

(8.5)

$$f(x_n, \theta) = \theta_0 + \sum_{d=1}^D \theta_d x_n^{(d)}.$$

预测器以表示单个样本 x_n 的特征向量为输入，并产生实数值输出，即 $f : \mathbb{R}^{D+1} \rightarrow \mathbb{R}$ 。本章前面的图表中，预测器是一条直线，这意味着我们假设了一个仿射函数。

除了线性函数外，我们可能还希望考虑非线性函数作为预测器。神经网络领域的最新进展使得能够高效地计算更复杂的非线性函数类。

给定这类函数，我们想要寻找一个好的预测器。现在，我们转向经验风险最小化的第二个要素：如何测量预测器与训练数据的匹配程度。

8.2.2 训练损失函数

考虑一个特定样本的标签 y_n ，以及我们基于 x_n 做出的相应预测 \hat{y}_n 。为了定义什么是良好的数据拟合，我们需要指定一个损失函数 $\ell(y_n, \hat{y}_n)$ ，该函数以真实标签和预测值为输入，并产生一个非负数值（称为损失），表示我们在该特定预测上犯了多大的错误。我们寻找一个好的参数向量 θ^* 的目标是最小化在 N 个训练样本集上的平均损失。



机器学习中的一个常见假设是，样本集 $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ 是独立同分布的。独立（第6.4.5节）一词意味着两个数据点 (\mathbf{x}_i, y_i) 和 (\mathbf{x}_j, y_j) 在统计上互不依赖，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 $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ ，我们引入示例矩阵 $\mathbf{X} := [x_1, \dots, x_N]^\top \in \mathbb{R}^{N \times D}$ 和标签向量 $\mathbf{y} := [y_1, \dots, y_N]^\top \in \mathbb{R}^N$ 的符号。使用这种矩阵符号，平均损失由下式给出：

$$\mathbf{R}_{\text{emp}}(f, \mathbf{X}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^N \ell(y_n, \hat{y}_n),$$

(8.6)

其中 $\hat{y}_n = f(\mathbf{x}_n, \boldsymbol{\theta})$ 。式 (8.6) 被称为经验风险，它取决于三个参数：预测器 f 和数据 \mathbf{X}, \mathbf{y} 。这种学习策略通常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示例，我们指定使用平方损失 $\ell(y_n, \hat{y}_n) = (y_n - \hat{y}_n)^2$ 来衡量训练过程中犯错的代价。我们希望最小化经验风险 (8.6)，即数据上损失的平均值

(8.7)

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^D} \frac{1}{N} \sum_{n=1}^N (y_n - f(\mathbf{x}_n, \boldsymbol{\theta}))^2,$$

其中我们用预测器 $\hat{y}_n = f(\mathbf{x}_n, \boldsymbol{\theta})$ 进行了替换。通过选择线性预测器 $f(\mathbf{x}_n, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x}_n$ ，我们得到优化问题

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^D} \frac{1}{N} \sum_{n=1}^N (y_n - \boldsymbol{\theta}^\top \mathbf{x}_n)^2.$$

(8.8)

这个方程可以等价地用矩阵形式表示

(8.9)

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^D} \frac{1}{N} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|^2.$$

这被称为最小二乘问题。通过求解正规方程，我们可以得到一个闭式解析解，这将在第9.2节中讨论。

我们并不关心仅在训练数据上表现良好的预测器。相反，我们寻求的是在未见的测试数据上表现良好（风险低）的预测器。更正式地说，我们感兴趣的是找到一个预测器 f （参数固定），该预测器能够最小化预期风险

(8.10)

$$\mathbf{R}_{\text{true}}(f) = \mathbb{E}_{x,y}[\ell(y, f(\mathbf{x}))],$$

其中 y 是标签， $f(x)$ 是基于样本 x 的预测。符号 $\mathbf{R}_{\text{true}}(f)$ 表示如果我们拥有无限量的数据，这就是真正的风险。该期望是针对所有可能的数据和标签的（无限）集合。从我们通常希望最小化预期风险的愿望中，产生了两个实际问题，我们将在以下两个小节中讨论：

- 我们应该如何改变训练过程以使其具有良好的泛化能力？
- 我们如何从（有限）数据中估计预期风险？

备注。许多机器学习任务都指定了相关的性能指标，例如预测的准确性或均方根误差。性能指标可能更复杂，对成本敏感，并捕获特定应用的详细信息。原则上，用于经验风险最小化的损失函数的设计应直接对应于机器学习任务指定的性能指标。但在实践中，损失函数的设计与性能指标之间往往存在不匹配。这可能是由于实现简便性或优化效率等问题导致的。

8.2.3 正则化以减少过拟合

本节介绍了一种对经验风险最小化的补充方法，使其能够很好地泛化（即近似最小化预期风险）。回顾一下，训练机器学习预测器的目的是使我们在未见过的数据上也能表现良好，即预测器具有很好的泛化能力。我们通过保留整个数据集的一部分来模拟这种未见过的数据，这部分数据被称为测试集。给定一个足够丰富的预测器 f 的函数类，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（因此是风险）来说很好，但我们不会期望预测器在未见过的数据上有很好的泛化能力。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，而测试集（在训练过程中机器学习算法未见过）用于评估泛化性能。重要的是，用户在观察测试集后不应回到训练的新一轮循环中。我们使用下标train和test来分别表示训练集和测试集。我们将在第8.2.4节中重新讨论使用有限数据集来评估预期风险的想法。

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，而不能很好地泛化到新数据（Mitchell, 1997）。这种在训练集上平均损失很小但在测试集上平均损失很大的普遍现象，往往发生在我们拥有少量数据和复杂假设类时。对于特定的预测器 f （参数固定），当过拟合现象发生时，来自训练数据的风险估计 $\mathbf{R}_{\text{emp}}(f, \mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}})$ 会低估预期风险 $\mathbf{R}_{\text{true}}(f)$ 。由于我们使用测试集上的经验风险 $\mathbf{R}_{\text{emp}}(f, \mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}})$ 来估计预期风险 $\mathbf{R}_{\text{true}}(f)$ ，如果测试风险远大于训练风险，这就是过拟合的迹象。我们将在第8.3.3节中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寻找经验风险最小化的最小化器，这使得优化器更难返回一个过于灵活的预测器。在机器学习中，这个惩罚项被称为正则化。正则化是在经验风险最小化的准确解与解的大小或复杂性之间做出妥协的一种方式。

示例 8.3 (正则化最小二乘法)

正则化是一种方法，用于阻止优化问题中出现复杂或极端的解决方案。最简单的正则化策略是通过添加一个仅涉及 θ 的惩罚项，将前一个示例中的最小二乘问题

$$\min_{\theta} \frac{1}{N} \|\mathbf{y} - \mathbf{X}\theta\|^2.$$

(8.11)

替换为“正则化”问题：

$$\min_{\theta} \frac{1}{N} \|\mathbf{y} - \mathbf{X}\theta\|^2 + \lambda \|\theta\|^2.$$

(8.12)

其中，附加项 $\|\theta\|^2$ 被称为正则化项，而参数 λ 被称为正则化参数。正则化参数在训练集上的损失最小化和参数 θ 的幅度之间进行了权衡。当出现过拟合时，参数值的幅度往往变得相对较大（Bishop, 2006）。

正则化项有时被称为惩罚项，它促使向量 θ 更接近原点。正则化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节中重新探讨这个思想。在第12章中，我们将看到正则化的思想与大间隔的思想是等价的。

8.2.4 交叉验证以评估泛化性能

我们在上一节中提到，我们通过将预测器应用于测试数据来估计泛化误差以衡量其性能。这些数据有时也被称为验证集。验证集是我们保留在外的可用训练数据的一个子集。这种方法的一个实际问题是数据量有限，而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。这将要求我们保持验证集 \mathcal{V} 较小，但这会导致预测性能的估计具有噪声（高方差）。解决这些相互矛盾的目标（大训练集、大验证集）的一个方法是使用交叉验证。 K 折交叉验证有效地将数据分成 K 个部分，其中 $K - 1$ 个部分形成训练集 \mathcal{R} ，最后一个部分作为验证集 \mathcal{V} （类似于前面概述的想法）。交叉验证（理想情况下）遍历将数据块分配给 \mathcal{R} 和 \mathcal{V} 的所有组合；见图8.4。此过程针对验证集的 K 种选择重复进行，并对 K 次运行的模型性能进行平均。

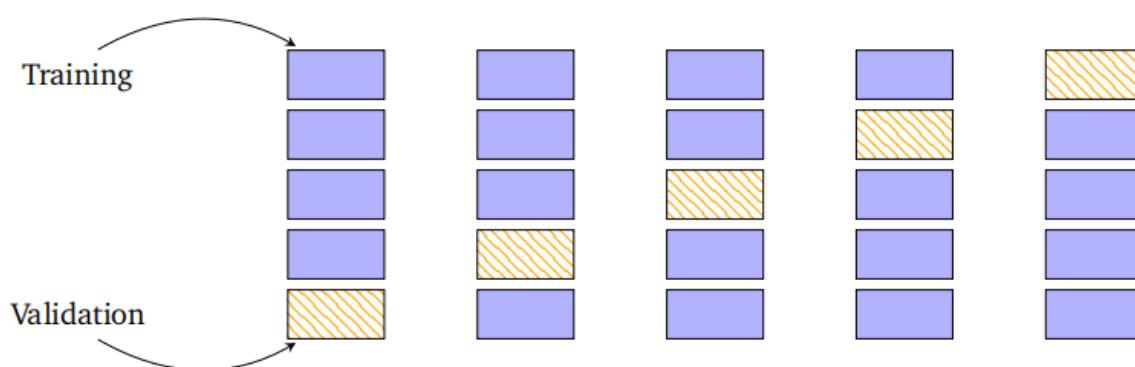


图8.4 k 倍交叉验证。数据集被分为 $K = 5$ 个块，其中 $K-1$ 作为训练集（蓝色），一个作为验证集（橙色孵化）。

我们将数据集分为两个不重叠的集合 $\mathcal{D} = \mathcal{R} \cup \mathcal{V}$ ($\mathcal{R} \cap \mathcal{V} = \emptyset$)，其中 \mathcal{V} 是验证集，我们在 \mathcal{R} 上训练模型。训练后，我们在验证集 \mathcal{V} 上评估预测器 f 的性能（例如，



通过计算验证集上训练模型的均方根误差（RMSE））。更准确地说，对于每个分区 k ，训练数据 $\mathcal{R}^{(k)}$ 产生一个预测器 $f^{(k)}$ ，然后将其应用于验证集 $\mathcal{V}^{(k)}$ 以计算经验风险 $R(f^{(k)}, \mathcal{V}^{(k)})$ 。我们遍历验证集和训练集的所有可能分区，并计算预测器的平均泛化误差。交叉验证近似于期望泛化误差

$$\mathbb{E}_{\mathcal{V}}[R(f, \mathcal{V})] \approx \frac{1}{K} \sum_{k=1}^K R(f^{(k)}, \mathcal{V}^{(k)})$$

(8.13)

其中 $R(f^{(k)}, \mathcal{V}^{(k)})$ 是预测器 $f^{(k)}$ 在验证集 $\mathcal{V}^{(k)}$ 上的风险（例如，RMSE）。该近似有两个来源：首先，由于有限的训练集，导致不是最佳的 $f^{(k)}$ ；其次，由于有限的验证集，导致对风险 $R(f^{(k)}, \mathcal{V}^{(k)})$ 的估计不准确。 K 折交叉验证的一个潜在缺点是训练模型 K 次的计算成本很高，如果训练成本在计算上很昂贵，这可能会成为负担。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，多个正则化参数），这些可能不是模型的直接参数。根据这些超参数评估模型的质量，可能会导致训练次数与模型参数数量成指数关系。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，交叉验证是一个令人尴尬的并行问题，即将问题分离为多个并行任务所需的努力很少。在拥有足够的计算资源（例如，云计算、服务器集群）的情况下，交叉验证所需的时间不会比单次性能评估更长。在本节中，我们了解到经验风险最小化基于以下概念：函数假设类、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想法的效果。

8.2.5 拓展阅读

由于经验风险最小化（Vapnik, 1998）的原始发展采用了大量理论性语言，随后的许多发展也多为理论性。这一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）。一本基于理论基础并开发了高效学习算法的最新机器学习教科书是 Shalev-Shwartz 和 Ben-David (2014) 所著。

正则化概念起源于不适定逆问题的求解（Neumaier, 1998）。本文介绍的方法称为 Tikhonov 正则化，并且有一个与之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（Bühlmann and Van De Geer, 2011）

有着深厚的联系。交叉验证的替代方法是自助法和刀切法（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 $p(\mathbf{x}, \mathbf{y})$ ，它控制着数据的生成。然而，经验风险最小化的方法对该分布的选择是不可知的。这与明确要求知道 $p(\mathbf{x}, \mathbf{y})$ 的标准统计方法形成对比。此外，由于分布是样本 \mathbf{x} 和标签 \mathbf{y} 的联合分布，标签可能是非确定性的。与标准统计不同，我们不需要为标签 \mathbf{y} 指定噪声分布。

< 上一章节

下一章节 >

8.1 数据, 模型与学习

8.3 参数估计



8.3 参数估计

在第8.2节中，我们没有使用概率分布来明确建模我们的问题。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引起的不确定性以及我们预测器参数中的不确定性。在第8.3.1节中，我们将介绍似然函数，它与经验风险最小化中的损失函数概念（第8.2.2节）类似。先验（第8.3.2节）的概念则与正则化（第8.2.3节）的概念类似。

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，使我们能够找到一个很好地拟合数据的模型。估计问题集中在似然函数上，或者更精确地说，是其负对数。对于由随机变量 x 表示的数据和由参数 θ 参数化的一组概率密度 $p(x | \theta)$ ，负对数似然由下式给出：

(8.14)

$$\mathcal{L}_x(\theta) = -\log p(x | \theta).$$

符号 $\mathcal{L}_x(\theta)$ 强调了参数 θ 在变化，而数据 x 是固定的。在书写负对数似然时，我们通常会省略对 x 的引用，因为它实际上是 θ 的函数，并在随机变量表示数据中的不确定性从上下文中清楚时，将其写为 $\mathcal{L}(\theta)$ 。

让我们解释对于固定的 $\dot{\theta}$ 值，概率密度 $p(x | \theta)$ 在建模什么。它是一个分布，用于建模给定参数设置下数据的不确定性。对于给定的数据集 x ，似然函数允许我们表达对不同参数设置 θ 的偏好，并可以选择更“可能”生成数据的设置。

从另一个互补的角度来看，如果我们认为数据是固定的（因为它已经被观测到），并且我们改变参数 θ ，那么 $\mathcal{L}(\theta)$ 告诉我们什么？它告诉我们对于观测值 x ， θ 的特定设置有多大的可能性。基于这一观点，最大似然估计器为我们提供了数据集中最可能的参数 θ 。

我们考虑监督学习设置，其中我们获得成对的 $(x_1, y_1), \dots, (x_N, y_N)$ ，其中 $x_n \in \mathbb{R}^D$ 且标签 $y_n \in \mathbb{R}$ 。我们感兴趣的是构建一个预测器，它以特征向量 x_n 作为输入，并产生预测 y_n （或接近它的值），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 下的示例指定了给定样本条件下标签的条件概率分布。

示例 8.4

经常使用的一个示例是指定给定示例的标签的条件概率为高斯分布。换句话说，我们假设可以通过均值为零的独立高斯噪声（参考第6.5节）来解释我们的观测不确定性，即 $\varepsilon_n \sim \mathcal{N}(0, \sigma^2)$ 。我们进一步假设使用线性模型 $\mathbf{x}_n^\top \boldsymbol{\theta}$ 进行预测。这意味着我们为每个示例-标签对 (\mathbf{x}_n, y_n) 指定了一个高斯似然函数，

$$p(y_n | \mathbf{x}_n, \boldsymbol{\theta}) = \mathcal{N}(y_n | \mathbf{x}_n^\top \boldsymbol{\theta}, \sigma^2). \quad (8.15)$$

图8.3展示了给定参数 θ 的高斯似然的一个图示。我们将在第9.2节中看到如何根据高斯分布明确展开上述表达式。

我们假设示例集 $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ 是独立同分布的（i.i.d.）。“独立”（第6.4.5节）一词意味着涉及整个数据集 $(\mathcal{Y} = \{y_1, \dots, y_N\}$ 和 $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ）的似然函数可以分解为每个单独示例似然函数的乘积

(8.16)

$$p(\mathcal{Y} | \mathcal{X}, \boldsymbol{\theta}) = \prod_{n=1}^N p(y_n | \mathbf{x}_n, \boldsymbol{\theta}),$$

其中 $p(y_n | \mathbf{x}_n, \boldsymbol{\theta})$ 是特定的分布（在示例8.4中是高斯分布）。表达式“同分布”意味着乘积(8.16)中的每个项都遵循相同的分布，并且它们共享相同的参数。从优化的角度来看，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

(8.17)

$$\mathcal{L}(\boldsymbol{\theta}) = -\log p(\mathcal{Y} | \mathcal{X}, \boldsymbol{\theta}) = -\sum_{n=1}^N \log p(y_n | \mathbf{x}_n, \boldsymbol{\theta}).$$

尽管在 $p(y_n | \mathbf{x}_n, \boldsymbol{\theta})$ (8.15) 中, $\boldsymbol{\theta}$ 位于条件符号的右侧, 因此可能被误解为是已观测且固定的, 但这种解释是不正确的。负对数似然 $\mathcal{L}(\boldsymbol{\theta})$ 是 $\boldsymbol{\theta}$ 的函数。因此, 为了找到一个能够很好地解释数据 $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ 的良好参数向量 $\boldsymbol{\theta}$, 我们需要对 $\boldsymbol{\theta}$ 最小化负对数似然 $\mathcal{L}(\boldsymbol{\theta})$ 。

备注。(8.17)中的负号是一个历史遗留问题, 源于我们想要最大化似然函数的惯例, 但数值优化文献倾向于研究函数的最小化。

示例 8.5

继续我们在高斯似然 (8.15) 的示例, 负对数似然可以重写为

(8.18a)

(8.18b)

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= -\sum_{n=1}^N \log p(y_n | \mathbf{x}_n, \boldsymbol{\theta}) = -\sum_{n=1}^N \log \mathcal{N}(y_n | \mathbf{x}_n^\top \boldsymbol{\theta}, \sigma^2) \\ &= -\sum_{n=1}^N \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_n - \mathbf{x}_n^\top \boldsymbol{\theta})^2}{2\sigma^2}\right) \\ &= -\sum_{n=1}^N \log \exp\left(-\frac{(y_n - \mathbf{x}_n^\top \boldsymbol{\theta})^2}{2\sigma^2}\right) - \sum_{n=1}^N \log \frac{1}{\sqrt{2\pi\sigma^2}} \\ &= \frac{1}{2\sigma^2} \sum_{n=1}^N (y_n - \mathbf{x}_n^\top \boldsymbol{\theta})^2 - \sum_{n=1}^N \log \frac{1}{\sqrt{2\pi\sigma^2}}.\end{aligned}$$

(8.18c)

(8.18d)

由于 σ 是给定的, 所以(8.18d)中的第二项是常数, 最小化 $\mathcal{L}(\boldsymbol{\theta})$ 等价于解决第一项表示的最小二乘问题 (与(8.8)比较)。

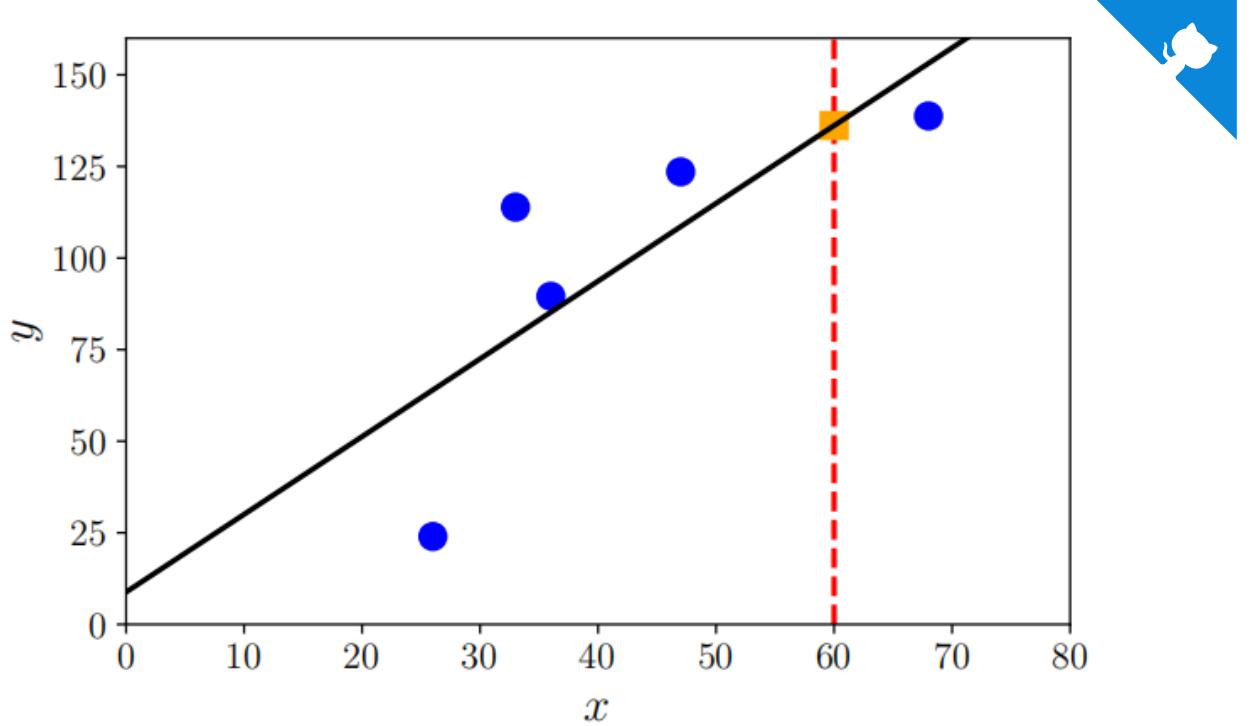


图8.5对于给定的数据，参数的最大似然估计结果为黑色对角线。橙色方方形表示 $x = 60$ 处的最大似然预测值。

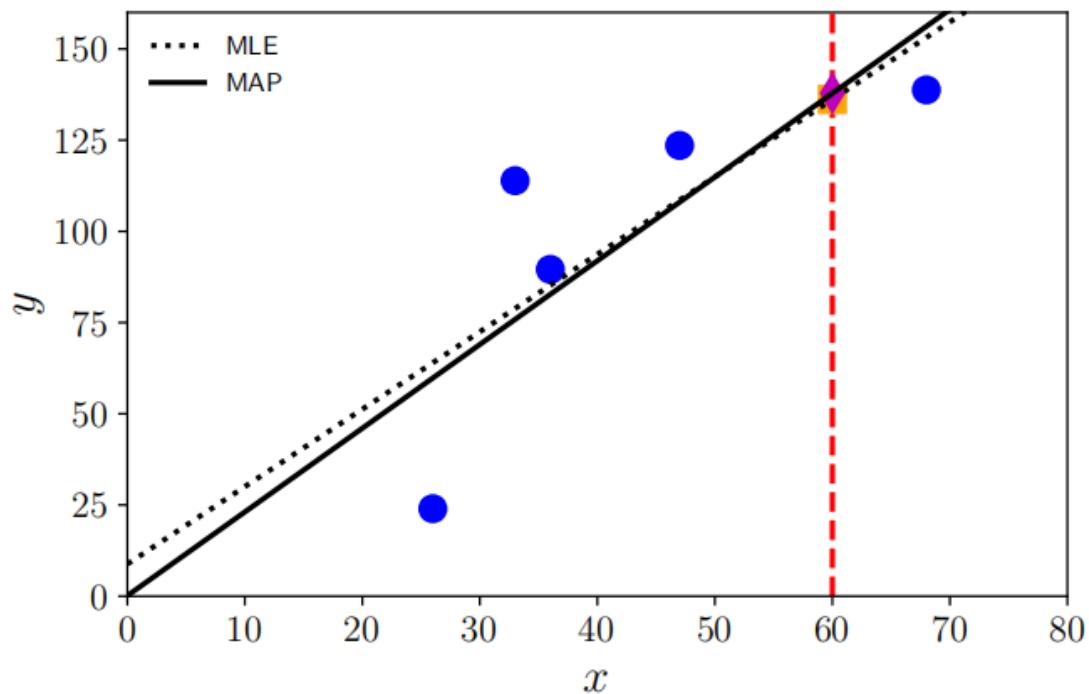


图8.6在 $x = 60$ 时与最大似然估计和MAP估计进行比较。先验使斜率较陡，截距接近于零。在这个例子中，将截距移近于零的偏差实际上增加了斜率。



事实证明，对于高斯似然函数，与最大似然估计相对应的优化问题有一个闭式解。我们将在第9章中看到更多细节。图8.5展示了一个回归数据集和由最大似然参数引起的函数。与未正则化的经验风险最小化（第8.2.3节）类似，最大似然估计可能会受到过拟合的影响（第8.3.3节）。对于其他似然函数，即如果我们用非高斯分布对噪声进行建模，则最大似然估计可能没有闭式解析解。在这种情况下，我们求助于第7章中讨论的数值优化方法。

8.3.2 最大后验估计

如果我们有关于参数 θ 分布的先验知识，我们可以在似然函数上乘以一个额外的项。这个额外的项是参数 θ 的先验概率分布 $p(\theta)$ 。给定一个先验分布，在观察到一些数据 x 之后，我们应该如何更新 θ 的分布？换句话说，在观察到数据 x 之后，我们应该如何表示对 θ 的更具体的知识？如第6.3节所述，贝叶斯定理为我们提供了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） $p(\theta)$ 和函数 $p(x | \theta)$ （该函数将参数 θ 和观测数据 x 联系起来，称为似然函数）中计算出参数 θ 的后验分布 $p(\theta | x)$ （更具体的知识）：

(8.19)

$$p(\theta | x) = \frac{p(x | \theta)p(\theta)}{p(x)}.$$

请注意，我们感兴趣的是找到使后验概率最大化的参数 θ 。由于分布 $p(x)$ 不依赖于 θ ，我们可以在优化过程中忽略分母的值，从而得到

(8.20)

$$p(\theta | x) \propto p(x | \theta)p(\theta).$$

前面的比例关系隐藏了数据密度 $p(x)$ ，这一密度可能很难估计。因此，我们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估计）。图8.6展示了添加一个均值为零的高斯先验的效果示例。

示例 8.6

除了前一个示例中关于高斯似然的假设外，我们还假设参数向量服从均值为零的多元高斯分布，即 $p(\theta) = \mathcal{N}(\mathbf{0}, \Sigma)$ ，其中 Σ 是协方差矩阵（第6.5

节）。请注意，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，将关于良好参数位置的先验知识纳入考虑是一个普遍存在的想法。我们在第8.2.3节中看到的另一种观点是正则化的思想，它引入了一个额外的项，使得得到的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世界和概率世界的桥梁，因为它明确承认了先验分布的需求，但它仍然只产生参数的点估计。

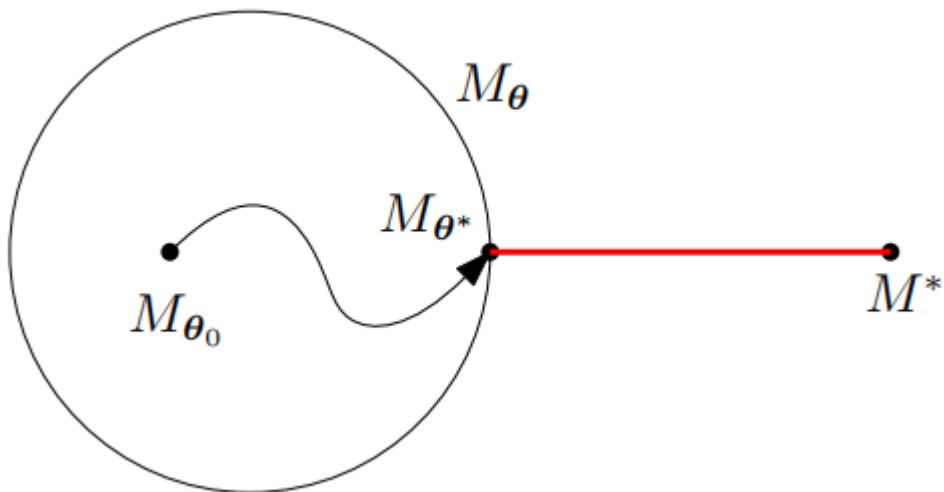


图8.7模型拟合。在参数化的模型 M_θ 类中，我们优化模型参数 θ ，以最小化到真（未知）模型 M^* 的距离。

备注。最大似然估计 θ_{ML} 具有以下性质（Lehmann和Casella, 1998; Efron和Hastie, 2016）：

- 演近一致性：在无限多观测值的极限下，最大似然估计会收敛到真实值，加上一个近似正态的随机误差。
- 实现这些性质所需的样本量可能相当大。
- 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”数据范围内，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）的原则是使用概率建模来推断数据和模型参数中的不确定性。然而，我们尚未将概率建模发挥到极致。在本节中，所得的训练过程仍然产生预测器的点估计，即训练返回一组表示最佳预测器的参数值。在第8.4节中，我

们将采取这样的观点，即参数值也应该被视为随机变量，并且在做出预测时，我们不使用完整的参数分布，而不是估计该分布中的“最佳”值。

8.3.3 模型拟合

考虑这样一个场景，我们被给定一个数据集，并希望将数据拟合到一个参数化模型中。当我们谈论“拟合”时，我们通常指的是优化/学习模型参数，以便它们能够最小化某个损失函数，例如负对数似然。在最大似然估计（第8.3.1节）和最大后验估计（第8.3.2节）中，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，我们可能将输入 x 和（无噪声）观测值 y 之间的关系定义为 $y = ax + b$ ，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 、截距为 b 的直线。假设数据来自一个我们未知的模型 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（例如训练数据上的平方损失）定义。图8.7展示了一个场景，其中我们有一个较小的模型类（由圆 M_θ 表示），而数据生成模型 M^* 位于我们考虑的模型集之外。我们从 M_{θ_0} 开始参数搜索。优化后，即当我们获得最佳参数 θ^* 时，我们区分以下三种不同情况：

- (i) 过拟合，(ii) 欠拟合，和 (iii) 拟合良好。我们将从高层次上直观理解这三个概念的含义。

粗略地说，过拟合指的是参数化模型类过于丰富，以至于无法很好地建模由 M^* 生成的数据集，即 M_θ 可以建模更复杂的数据集。例如，如果数据集是由一个线性函数生成的，但我们定义 M_θ 为七次多项式的类，那么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。我们经常观察到的一个现象是，过于灵活的模型类 M_θ 会使用其全部建模能力来减少训练误差。如果训练数据包含噪声，模型很可能会从噪声中捕捉到一些看似有用的信号。这在我们对训练数据之外的数据进行预测时，将会引发严重的问题。图8.8(a)给出了一个回归中的过拟合示例，其中模型参数是通过最大似然估计（见第8.3.1节）学习的。

我们将在第9.2.2节中更详细地讨论回归中的过拟合问题。

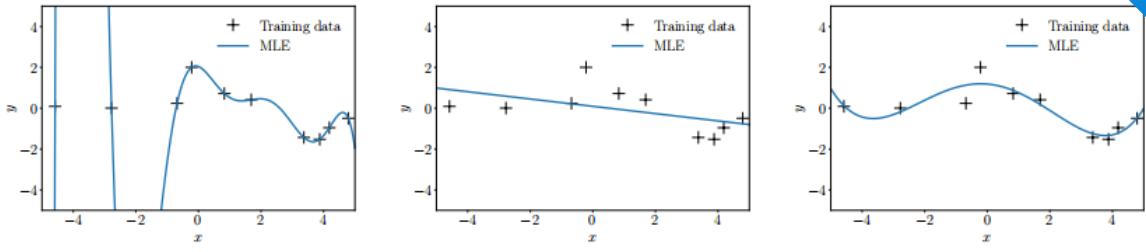


图8.8对不同模型类的拟合（通过最大似然值）到一个回归数据集。

当我们遇到欠拟合时，我们遇到了相反的问题，即模型类 M_θ 不够丰富。例如，如果我们的数据集是由正弦函数生成的，但 θ 只参数化直线，那么即使是最优的优化过程也无法使我们接近真实模型。然而，我们仍然会优化参数并找到建模数据集的最佳直线。图8.8(b)展示了一个因灵活性不足而欠拟合的模型示例。欠拟合的模型通常参数较少。

第三种情况是参数化模型类恰到好处。那么，我们的模型就拟合得很好，即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，可以描述给定的数据集。图8.8(c)展示了一个相当好地拟合给定数据集的模型。理想情况下，这是我们希望使用的模型类，因为它具有良好的泛化性能。

在实践中，我们经常定义非常丰富的模型类 M_θ ，其中包含许多参数，如深度神经网络。为了减轻过拟合问题，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节中讨论如何选择模型类。

8.3.4 拓展阅读

在考虑概率模型时，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章中详细讨论这一点。当将预测器限制为具有线性形式，并对输出应用额外的非线性函数 φ 时，即，

(8.21)

$$p(y_n | \mathbf{x}_n, \boldsymbol{\theta}) = \varphi(\boldsymbol{\theta}^\top \mathbf{x}_n),$$

我们可以考虑用于其他预测任务的其他模型，如二元分类或计数数据建模（McCullagh 和 Nelder, 1989）。另一种观点是考虑来自指数族（第6.6节）的似然

函数。这类模型在参数和数据之间具有线性依赖关系，并可能具有非线性变换 φ （称为链接函数），被称为广义线性模型（Agresti, 2002, 第4章）。

最大似然估计具有悠久的历史，最初由罗纳德·费希尔爵士（Sir Ronald Fisher）在20世纪30年代提出。我们将在第8.4节中进一步阐述概率模型的思想。在使用概率模型的研究者中，一个争论点是贝叶斯统计和频率统计之间的讨论。如第6.1.1节所述，这归根结底是概率的定义问题。回想第6.1节，我们可以将概率视为逻辑推理（通过允许不确定性）的推广（Cheeseman, 1985; Jaynes, 2003）。最大似然估计方法在本质上是频率的，有兴趣的读者可以参阅Efron和Hastie（2016）以获得对贝叶斯和频率统计的均衡观点。

有些概率模型中，可能无法使用最大似然估计。读者可以参阅更高级的统计教材，如Casella和Berger（2002），了解其他方法，如矩估计法、 M -估计法和估计方程法。

< 上一章节

下一章节 >

8.2 经验风险最小化

8.4 概率建模与推断



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，例如对未来事件的预测和决策制定。为了使这项任务更易于处理，我们通常会构建模型来描述生成观测数据的生成过程。

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 ，它作为伯努利分布（第6章）的参数，描述了出现“正面”的概率；其次，我们可以从伯努利分布 $p(x | \mu) = \text{Ber}(\mu)$ 中抽取一个结果 $x \in \{\text{head}, \text{tail}\}$ 。参数 μ 产生了特定的数据集 χ ，并且取决于所使用的硬币。由于 μ 是未知的，且永远无法直接观测到，因此我们需要机制来根据抛硬币实验的观察结果来学习关于 μ 的信息。接下来，我们将讨论如何使用概率建模来实现这一目的。

8.4.1 概率模型

概率模型将实验中的不确定部分表示为概率分布。使用概率模型的好处在于，它们通过概率论提供了一套统一且一致的工具集，这些工具集包括随机变量（第6章），用于建模、推断、预测和模型选择。

在概率建模中，观测变量 x 和隐藏参数 θ 的联合分布 $p(x, \theta)$ 至关重要：它包含了以下信息：

- 先验和似然（乘积规则，第6.3节）。
- 边缘似然 $p(x)$ ，在模型选择（第8.6节）中扮演重要角色，可以通过联合分布并积分掉参数来计算（求和规则，第6.3节）。
- 后验分布，可以通过将联合分布除以边缘似然来获得。

只有联合分布具有这样的性质。因此，概率模型是由其所有随机变量的联合分布来指定的。

8.4.2 贝叶斯推断

机器学习中的一个关键任务是利用模型和数据，在给定观测变量 x 的情况下，揭示模型隐藏变量 θ 的值。在第8.3.1节中，我们已经讨论了使用最大似然估计或最大后验估

计来估计模型参数 θ 的两种方法。在这两种情况下，我们都获得了 θ 的一个最佳单一值，因此参数估计的关键算法问题是解决一个优化问题。一旦这些点估计 θ^* 已知，我们就使用它们来进行预测。更具体地说，预测分布将是 $p(\mathbf{x} | \boldsymbol{\theta}^*)$ ，其中我们在似然函数中使用 $\boldsymbol{\theta}^*$ 。

正如第6.3节所讨论的，仅关注后验分布的某些统计量（如使后验最大化的参数 $\boldsymbol{\theta}^*$ ）会导致信息丢失，这在使用预测 $p(\mathbf{x} | \boldsymbol{\theta}^*)$ 来做决策的系统中可能是至关重要的。这些决策系统通常具有与似然函数不同的目标函数，例如平方误差损失或分类错误率。因此围绕参数的后验分布可以非常有用，并导致更稳健的决策。贝叶斯推断就是寻找这个后验分布（Gelman等，2004）。对于数据集 \mathcal{X} 、参数先验 $p(\boldsymbol{\theta})$ 和似然函数，后验分布

$$p(\boldsymbol{\theta} | \mathcal{X}) = \frac{p(\mathcal{X} | \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{X})}, \quad p(\mathcal{X}) = \int p(\mathcal{X} | \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta},$$

(8.22)

是通过应用贝叶斯定理获得的。关键思想是利用贝叶斯定理来反转参数 $\boldsymbol{\theta}$ 和数据 \mathcal{X} （由似然函数给出）之间的关系，以获得后验分布 $p(\boldsymbol{\theta} | \mathcal{X})$ 。

参数后验分布的意义在于，它可以用来将参数的不确定性传播到数据上。更具体地说，如果我们有参数上的分布 $p(\boldsymbol{\theta})$ ，那么我们的预测将是

$$p(\mathbf{x}) = \int p(\mathbf{x} | \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta} = \mathbb{E}_{\boldsymbol{\theta}}[p(\mathbf{x} | \boldsymbol{\theta})],$$

(8.23)

并且这些预测不再依赖于模型参数 $\boldsymbol{\theta}$ ，因为 $\boldsymbol{\theta}$ 已经被边缘化/积分掉了。方程(8.23)表明，预测是所有合理参数值 $\boldsymbol{\theta}$ 上的平均值，其中合理性由参数分布 $p(\boldsymbol{\theta})$ 所体现。

在8.3节中讨论了参数估计，并在此处讨论了贝叶斯推断，让我们比较这两种学习方法。通过最大似然估计或最大后验估计（MAP）进行的参数估计会产生参数的一致点估计 $\boldsymbol{\theta}^*$ ，需要解决的关键计算问题是优化。相比之下，贝叶斯推断产生了一个（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是直接的，而在贝叶斯框架下进行预测则需要解决另一个积分问题；参见(8.23)。然而，贝叶斯推断为我们提供了一种有原则的方法来整合先验知识、考虑辅助信息，并融入结构知识，这在参数估计的背景下并不容易实现。此外，在数据高效学习的背景下，将参数不确定性传播到预测中对于决策系统中的风险评估和探索非常有价值（Deisenroth等，2015；Kamthe和Deisenroth，2018）。



虽然贝叶斯推断是一个在数学上有原则的参数学习和预测框架，但由于我们需要解决的积分问题，它也存在一些实际挑战；参见(8.22)和(8.23)。更具体地说，如果我们没有为参数选择共轭先验（第6.6.1节），则(8.22)和(8.23)中的积分在解析上不可处理，我们无法以闭式形式计算后验分布、预测或边缘似然。在这些情况下，我们需要诉诸于近似方法。在这里，我们可以使用随机近似，如马尔可夫链蒙特卡洛（MCMC）（Gilks等，1996），或使用确定性近似，如拉普拉斯近似（Bishop，2006；Barber，2012；Murphy，2012）、变分推断（Jordan等，1999；Blei等，2017）或期望传播（Minka，2001a）。

尽管存在这些挑战，但贝叶斯推断已成功应用于各种问题，包括大规模主题建模（Hoffman等，2013）、点击率预测（Graepel等，2010）、控制系统中的数据高效强化学习（Deisenroth等，2015）、在线排名系统（Herbrich等，2007）和大规模推荐系统。还有一些通用工具，如贝叶斯优化（Brochu等，2009；Snoek等，2012；Shahriari等，2016），它们是高效搜索模型或算法元参数时非常有用的组成部分。

备注。在机器学习文献中，“变量”和“参数”之间可能存在一些任意的区别。通常，参数是通过估计得到的（例如，通过最大似然估计），而变量通常会被边缘化。在这本书中，我们对这种区分并不那么严格，因为原则上，我们可以对任何参数设置先验并将其积分出来，这将根据上述区分将该参数转变为随机变量。

8.4.3 隐变量模型

在实际应用中，除了模型参数 θ 外，将额外的隐变量 z （隐变量）作为模型的一部分是有用的（Moustaki等，2015）。这些隐变量与模型参数 θ 不同，因为它们不显式地对模型进行参数化。隐变量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结构，使我们能够定义更简单且更丰富的模型结构。模型结构的简化通常伴随着模型参数数量的减少（Paquet，2008；Murphy，2012）。隐变量模型中的学习（至少通过最大似然估计）可以通过一种有原则的方式使用期望最大化（EM）算法来完成（Dempster等，1977；Bishop，2006）。在这些隐变量有助于的场景中，例子包括用于降维的主成分分析（第10章）、用于密度估计的高斯混合模型（第11章）、用于时间序列建模的隐马尔可夫模型（Maybeck，1979）或动态系统（Ghahramani和Roweis，1999；Ljung，1999），以及元学习和任务泛化（Hausman等，2018；Sæmundsson等，2018）。虽然引入这些隐变量可能会使模型结构和生成过程变得更简单，但隐变量模型中的学习通常很难，我们将在第11章中看到这一点。由于隐变量模型还允许我们定义从参数生成数据的过程，让我们来看

一下这个生成过程。用 \mathbf{x} 表示数据， $\boldsymbol{\theta}$ 表示模型参数， \mathbf{z} 表示隐变量，我们得到条件分布

(8.24)

$$p(\mathbf{x} | \mathbf{z}, \boldsymbol{\theta})$$

该分布允许我们为任何模型参数和隐变量生成数据。由于 \mathbf{z} 是隐变量，我们对它们放置了一个先验 $p(\mathbf{z})$ 。与我们之前讨论的模型一样，具有隐变量的模型可以在我们在8.3节和8.4.2节中讨论的框架内用于参数学习和推断。为了促进学习（例如，通过最大似然估计或贝叶斯推断），我们遵循一个两步过程。首先，我们计算模型的似然 $p(\mathbf{x} | \boldsymbol{\theta})$ ，它不依赖于隐变量。其次，我们使用此似然进行参数估计或贝叶斯推断，其中我们分别使用与8.3节和8.4.2节中完全相同的表达式。

由于似然函数 $p(\mathbf{x} | \boldsymbol{\theta})$ 是在给定模型参数下数据的预测分布，我们需要对隐变量进行边缘化，以便

$$p(\mathbf{x} | \boldsymbol{\theta}) = \int p(\mathbf{x} | \mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})d\mathbf{z},$$

(8.25)

其中 $p(\mathbf{x} | \mathbf{z}, \boldsymbol{\theta})$ 在(8.24)中给出，且 $p(\mathbf{z})$ 是隐变量的先验。请注意，似然不应涉及 \mathbf{z} ，而只是数据 \mathbf{x} 和模型参数 $\boldsymbol{\theta}$ 的函数。

(8.25)中的似然直接允许通过最大似然估计进行参数估计。如8.3.2节所述，对模型参数 $\boldsymbol{\theta}$ 附加一个先验分布后，MAP估计也变得直接明了。此外，使用(8.25)中的似然，隐变量模型中的贝叶斯推断（8.4.2节）以常规方式进行：我们对模型参数放置一个先验 $p(\boldsymbol{\theta})$ ，并使用贝叶斯定理获得给定数据集 \mathcal{X} 之后的后验分布。

(8.26)

$$p(\boldsymbol{\theta} | \mathcal{X}) = \frac{p(\mathcal{X} | \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{X})}$$

在贝叶斯推断框架内，(8.26)中的后验可用于进行预测；参见(8.23)。在这个隐变量模型中，我们面临的一个挑战是，似然 $p(\mathcal{X} | \boldsymbol{\theta})$ 需要根据(8.25)对隐变量进行边缘化。除非我们选择 $p(\mathbf{x} | \mathbf{z}, \boldsymbol{\theta})$ 的共轭先验 $p(\mathbf{z})$ ，否则(8.25)中的边缘化在解析上不可处理，我们需要求助于近似方法（Bishop, 2006; Paquet, 2008; Murphy, 2012; Moustaki等, 2015）

类似于参数后验 (8.26) , 我们可以根据以下公式计算隐变量的后验:

$$p(\mathbf{z} \mid \mathcal{X}) = \frac{p(\mathcal{X} \mid \mathbf{z})p(\mathbf{z})}{p(\mathcal{X})}, \quad p(\mathcal{X} \mid \mathbf{z}) = \int p(\mathcal{X} \mid \mathbf{z}, \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta},$$

其中, $p(\mathbf{z})$ 是隐变量的先验, 而 $p(\mathcal{X} \mid \mathbf{z})$ 需要我们对模型参数 $\boldsymbol{\theta}$ 进行积分。鉴于解析求解积分的困难性, 显然, 在一般情况下, 同时边缘化隐变量和模型参数是不可能的 (Bishop, 2006; Murphy, 2012) 。一个更容易计算的量是给定模型参数条件下的隐变量后验分布, 即:

$$p(\mathbf{z} \mid \mathcal{X}, \boldsymbol{\theta}) = \frac{p(\mathcal{X} \mid \mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})}{p(\mathcal{X} \mid \boldsymbol{\theta})},$$

(8.28) 其中, $p(\mathbf{z})$ 是隐变量的先验, 而 $p(\mathcal{X} \mid \mathbf{z}, \boldsymbol{\theta})$ 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和高斯混合模型中隐变量的后验分布 (8.28) 。备注。在后续章节中, 我们可能不会在隐变量 \mathbf{z} 和不确定的模型参数 $\boldsymbol{\theta}$ 之间做出如此清晰的区分, 并且也会将模型参数称为“隐变量”或“隐藏变量”, 因为它们也是不可观测的。在第10章和第11章中, 当我们使用隐变量 \mathbf{z} 时, 我们会注意到这一点, 因为我们将在两种不同类型的隐藏变量: 模型参数 $\boldsymbol{\theta}$ 和隐变量 \mathbf{z} 。◊ 我们可以利用概率模型中所有元素都是随机变量的这一事实, 来定义一种统一的表示语言。在第8.5节中, 我们将看到一种简洁的图形语言, 用于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 进一步阅读

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的回顾。给定一个概率模型, 我们或许足够幸运, 能够用解析方法来计算感兴趣的参数。然而, 一般来说, 解析解是罕见的, 因此通常使用计算方法, 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) 为潜在变量模型中的贝叶斯推断提供了很好的概述。

近年来, 提出了几种编程语言, 旨在将软件中定义的变量视为与概率分布相对应的随机变量。其目标是能够编写概率分布的复杂函数, 同时在底层由编译器自动处理贝叶斯推断的规则。这个快速发展的领域被称为概率编程。



< 上一章节

下一章节 >

8.3 参数估计

8.5 有向图模型



8.5 有向图模型

在本节中，我们介绍了一种用于指定概率模型的图形语言，称为有向图模型。它提供了一种紧凑且简洁的方式来指定概率模型，并允许读者直观地解析随机变量之间的依赖关系。图形模型以可视化的方式捕捉了所有随机变量的联合分布如何被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，我们将概率模型的联合分布确定为关键关注量，因为它包含了关于先验、似然和后验的信息。然而，联合分布本身可能相当复杂，并且它没有告诉我们关于概率模型结构特性的任何信息。例如，联合分布 $p(a, b, c)$ 并没有告诉我们关于独立关系的信息。这正是图形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的概念。

在图形模型中，节点是随机变量。在图8.9(a)中，节点代表随机变量 a, b, c 。边代表变量之间的概率关系，例如条件概率。

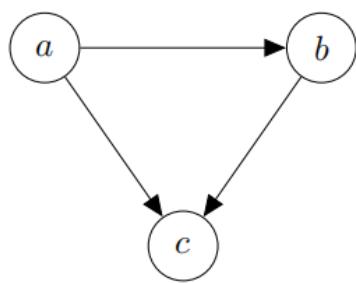
备注：并非每个分布都可以用特定类型的图形模型来表示。关于此点的讨论可以在 Bishop (2006) 中找到。

概率图形模型具有一些方便的特性：

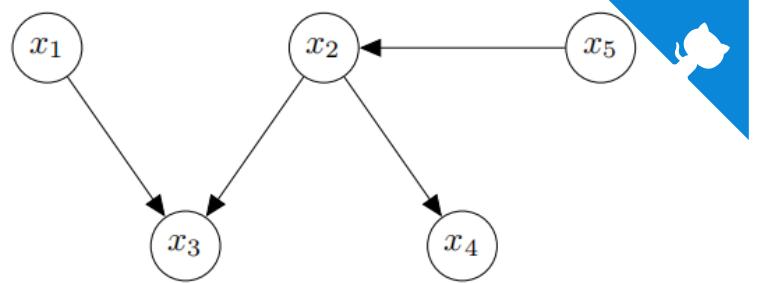
- 它们是可视化概率模型结构的一种简单方式。
- 它们可用于设计或激励新型统计模型。
- 仅通过检查图形，我们就可以洞察其属性，例如条件独立性。
- 统计模型中推断和学习的复杂计算可以表达为图形操作。

8.5.1 图形语义

有向图模型/贝叶斯网络是一种在概率模型中表示条件依赖性的方法。它们通过图形化的方式描述条件概率，从而为描述复杂的相互依赖关系提供了一种简洁的语言。模块化的描述也带来了计算上的简化。假设之间（即两个节点或随机变量之间）的有向链接（箭头）表示条件概率。例如，在图8.9(a)中， a 和 b 之间的箭头给出了在给定 a 的条件下 b 的条件概率 $p(b | a)$ 。



(a) Fully connected.



(b) Not fully connected.

图8.9有向图形化模型的示例。

如果我们知道联合分布的因式分解情况，那么就可以从联合分布推导出有向图模型。

示例 8.7

考虑三个随机变量 a, b, c 的联合分布

(8.29)

$$p(a, b, c) = p(c \mid a, b)p(b \mid a)p(a)$$

联合分布在(8.29)中的因式分解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 。

根据(8.29)中的因式分解，我们得到了图8.9(a)中的有向图模型。

一般来说，我们可以从联合分布的因式分解中构造出相应的有向图模型，具体步骤如下：

1. 为所有随机变量创建一个节点。
2. 对于每个条件分布，我们在图中从该分布所依赖的变量对应的节点出发，添加一个有向链接（箭头）。

图的布局取决于联合分布的因式分解的选择。

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，并描述如何从给定的图形模型中提取一组随机变量的联合分布。

示例 8.8

观察图8.9(b)中的图形模型，我们利用了两个属性：

- 我们所寻求的联合分布 $p(x_1, \dots, x_5)$ 是一组条件概率的乘积，图中每个节点对应一个条件概率。在这个特定示例中，我们需要五个条件概率。
- 每个条件概率仅依赖于图中对应节点的父节点。例如， x_4 将依赖于 x_2 。

这两个属性给出了联合分布的所需因式分解：

(8.30)

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_5)p(x_2 | x_5)p(x_3 | x_1, x_2)p(x_4 | x_2).$$

一般来说，联合分布 $p(\mathbf{x}) = p(x_1, \dots, x_K)$ 可以表示为

(8.31)

$$p(\mathbf{x}) = \prod_{k=1}^K p(x_k | \mathbf{Pa}_k),$$

其中 \mathbf{Pa}_k 表示“ x_k 的父节点”。 x_k 的父节点是指有箭头指向 x_k 的节点。

我们以抛硬币实验的具体示例来结束这一小节。考虑一个伯努利实验（示例6.8），其中该实验的结果 x 为“正面”的概率为

(8.32)

$$p(x | \mu) = \text{Ber}(\mu).$$

现在我们重复这个实验 N 次，并观察结果 x_1, \dots, x_N ，从而得到联合分布

(8.33)



$$p(x_1, \dots, x_N | \mu) = \prod_{n=1}^N p(x_n | \mu).$$

由于实验是独立的，因此等式右侧是每个单独结果的伯努利分布的乘积。回顾6.4.5节，统计独立性意味着分布可以因式分解。为了为这种情况编写图形模型，我们需要区分未观测/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)中的图形模型。我们看到，单个参数 μ 对于所有 $x_n, n = 1, \dots, N$ 都是相同的，因为结果 x_n 是同分布的。对于这种情况，一个更紧凑但等效的图形模型如图8.10(b)所示，其中我们使用了板块符号。板块（框）重复其内部的所有内容

（在本例中，重复观测 x_n ） N 次。因此，这两个图形模型是等效的，但板块符号更为紧凑。图形模型使我们能够立即在 μ 上放置一个超先验。超先验是第一层先验参数上的先验分布的第二层超先验。图8.10(c)在潜在变量 μ 上放置了一个Beta(α, β)先验。如果我们将 α 和 β 视为确定性参数（即不是随机变量），则省略其周围的圆圈。

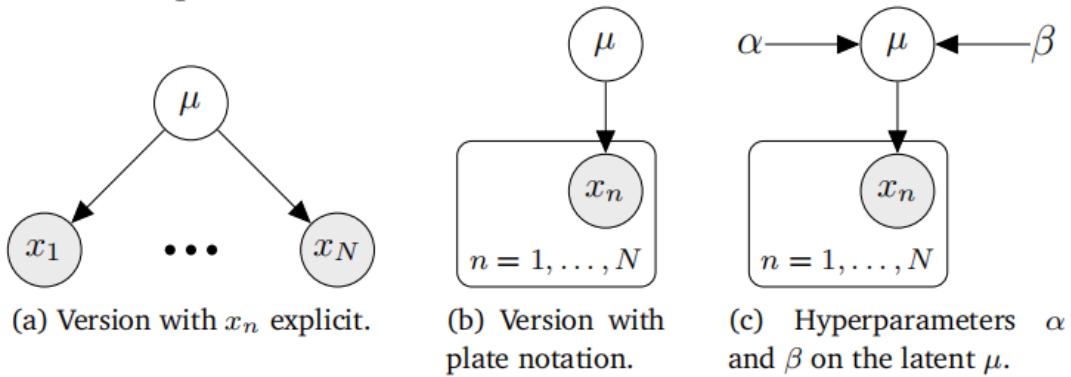


图8.10重复伯努利实验的图形模型。

8.5.2 条件独立性和d-分离

有向图模型允许我们仅通过查看图形来找到联合分布的条件独立性（第6.4.5节）关系属性。这其中的关键是一个称为d-分离（Pearl, 1988）的概念。

考虑一个一般的有向图，其中 \mathcal{A} 、 \mathcal{B} 、 \mathcal{C} 是节点的不相交集合（它们的并集可能小于图中节点的完整集合）。我们希望确定给定的有向无环图是否隐含了特定的条件独立性陈述，“给定 \mathcal{C} ， \mathcal{A} 与 \mathcal{B} 条件独立”，表示为

(8.34)

$$\mathcal{A} \perp \mathcal{B} | \mathcal{C},$$

为此，我们考虑从 \mathcal{A} 中的任何节点到 \mathcal{B} 中任何节点的所有可能的轨迹（忽略箭头方向的路径）。如果这样的路径包含任何节点，使得以下任一条件为真，则该路径被认为被阻塞的：

- 路径上的箭头在节点处相遇，要么是头到尾，要么是尾到尾，并且该节点在集合 \mathcal{C} 中。
- 箭头在节点处头到头相遇，并且该节点及其任何后代都不在集合 \mathcal{C} 中（注意这里原文中集合名称应为 \mathcal{C} ，可能是笔误）。

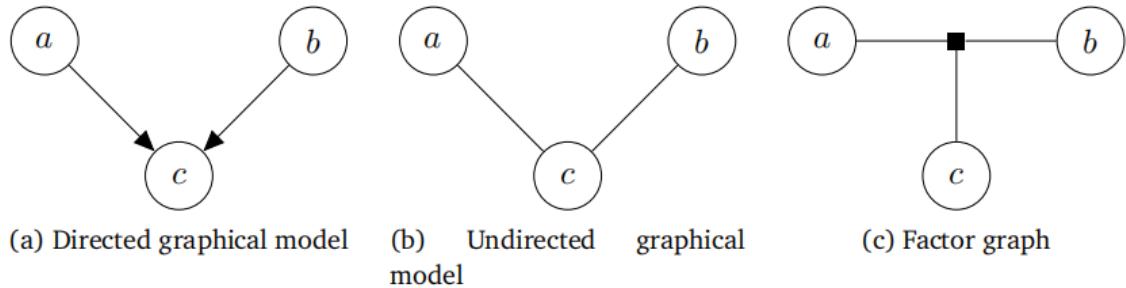


图8.12三种类型的图形模型：(a)有向图形模型（贝叶斯网络）；(b)无向图形模型（马尔可夫随机场）；(c)因子图。

如果所有路径都被阻塞，则称 \mathcal{A} 与 \mathcal{B} 被 \mathcal{C} d-分离，图中所有变量的联合分布将满足 $\mathcal{A} \perp \mathcal{B} | \mathcal{C}$ 。

例8.9 条件独立性

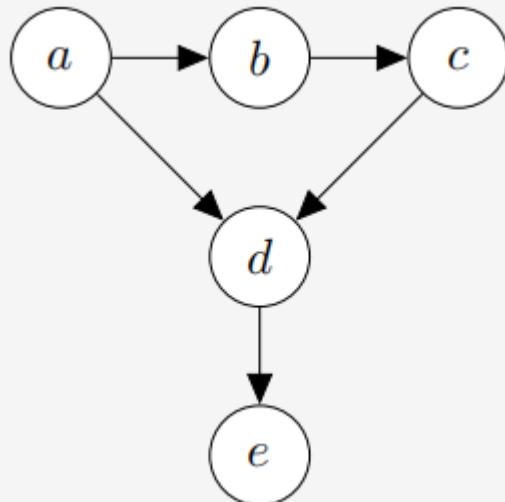


图8.11D-分离示例。

观察图8.11中的图模型。视觉信息告诉我们：

$$\begin{aligned} b \perp d &| a, c \\ a \perp c &| b \\ b \not\perp d &| c \\ a \not\perp c &| b, e \end{aligned}$$

有向图模型允许对概率模型进行紧凑的表示，我们将在第9、10和11章中看到有向图模型的例子。这种表示方式，结合条件独立性的概念，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所做的设计选择对模型结构的影响。我们通常需要对模型的结构做出高级假设。这些建模假设（超参数）会影响预测性能，但无法直接使用我们目前所看到的方法来选择。我们将在第8.6节中讨论选择结构的不同方法。

8.5.3 拓展阅读

Bishop (2006, 第8章) 提供了概率图模型的入门介绍，而Koller和Friedman (2009) 的书籍则对不同应用及其相应的算法影响进行了详尽描述。概率图模型主要分为以下三种类型：

- 有向图模型（贝叶斯网络）；见图8.12(a)
- 无向图模型（马尔可夫随机场）；见图8.12(b)
- 因子图；见图8.12(c)

图模型允许使用基于图的算法进行推理和马尔可夫随机学习，例如通过局部消息传递。其应用范围广泛，从在线游戏中的秩因子分解 (Herbrich et al., 2007) 和计算机视觉（如图像分割、语义标注、图像去噪、图像恢复 (Kittler和Föglein, 1984; Sucar和Gillies, 1994; Shotton et al., 2006; Szeliski et al., 2008)) 到编码理论 (McEliece et al., 1998) 、线性方程组求解 (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和Mohamed, 2012) 。

本书未讨论但在实际应用中特别重要的一个话题是结构预测 (Bakir et al., 2007; Nowozin et al., 2014) 的概念，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应用，从而产生了许多结构模型的有用应用（Goodfellow et al., 2016, 第16章）。近年来，概率图模型在因果推断领域也重新获得了关注（Pearl, 2009; Imbens和Rubin, 2015; Peters et al., 2017; Rosenbaum, 2017）。

< 上一章节

下一章节 >

8.4 概率建模与推断

8.6 模型选择



8.6 模型选择

在机器学习中，我们往往需要做出高层次的建模决策，这些决策对模型的性能有着至关重要的影响。我们所做的选择（例如，似然函数的形式）会影响模型中自由参数的数量和类型，进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加灵活，因为它们能够用来描述更多的数据集。例如，一个1次多项式（即一条直线 $y = a_0 + a_1x$ ）只能用来描述输入 x 和观测值 y 之间的线性关系。而通过将 a_2 设为0（即二次项系数为0），我们可以得到一个2次多项式，它除了能描述线性关系外，还能描述输入和观测值之间的二次关系。

现在，人们可能会认为，由于非常灵活的模型更具表达力，因此它们通常比简单的模型更受青睐。但一个普遍的问题是，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，我们真正关心的并不是模型在训练集上的表现。在第8.3节中，我们已经看到，最大似然估计可能会导致过拟合，尤其是在训练数据集较小时。理想情况下，我们的模型（也）应该在测试集上表现良好（而测试集在训练时是不可用的）。因此，我们需要一些机制来评估模型对未见过的测试数据的泛化能力。模型选择正是关注于这一问题。

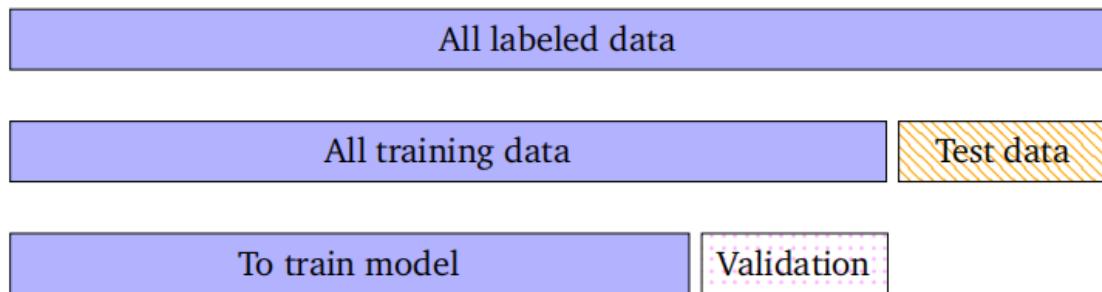


图8.13：嵌套的交叉验证。我们进行了两个层次的k倍交叉验证。

8.6.1 嵌套交叉验证

我们已经看到了一种（在第8.2.4节中的交叉验证）可用于模型选择的方法。回顾一下，交叉验证通过反复将数据集拆分为训练集和验证集来估计泛化误差。我们可以再次应用这个想法，即对于每次拆分，我们可以再进行一轮交叉验证。这有时被称为嵌套交叉验证；见图8.13。内层用于估计在内部验证集上特定模型或超参数选择的性

能。外层则用于估计内层循环选择的最佳模型选择方案的泛化性能。我们可以在内层循环中测试不同的模型和超参数选择。为了区分这两个层次，通常将用于估计泛化性能的集合称为测试集，而将用于选择最佳模型的集合称为验证集。内层循环通过在验证集上的经验误差来近似给定模型的泛化误差的期望值（8.39），即：

$$\mathbb{E}_{\mathcal{V}}[\mathbf{R}(\mathcal{V} | M)] \approx \frac{1}{K} \sum_{k=1}^K \mathbf{R}(\mathcal{V}^{(k)} | M),$$

(8.39)

其中， $\mathbf{R}(\mathcal{V} | M)$ 是模型 M 在验证集 \mathcal{V} 上的经验风险（例如，均方根误差）。我们对所有模型重复此过程，并选择表现最佳的模型。请注意，交叉验证不仅为我们提供了预期的泛化误差，我们还可以获得高阶统计量，例如标准误差，它是对均值估计的不确定性的估计。一旦选择了模型，我们就可以在测试集上评估其最终性能。

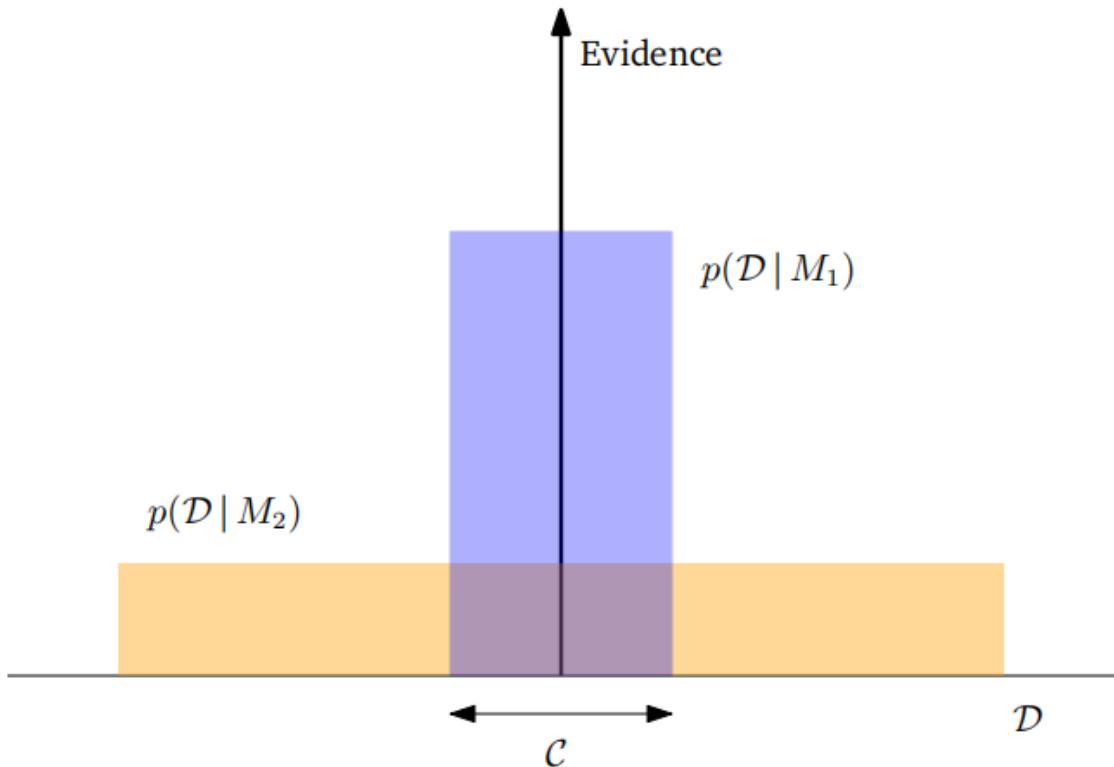


图8.14贝叶斯推理体现了奥卡姆剃刀原则。横轴描述了所有可能的数据集的空间 \mathbf{d} 。证据（纵轴）评估了一个模型对可用数据的预测程度。由于 $p(\mathbf{D} | M_i)$ 需要集成到1中，所以我们应该选择证据最大的模型。改编自MacKay (2003)。

8.6.2 贝叶斯模型选择



模型选择有许多方法，本节将介绍其中一些。一般来说，它们都在尝试在模型复杂度和数据拟合度之间做出权衡。我们假设简单模型比复杂模型更不易过拟合，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。这个概念也被称为奥卡姆剃刀原则。

奥卡姆剃刀原则

备注：如果我们把模型选择视为一个假设检验问题，那么我们正在寻找的是与数据一致的最简单假设（Murphy, 2012）。

◇

有人可能会考虑在模型上放置一个先验，以偏好更简单的模型。然而，这并非必要：在贝叶斯概率的应用中，“自动奥卡姆剃刀”是定量体现的（Smith 和 Spiegelhalter, 1980; Jefferys 和 Berger, 1992; MacKay, 1992）。图8.14（改编自MacKay, 2003）给出了一个基本直觉，解释了为什么复杂且极具表达力的模型在建模给定数据集 \mathcal{D} 时可能不是一个较优选择。让我们将水平轴视为代表所有可能数据集 \mathcal{D} 的空间的预测。如果我们关注的是给定数据 \mathcal{D} 下模型 M_i 的后验概率 $p(M_i | \mathcal{D})$ 的量化表示，我们可以使用贝叶斯定理。假设所有模型上的先验 $p(M)$ 是均匀的，贝叶斯定理会根据模型预测已发生数据的程度来奖励模型，即需要整合/求和到1。

给定模型 M_i 下数据 \mathcal{D} 的预测，即 $p(\mathcal{D} | M_i)$ ，被称为 M_i 的证据。一个简单的模型 M_1 只能预测一小部分数据集，这由 $p(\mathcal{D} | M_1)$ 表示；一个更强大的模型 M_2 （例如，具有比 M_1 更多的自由参数）能够预测更多种类的数据集。然而，这意味着 M_2 在区域 C 中对数据集的预测不如 M_1 。假设这两个模型的先验概率是相等的。那么，如果数据集落在区域 C 中，则较弱的模型 M_1 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，即应该有一种方法可以从给定模型中生成数据。此外，如果模型已经从数据中得到了适当的学习，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理问题是很有帮助的，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 $M = \{M_1, \dots, M_K\}$ ，其中每个模型 M_k 都拥有参数 θ_k 。在贝叶斯模型选择中，我们在模型集上放置一个先验 $p(M)$ 。允许我们从该模型生成数据的相应生成过程是

$$\begin{aligned} M_k &\sim p(M) \\ \theta_k &\sim p(\theta | M_k) \\ \mathcal{D} &\sim p(\mathcal{D} | \theta_k) \end{aligned}$$

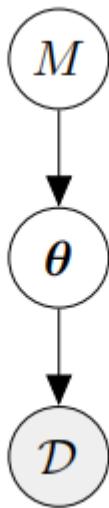


图8.15贝叶斯模型选择中的层次生成过程说明。我们在一组模型上放置一个先验的 $p(M)$ 。对于每个模型，在相应的模型参数上都有一个分布 $p(\theta | M)$ ，用于生成数据 D 。

并且如图8.15所示。给定一个训练集 \mathcal{D} ，我们应用贝叶斯定理并计算模型上的后验分布为

(8.43)

$$p(M_k | \mathcal{D}) \propto p(M_k)p(\mathcal{D} | M_k).$$

注意，这个后验分布不再依赖于模型参数 θ_k ，因为在贝叶斯设置中它们已经被积分掉了，即

$$p(\mathcal{D} | M_k) = \int p(\mathcal{D} | \theta_k)p(\theta_k | M_k)d\theta_k,$$

(8.44)

其中 $p(\theta_k | M_k)$ 是模型 M_k 的参数 θ_k 的先验分布。(8.44)项被称为模型证据或边缘似然。从(8.43)中的后验分布中，我们确定最大后验(MAP)估计

(8.45)

$$M^* = \arg \max_{M_k} p(M_k | \mathcal{D}).$$

如果采用均匀先验 $p(M_k) = \frac{1}{K}$ ，即给予每个模型相等的（先验）概率，那么确定模型上的MAP估计就等价于选择使模型证据(8.44)最大化的模型。

备注（似然与边缘似然）：似然与边缘似然（证据）之间存在一些重要差异：虽然似然容易过拟合，但边缘似然通常不会，因为模型参数已经被边缘化（即我们不再需要拟合参数）。此外，边缘似然自动体现了模型复杂性和数据拟合度之间的权衡（奥卡姆剃刀原则）。

8.6.3 模型比较的贝叶斯因子

考虑在给定数据集 \mathcal{D} 的情况下，比较两个概率模型 M_1, M_2 的问题。如果我们计算后验概率 $p(M_1 | \mathcal{D})$ 和 $p(M_2 | \mathcal{D})$ ，则可以计算后验概率的比率

(8.46)

$$\underbrace{\frac{p(M_1 | \mathcal{D})}{p(M_2 | \mathcal{D})}}_{\text{后验比率}} = \frac{\frac{p(\mathcal{D} | M_1)p(M_1)}{p(\mathcal{D})}}{\frac{p(\mathcal{D} | M_2)p(M_2)}{p(\mathcal{D})}} = \underbrace{\frac{p(M_1)}{p(M_2)}}_{\text{先验比率}} \underbrace{\frac{p(\mathcal{D} | M_1)}{p(\mathcal{D} | M_2)}}_{\text{贝叶斯因子}}.$$

后验概率的比率也被称为后验比率。等式(8.46)右侧的第一个分数，即先验比率，衡量了我们的先验（初始）信念在多大程度上偏向于 M_1 而非 M_2 。边缘似然（右侧第二个分数）的比率被称为**贝叶斯因子**，它衡量了与 M_2 相比，数据 D 被 M_1 预测得有多好。

备注。杰弗里斯-林德利悖论指出，“由于复杂模型在先验分布较为分散的情况下数据概率将非常小，因此贝叶斯因子总是偏向于更简单的模型”（Murphy, 2012）。这里，扩散先验指的是一种不偏向特定模型的先验分布，即在该先验下，许多模型理论上都是合理的。

◇

如果我们选择模型上的均匀先验，则(8.46)中的先验比率项为1，即后验比率就是边缘似然（贝叶斯因子）的比率

$$\frac{p(\mathcal{D} | M_1)}{p(\mathcal{D} | M_2)}.$$

(8.47)

如果贝叶斯因子大于1，我们选择模型 M_1 ，否则选择模型 M_2 。与频率统计类似，关于在结果的“显著性”之前应考虑的比率大小，存在相应的指导原则。（Jeffreys, 1961）。



备注（计算边缘似然）。边缘似然在模型选择中扮演着重要角色：我们需要计算贝叶斯因子(8.46)和模型上的后验分布(8.43)。不幸的是，计算边缘似然需要我们求解一个积分(8.44)。这个积分通常无法解析求解，因此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方法的随机近似（Murphy, 2012），或贝叶斯Monte Carlo技术（O'Hagan, 1991; Rasmussen and Ghahramani, 2003）。

然而，也有一些特殊情况可以求解。在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 $p(\theta)$ ，则可以以闭合形式计算边缘似然。在第9章中，我们将在线性回归的上下文中正是这样做。



本章我们已经简要介绍了机器学习的基本概念。在本书的其余部分，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 拓展阅读

我们在本节的开头提到，存在一些高级建模选择，它们会影响模型的性能。这些例子包括：

- 回归设置中多项式的次数
- 混合模型中的组件数量
- （深度）神经网络的网络架构
- 支持向量机中的核函数类型
- 主成分分析（PCA）中潜在空间的维度
- 优化算法中的学习率（调度）

Rasmussen和Ghahramani（2001）指出，自动的奥卡姆剃刀原则并不一定会惩罚模型中的参数数量，但它确实在函数复杂度方面起作用。他们还表明，自动的奥卡姆剃刀原则也适用于具有许多参数的贝叶斯非参数模型，例如高斯过程。

如果我们关注最大似然估计，那么存在许多用于模型选择的启发式方法，这些方法可以阻止过拟合。它们被称为信息准则，我们选择具有最大值的模型。赤池信息量准则（AIC）（Akaike, 1974）

(8.48)

$$\log p(\mathbf{x} \mid \boldsymbol{\theta}) - M$$



通过添加一个惩罚项来补偿具有大量参数的更复杂模型的过拟合，从而校正最大似然估计的偏差。这里， M 是模型参数的数量。**AIC**估计了给定模型所损失的相对信息量。贝叶斯信息准则（**BIC**）（Schwarz, 1978）

(8.49)

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \approx \log p(\mathbf{x} \mid \boldsymbol{\theta}) - \frac{1}{2}M \log N$$

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**对模型复杂度的惩罚比**AIC**更重。

< 上一章节

下一章节 >

8.5 有向图模型

习题



404 - Not found

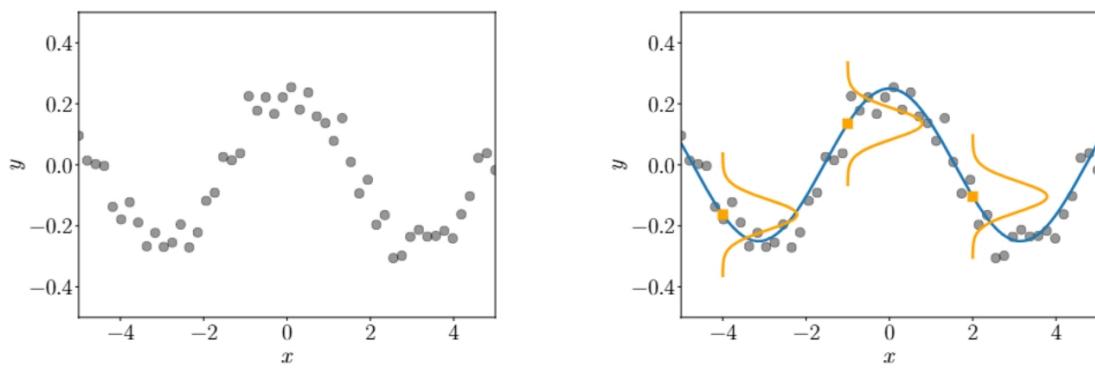


第9章 线性回归

在接下来的内容中，我们将应用第2章、第5章、第6章和第7章中的数学概念来解决线性回归（曲线拟合）问题。在线性回归中，我们的目标是找到一个函数 f ，它将输入 $\mathbf{x} \in \mathbb{R}^D$ 映射到对应的函数值 $f(\mathbf{x}) \in \mathbb{R}$ 。

我们假设给定一组训练输入 \mathbf{x}_n 以及对应的带噪声观测值： $y_n = f(\mathbf{x}_n) + \epsilon$ ，其中， ϵ 是一个独立同分布（i.i.d.）的随机变量，用于描述测量/观测噪声以及可能未被建模的过程（在本章中我们不会再进一步讨论这些）。在本章中，我们假设噪声是零均值高斯噪声。我们的任务是找到一个函数，该函数不仅能够拟合训练数据，而且能够很好地推广到预测训练数据之外的输入位置处的函数值（参见第8章）。图9.1展示了这样一个回归问题的示例。

典型的回归设置如图9.1(a)所示：对于某些输入值 \mathbf{x}_n ，我们观测到带噪声的函数值 $y_n = f(\mathbf{x}_n) + \epsilon$ 。任务是从这些数据中推断出生成数据的函数 f ，并使其能够很好地推广到新的输入位置处的函数值。图9.1(b)给出了一个可能的解决方案，我们还在其中展示了三个以函数值 $f(x)$ 为中心的分布，这些分布表示数据中的噪声。



(a) Regression problem: observed noisy function values from which we wish to infer the underlying function that generated the data.

(b) Regression solution: possible function that could have generated the data (blue) with indication of the measurement noise of the function value at the corresponding inputs (orange distributions).

图 9.1: (a) 数据集；(b) 回归问题可能的解决方法

回归是机器学习中的一个基础问题，回归问题出现在多种研究领域和应用中，包括时间序列分析（例如系统辨识）、控制与机器人学（例如强化学习、正向/逆向模型学习）、优化（例如线搜索、全局优化）以及深度学习应用（例如电脑游戏、语音转文本翻译、图像识别、自动视频标注）。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以下多种问题：

- **模型（类型）的选择以及回归函数的参数化**。给定一个数据集，哪些函数类别（例如多项式）是建模数据的良好候选，以及我们应该选择什么样的特定参数化（例如多项式的阶数）？正如第8.6节中讨论的模型选择那样，它允许我们比较各种模型，以找到能够合理地解释训练数据的最简单模型。
- **寻找良好的参数**。在选择了回归函数的模型之后，我们如何找到良好的模型参数？在这里，我们需要查看不同的损失/目标函数（它们决定了什么是“良好”的拟合）以及优化算法，这些算法允许我们最小化这个损失。
- **过拟合和模型选择**。当回归函数“过于完美”地拟合训练数据，但无法推广到未见的测试数据时，就会出现过拟合问题。过拟合通常发生在底层模型（或其参数化）过于灵活和富有表现力时；参见第8.6节。我们将探讨其背后的原因，并讨论如何减轻过拟合的影响。
- **损失函数与参数先验之间的关系**。损失函数（优化目标）通常是由概率模型激发并诱导的。我们将探讨损失函数与诱导这些损失的底层先验假设之间的联系。
- **不确定性建模**。在任何实际设置中，我们只有有限的（可能数量很大）训练数据用于选择模型类别和对应的参数。由于这有限的训练数据无法涵盖所有可能的情况，我们可能希望描述剩余的参数不确定性，以在测试时获得模型预测的置信度量；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提供了置信区间。

在接下来的内容中，我们将使用第3章、第5章、第6章和第7章中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，以找到最优的模型参数。利用这些参数估计，我们将简要探讨泛化误差和过拟合。在本章的最后，我们将讨论贝叶斯线性回归，它允许我们从更高层次上对模型参数进行推理，从而解决最大似然和MAP估计中遇到的一些问题。

< 上一章节

第八章 当模型遇上数据

下一章节 >

第十章 降维和主成分分析



9.1 问题形式化

由于观测噪声的存在，我们将采用概率方法，并明确使用似然函数来建模噪声。更具体地说，在本章中，我们考虑一个具有似然函数的回归问题：

$$p(y|\mathbf{x}) = \mathcal{N}(y|f(\mathbf{x}), \sigma^2) \quad (9.1)$$

其中， $\mathbf{x} \in \mathbb{R}^D$ 是输入， $y \in \mathbb{R}$ 是带噪声的函数值（目标值）。根据式(9.1)， \mathbf{x} 和 y 之间的函数关系可以表示为：

$$y = f(\mathbf{x}) + \epsilon \quad (9.2)$$

其中， $\epsilon \sim \mathcal{N}(0, \sigma^2)$ 是独立同分布 (i.i.d.) 的高斯测量噪声，其均值为 0，方差为 σ^2 。我们的目标是找到一个函数，使其尽可能接近生成数据的未知函数 f ，并且能够很好地推广。

在本章中，我们专注于参数化模型，即我们选择一个参数化的函数，并找到能够“良好”拟合数据的参数 $\boldsymbol{\theta}$ 。暂时假设噪声方差 σ^2 是已知的，专注于学习模型参数 $\boldsymbol{\theta}$ 。在线性回归中，我们考虑参数 $\boldsymbol{\theta}$ 线性出现在模型中的特殊情况。线性回归的一个例子是：

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{x}^\top \boldsymbol{\theta}, \sigma^2) \quad (9.3)$$

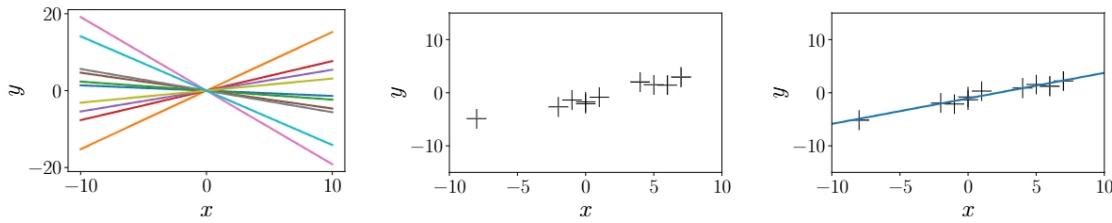
$$\iff y = \mathbf{x}^\top \boldsymbol{\theta} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \quad (9.4)$$

其中， $\boldsymbol{\theta} \in \mathbb{R}^D$ 是我们要求的参数。式 (9.4) 中描述的函数类别是通过原点的直线。在式 (9.4) 中，我们选择参数化 $f(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\theta}$ 。狄拉克 δ 函数 (δ 函数) 在除一个点之外的所有地方都为零，其积分值为 1。它可以被视为高斯分布的极限情况，当 $\sigma^2 \rightarrow 0$ 时。

似然函数 $p(y|\mathbf{x}, \boldsymbol{\theta})$ 是在 $\mathbf{x}^\top \boldsymbol{\theta}$ 处评估的 y 的概率密度函数。注意，唯一的不确定性来源来自于观测噪声（因为 \mathbf{x} 和 $\boldsymbol{\theta}$ 在式 (9.3) 中被认为是已知的）。如果没有观测噪声， \mathbf{x} 和 y 之间的关系将是确定性的，而式 (9.3) 将是一个狄拉克 δ 函数。

例9.1 对于 $x, \theta \in \mathbb{R}$ ，式(9.4)中的线性回归模型描述了直线（线性函数），

参数 θ 是直线的斜率。图9.2(a)展示了不同 θ 值对应的示例函数。



(a) Example functions (straight lines) that can be described using the linear model in (9.4).

(b) Training set.

(c) Maximum likelihood estimate.

图 9.2 线性回归示例。(a) 属于此类别的示例函数; (b) 训练集; (c) 极大似然估计。

式 (9.3)–(9.4) 中的线性回归模型不仅在参数上是线性的，而且在输入 \mathbf{x} 上也是线性的。图 9.2(a) 展示了这类函数的示例。我们稍后会看到， $y = \phi^\top(\mathbf{x})\boldsymbol{\theta}$ 对于非线性变换 ϕ 也是一个线性回归模型，因为“线性回归”指的是“参数线性”的模型，即通过输入特征的线性组合来描述函数。在这里，“特征”是输入 \mathbf{x} 的表示 $\phi(\mathbf{x})$ 。

在接下来的内容中，我们将更详细地讨论如何找到良好的参数 $\boldsymbol{\theta}$ ，以及如何评估一组参数是否“有效”。暂时假设噪声方差 σ^2 是已知的。

下一章节 >

9.2 参数估计



9.2 参数估计

考虑线性回归设置（式(9.4)），假设我们有一个训练集 $\mathcal{D} := \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ ，其中包含 N 个输入 $\mathbf{x}_n \in \mathbb{R}^D$ 和对应的观测值/目标值 $y_n \in \mathbb{R}$, $n = 1, \dots, N$ 。对应的概率图模型如图 9.3 所示。注意，给定各自的输入 $\mathbf{x}_i, \mathbf{x}_j$, y_i 和 y_j 是条件独立的，因此似然函数可以分解为：

$$p(\mathcal{Y}|\mathcal{X}, \boldsymbol{\theta}) = p(y_1, \dots, y_N | \mathbf{x}_1, \dots, \mathbf{x}_N, \boldsymbol{\theta}) \quad (9.5a)$$

$$= \prod_{n=1}^N p(y_n | \mathbf{x}_n, \boldsymbol{\theta}) = \prod_{n=1}^N \mathcal{N}(y_n | \mathbf{x}_n^\top \boldsymbol{\theta}, \sigma^2) \quad (9.5b)$$

其中，我们定义 $\mathbf{X} := \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 和 $\mathbf{Y} := \{y_1, \dots, y_N\}$ 分别为训练输入和对应的目标值集合。由于噪声分布，似然函数和因子 $p(y_n | \mathbf{x}_n, \boldsymbol{\theta})$ 是高斯分布的；见式 (9.3)。

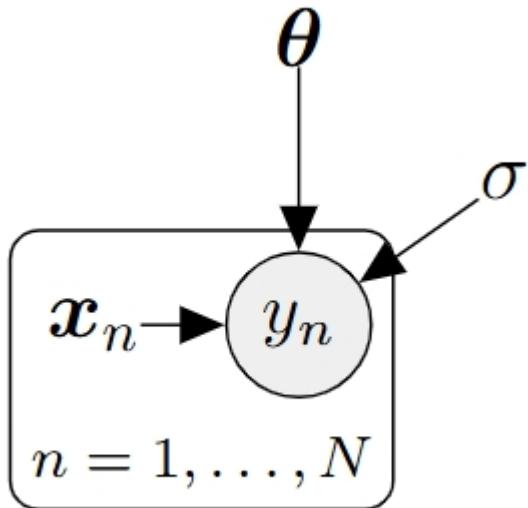


图 9.3 线性回归的概率图模型。已观测的随机变量用阴影表示，确定性/已知值则无圆圈标记。

在接下来的内容中，我们将讨论如何找到线性回归模型（式(9.4)）的最优参数 $\boldsymbol{\theta}^* \in \mathbb{R}^D$ 。一旦找到参数 $\boldsymbol{\theta}^*$ ，我们就可以使用这个参数估计值在式(9.4)中预测函数值，这样在任意测试输入 \mathbf{x}^* 处，对应目标值 y^* 的分布为：

$$p(y^* | \mathbf{x}^*, \boldsymbol{\theta}^*) = \mathcal{N}(y^* | \mathbf{x}^{*\top} \boldsymbol{\theta}^*, \sigma^2) \quad (9.6)$$

在接下来的内容中，我们将研究通过最大化似然来估计参数，我们在第8.3节中已经部分涉及了这个主题。

9.2.1 最大似然估计

一种广泛用于找到期望参数 θ_{ML} 的方法是**最大似然估计**，我们寻找能够最大化似然（式(9.5b)）的参数 θ_{ML} 。直观上，最大化似然意味着最大化给定模型参数时训练数据的预测分布。我们得到的最大似然参数为：

$$\theta_{ML} = \arg \max_{\theta} p(Y|X, \theta) \quad (9.7)$$

似然函数在参数上不是概率分布。

注释。似然 $p(y|x, \theta)$ 在 θ 上不是概率分布：它仅仅是参数 θ 的一个函数，但它并没有归一化（即，它不积分为1），并且可能甚至无法相对于 θ 进行积分。然而，式(9.7)中的似然是 y 的一个归一化概率分布。

为了找到能够最大化似然的期望参数 θ_{ML} ，我们通常执行梯度上升（或对负似然进行梯度下降）。然而，在我们这里考虑的线性回归中，存在一个闭式解，这使得迭代梯度下降变得不必要。实际上，我们不是直接最大化似然，而是对似然函数进行对数变换，并最小化负对数似然。

注释（对数变换）。由于似然（式(9.5b)）是 N 个高斯分布的乘积，对数变换很有用，因为（a）它不会受到数值下溢的影响，（b）微分规则将变得更简单。更具体地说，当我们把 N 个概率相乘时，数值下溢会成为一个问题，因为当 N 是数据点的数量时，我们无法表示非常小的数字，例如 10^{-256} 。此外，对数变换将乘积转换为对数概率之和，使得对应的梯度是对数概率的和，而不是应用乘积规则（式(5.46)）来计算 N 项乘积的梯度。

为了找到我们线性回归问题的最优参数 θ_{ML} ，我们最小化负对数似然：

\$\$

- $\log p(Y | X, \theta) = - \log \prod_{n=1}^N p(y_n | x_n, \theta) = - \sum_{n=1}^N \log p(y_n | x_n, \theta)$ \$\$

其中，我们利用了由于对训练集的独立性假设，似然（式(9.5b)）在数据点数量上是可分解的。在线性回归模型（式(9.4)）中，由于高斯加性噪声项，似然是高斯的，因此我们得到：

$$\log p(y_n|x_n, \theta) = -\frac{1}{2\sigma^2}(y_n - x_n^\top \theta)^2 + \text{常数} \quad (9.9)$$

其中常数包括所有与 θ 无关的项。将式(9.9)代入负对数似然 (式(9.8))，我们得到 (忽略常数项)：

$$\mathcal{L}(\theta) := \frac{1}{2\sigma^2} \sum_{n=1}^N (y_n - x_n^\top \theta)^2 \quad (9.10a)$$

$$= \frac{1}{2\sigma^2} \|y - X\theta\|^2 \quad (9.10b)$$

其中，我们定义设计矩阵 $X := [x_1, \dots, x_N]^\top \in \mathbb{R}^{N \times D}$ 作为训练输入的集合， $y := [y_1, \dots, y_N]^\top \in \mathbb{R}^N$ 作为收集所有训练目标值的向量。注意，设计矩阵 X 的第 n 行对应于训练输入 x_n 。在式(9.10b)中，我们利用了观测值 y_n 与对应模型预测 $x_n^\top \theta$ 之间的平方误差之和等于 y 和 $X\theta$ 之间的平方距离这一事实。

回忆第3.1节的内容，如果选择点积作为内积，则 $\|x\|^2 = x^\top x$ 。有了式(9.10b)，我们现在有了一个具体的负对数似然函数形式，需要对其进行优化。我们立即可以看出，式(9.10b)在 θ 上是二次的。这意味着我们可以找到一个唯一的全局解 θ_{ML} ，用于最小化负对数似然 \mathcal{L} 。我们可以通过计算 \mathcal{L} 关于参数的梯度，将其设置为0，并求解 θ 来找到全局最优值。利用第5章的结果，我们计算 \mathcal{L} 关于参数的梯度为：

$$\frac{d\mathcal{L}}{d\theta} = \frac{d}{d\theta} \left(\frac{1}{2\sigma^2} (y - X\theta)^\top (y - X\theta) \right) \quad (9.11a)$$

$$= \frac{1}{2\sigma^2} \frac{d}{d\theta} (y^\top y - 2y^\top X\theta + \theta^\top X^\top X\theta) \quad (9.11b)$$

$$= \frac{1}{\sigma^2} (-y^\top X + \theta^\top X^\top X) \in \mathbb{R}^{1 \times D} \quad (9.11c)$$

最大似然估计 θ_{ML} 解决了 $\frac{d\mathcal{L}}{d\theta} = 0^\top$ (必要最优化条件)，我们得到：

$$\frac{d\mathcal{L}}{d\theta} = 0^\top \quad (9.11c)$$

$$\Leftrightarrow \theta_{ML}^\top X^\top X = y^\top X \quad (9.12a)$$

$$\Leftrightarrow \theta_{ML}^\top = y^\top X (X^\top X)^{-1} \quad (9.12b)$$

$$\Leftrightarrow \theta_{ML} = (X^\top X)^{-1} X^\top y \quad (9.12c)$$

我们可以将 $X^\top X$ 的第一方程右侧乘以 $(X^\top X)^{-1}$ ，因为如果 $\text{rank}(X) = D$ ，则 $X^\top X$ 是正定的，其中 $\text{rank}(X)$ 表示 X 的秩。

注释。将梯度设为 $\mathbf{0}^\top$ 是一个必要且充分的条件，我们得到一个全局最小值，因为海森矩阵 $\nabla_\theta^2 \mathcal{L}(\theta) = X^\top X \in \mathbb{R}^{D \times D}$ 是正定的。

注释。最大似然解（式(9.12c)）要求我们解一个形式为 $A\theta = b$ 的线性方程组，其中 $A = (X^\top X)$ ， $b = X^\top y$ 。

示例 9.2 (拟合直线) 让我们看看图9.2，我们试图用最大似然估计拟合一条直线 $f(x) = \theta x$ ，其中 θ 是一个未知的斜率。图9.2(a)展示了这个模型类别的示例函数（直线）。对于图9.2(b)中的数据集，我们使用式(9.12c)找到斜率参数 θ 的最大似然估计，并在图9.2(c)中得到了最大似然线性函数。

最大似然估计与特征

到目前为止，我们考虑的线性回归设置如式(9.4)所示，允许我们用最大似然估计拟合直线。然而，直线在拟合更有趣的数据时表达能力不足。幸运的是，线性回归为我们提供了一种在不离开线性回归框架的情况下拟合非线性函数的方法：由于“线性回归”仅指“参数线性”，我们可以在输入 x 上执行任意非线性变换 $\phi(x)$ ，然后线性组合这个变换的各个分量。对应的线性回归模型为：

$$\begin{aligned} p(y|x, \theta) &= \mathcal{N}(y|\phi(x)^\top \theta, \sigma^2) \\ \iff y &= \phi(x)^\top \theta + \epsilon = \sum_{k=0}^{K-1} \theta_k \phi_k(x) + \epsilon \end{aligned} \quad (9.13)$$

其中， $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^K$ 是输入 x 的（非线性）变换， $\phi_k : \mathbb{R}^D \rightarrow \mathbb{R}$ 是特征向量 ϕ 的第 k 个分量。注意，特征向量的模型参数 θ 仍然仅线性出现。

示例 9.3 (多项式回归) 我们关注一个回归问题 $y = \phi(x)^\top \theta + \epsilon$ ，其中 $x \in \mathbb{R}$ ， $\theta \in \mathbb{R}^K$ 。在这种情况下常用的一个变换是：

$$\phi(x) = [\phi_0(x) \quad \phi_1(x) \quad \cdots \quad \phi_{K-1}(x)]^\top = [1 \quad x \quad x^2 \quad \cdots \quad (9.14)]^\top \in \mathbb{R}^K$$



这意味着我们将原始的一维输入空间“提升”到一个 K 维特征空间，包含所有单项式 x^k , $k = 0, \dots, K - 1$ 。通过这些特征，我们可以在线性回归框架内建模 $K - 1$ 阶多项式：一个 $K - 1$ 阶多项式为：

$$f(x) = \sum_{k=0}^{K-1} \theta_k x^k = \phi(x)^\top \theta \quad (9.15)$$

其中 ϕ 定义在式(9.14)中， $\theta = [\theta_0, \dots, \theta_{K-1}]^\top \in \mathbb{R}^K$ 包含线性参数 θ_k 。

现在，让我们看看在线性回归模型（式(9.13)）中，如何用最大似然估计来估计参数 θ 。我们考虑训练输入 $x_n \in \mathbb{R}^D$ 和目标值 $y_n \in \mathbb{R}$, $n = 1, \dots, N$ ，并定义特征矩阵（设计矩阵）为：

$$\Phi := \begin{bmatrix} \phi(x_1)^\top \\ \phi(x_2)^\top \\ \vdots \\ \phi(x_N)^\top \end{bmatrix} = \begin{bmatrix} \phi_0(x_1) & \cdots & \phi_{K-1}(x_1) \\ \phi_0(x_2) & \cdots & \phi_{K-1}(x_2) \\ \vdots & \ddots & \vdots \\ \phi_0(x_N) & \cdots & \phi_{K-1}(x_N) \end{bmatrix} \in \mathbb{R}^{N \times K} \quad (9.16)$$

其中 $\Phi_{ij} = \phi_j(x_i)$, $\phi_j : \mathbb{R}^D \rightarrow \mathbb{R}$ 。

示例 9.4（二阶多项式的特征矩阵）对于一个二阶多项式和 N 个训练点 $x_n \in \mathbb{R}$, $n = 1, \dots, N$ ，特征矩阵为：

$$\Phi = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 \end{bmatrix} \quad (9.17)$$

有了式(9.16)定义的特征矩阵 Φ ，线性回归模型（式(9.13)）的负对数似然可以写为：

\$\$

- $$\log p(Y | X, \theta) = \frac{1}{2\sigma^2} |y - \Phi \theta|^2 + \text{常数}$$



将式(9.18)与没有特征的模型的负对数似然 (式(9.10b)) 进行比较, 我们立即可以看出, 我们只需要将 X 替换为 Φ 。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:

$$\theta_{ML} = (\Phi^\top \Phi)^{-1} \Phi^\top y \quad (9.19)$$

对于具有非线性特征的线性回归问题。

注释。当我们不使用特征时, 我们要求 $X^\top X$ 可逆, 这在 $\text{rank}(X) = D$ 时成立, 即 X 的列 $\mathbb{R}^{K \times K}$ 可逆。这在 $\text{rank}(\Phi) = K$ 时成立。

示例 9.5 (最大似然多项式拟合)

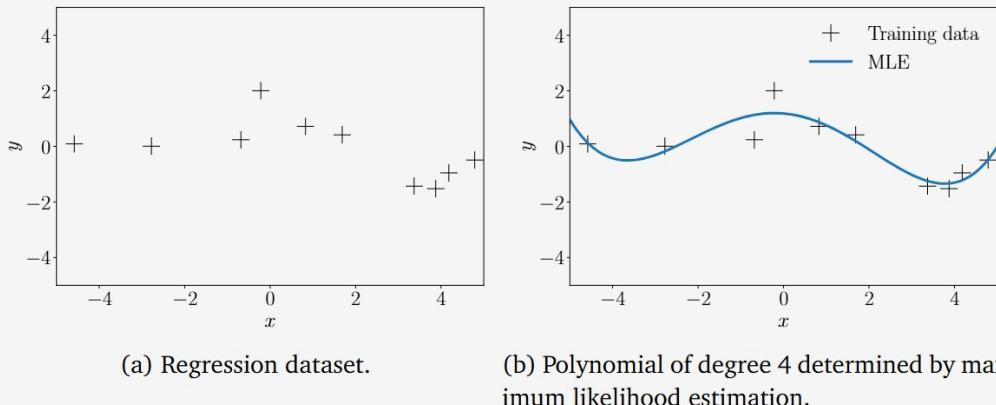


图 9.4: 多项式回归: (a) 包含 $((x_n, y_n))$ 对的数据集, 其中 ($n = 1, \dots, 10$); (b) 四阶最大似然多项式。考虑图9.4(a)中的数据集。该数据集包含 $N = 10$ 对 (x_n, y_n) , 其中 $x_n \sim \mathcal{U}[-5, 5]$, $y_n = -\sin(x_n/5) + \cos(x_n) + \epsilon$, 且 $\epsilon \sim \mathcal{N}(0, 0.2^2)$ 。我们使用最大似然估计拟合一个4阶多项式, 即参数 θ_{ML} 由式(9.19)给出。最大似然估计在任意测试位置 x^* 处得到的函数值为 $\phi(x^*)^\top \theta_{ML}$ 。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的原理来获得噪声方差的最大似然估计 σ_{ML}^2 。为此, 我们遵循标准程序: 写出对数



似然，对其关于 $\sigma^2 > 0$ 求导，将其设为0，并求解。对数似然为：

$$\begin{aligned} \text{\$\$ } & \begin{aligned} & \backslash\begin{aligned} & \log p(\mathcal{Y} \mid \mathcal{X}, \boldsymbol{\theta}, \sigma^2) \\ & \&= \sum_{n=1}^N \log \mathcal{N} \left(y_n \mid \boldsymbol{\phi}(x_n)^\top \boldsymbol{\theta}, \sigma^2 \right) \\ & \&= \sum_{n=1}^N \left(-\frac{1}{2} \log (2\pi) - \frac{1}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{n=1}^N \left(y_n - \boldsymbol{\phi}(x_n)^\top \boldsymbol{\theta} \right)^2 \right) \end{aligned} \\ & \text{\$\$} \end{aligned}$$

其中 $s := \sum_{n=1}^N (y_n - \boldsymbol{\phi}(x_n)^\top \boldsymbol{\theta})^2$ 。

对数似然关于 σ^2 的偏导数为：

$$\frac{\partial \log p(Y|X, \theta, \sigma^2)}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{s}{2\sigma^4} = 0 \quad (9.21a)$$

$$\Rightarrow \frac{N}{2\sigma^2} = \frac{s}{2\sigma^4} \quad (9.21b)$$

因此，我们得到：

$$\sigma_{\text{ML}}^2 = \frac{s}{N} = \frac{1}{N} \sum_{n=1}^N (y_n - \boldsymbol{\phi}(x_n)^\top \boldsymbol{\theta})^2 \quad (9.22)$$

因此，噪声方差的最大似然估计是噪声自由函数值 $\boldsymbol{\phi}(x_n)^\top \boldsymbol{\theta}$ 与对应噪声观测值 y_n 在输入位置 x_n 处的平方距离的均值。

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来拟合线性模型（例如多项式）到数据。我们可以通过计算误差/损失来评估模型的质量。一种方法是计算负对数似然（式 (9.10b)），我们最小化它以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 $\frac{1}{\sigma^2}$ 这一因子，因此我们得到一个平方误差损失函数 $\|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2$ 。与其使用这个平方损失，我们通常使用均方根误差（RMSE）：

$$\sqrt{\frac{1}{N} \|y - \Phi\theta\|^2} = \sqrt{\frac{1}{N} \sum_{n=1}^N (y_n - \phi(x_n)^\top \theta)^2} \quad (9.23)$$



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 具有相同的量纲和单位。例如, 如果我们拟合一个模型, 将邮编 (x 以纬度和经度表示) 映射到房价 (y 以欧元表示), 那么 RMSE 也以欧元为单位, 而平方误差则以欧元平方为单位。如果我们选择包括原始负对数似然 (式(9.10b)) 中的 σ^2 因子, 那么我们得到的是无量纲的目标, 即在前面的例子中, 我们的目标不再以欧元或欧元平方为单位。对于模型选择 (见第8.6节), 我们可以使用 RMSE (或负对数似然) 来确定最佳多项式的阶数, 通过找到使目标最小化的多项式阶数 M 。鉴于多项式的阶数是自然数, 我们可以进行穷举搜索, 并枚举所有 (合理的) M 值。对于大小为 N 的训练集, 测试 $0 \leq M \leq N - 1$ 是足够的。对于 $M < N$, 最大似然估计器是唯一的。对于 $M \geq N$, 我们有更多的参数比数据点多, 我们需要解一个欠定的线性方程组 (在式(9.19)中 $\Phi^\top \Phi$ 也将不可逆), 因此有无数个可能的最大似然估计器。

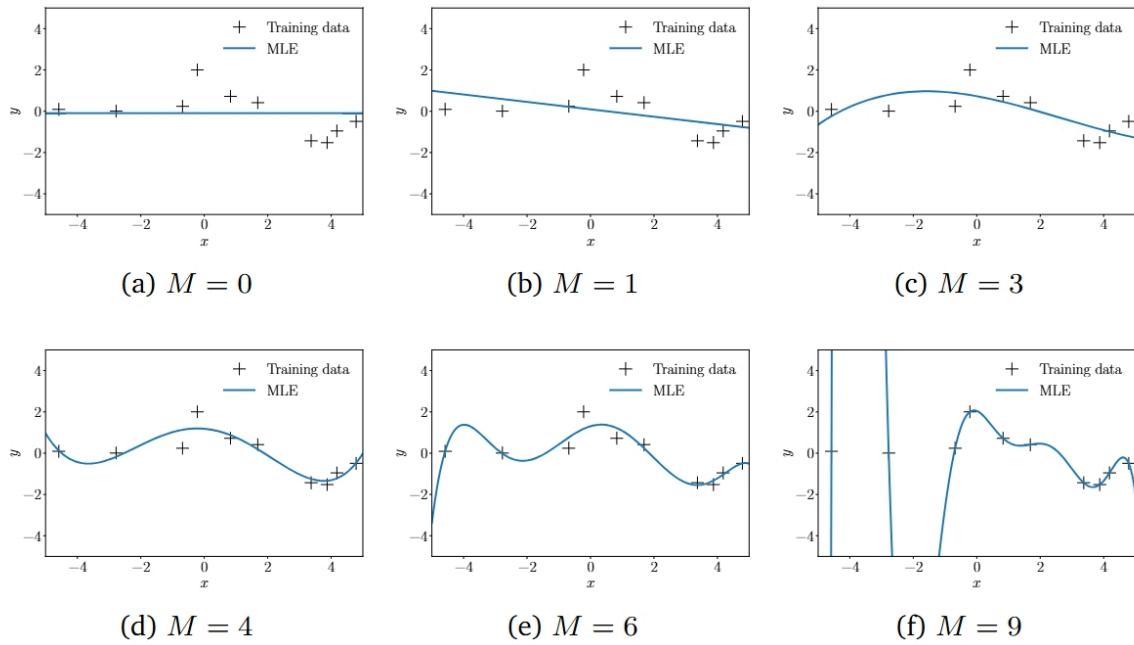


图 9.5: 不同多项式阶数 (M) 的最大似然拟合。

图9.5展示了使用最大似然为图9.4(a)中的数据集确定的不同阶数多项式的拟合结果, 该数据集包含 $N = 10$ 个观测值。我们注意到, 低阶多项式 (例如常数 $M = 0$ 或线性 $M = 1$) 拟合数据较差, 因此是对真实底层函数的糟糕表示。对于阶数为 $M = 3, \dots, 6$ 的多项式, 拟合结果看起来是合理的, 并且能够平滑地插值数据。当我们转向更高阶多项式时, 我们注意到它们对数据的拟合越来越好。在 $M = N - 1 = 9$ 的极端情况下, 函数将通过每一个数据点。然而, 这些高阶多项式会剧烈振荡, 并且是对生成数据的真实函数的糟糕表示, 因此我们遭受了过拟合。

注释。噪声方差 $\sigma^2 > 0$ 。



目标是通过在新（未见）数据上做出准确预测来实现良好的泛化。我们通过考虑一个单独的测试集来获得关于泛化性能对阶数为 M 的多项式依赖性的定量见解，该测试集包含 200 个数据点，这些数据点是使用生成训练集的确切相同程序生成的。作为测试输入，我们在 $[-5, 5]$ 区间内选择了一个线性网格的 200 个点。对于每个 M 的选择，我们使用均方根误差（RMSE，式(9.23)）来评估训练数据和测试数据的误差。

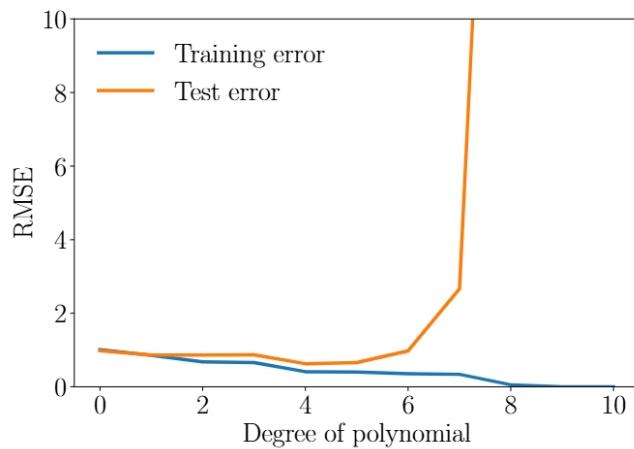


图 9.6: 训练误差与测试误差。

现在来看测试误差，它是对应多项式泛化性能的定性度量，我们注意到最初测试误差会降低；见图9.6（橙色）。对于四阶多项式，测试误差相对较低，并且在阶数为 5 之前保持相对稳定。然而，从阶数 6 开始，测试误差显著增加，高阶多项式的泛化性能非常差。在这个特定例子中，这也从图9.5 中对应的多项式拟合中显而易见。注意，训练误差（图9.6 中的蓝色训练误差曲线）在多项式的阶数增加时永远不会增加。在我们的例子中，最佳泛化（最小测试误差的点）是对于阶数为 $M = 4$ 的多项式获得的。

9.2.3 最大后验估计

我们刚刚看到，最大似然估计容易过拟合。我们通常观察到，当出现过拟合时，参数值的幅度会变得相对较大 (Bishop, 2006)。为了减轻参数值过大的影响，我们可以在参数上放置一个先验分布 $p(\theta)$ 。先验分布明确地编码了哪些参数值是合理的（在看到任何数据之前）。例如，一个高斯先验 $p(\theta) = \mathcal{N}(0, b^2 I)$ 对于单个参数 θ 表示参数值预期位于区间 $[-2b, 2b]$ 内（均值周围的两个标准差）。



一旦有了数据集 \mathbf{X} 和 \mathbf{Y} ，我们不是最大化似然，而是寻找最大化后验分布 $p(\theta|\mathbf{X}, \mathbf{Y})$ 的参数。这个过程称为**最大后验 (MAP) 估计**。后验分布 $p(\theta|\mathbf{X}, \mathbf{Y})$ 可以通过贝叶斯定理（第6.3节）得到：

$$p(\theta|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \theta)p(\theta)}{p(\mathbf{Y}|\mathbf{X})} \quad (9.24)$$

由于后验分布明确依赖于参数先验 $p(\theta)$ ，先验将对后验的最大值产生影响。我们将在后面更明确地看到这一点。参数向量 θ_{MAP} 是最大化后验（式(9.24)）的值，称为**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，并计算对数后验：

$$\log p(\theta|\mathbf{X}, \mathbf{Y}) = \log p(\mathbf{Y}|\mathbf{X}, \theta) + \log p(\theta) + \text{常数} \quad (9.25)$$

其中常数包括与 θ 无关的项。我们看到式(9.25)中的对数后验是似然 $p(\mathbf{Y}|\mathbf{X}, \theta)$ 的对数和先验 $\log p(\theta)$ 的和，因此**MAP**估计将是似然（数据依赖部分）和先验（我们对参数值的先验假设）之间的“折中”。

为了找到**MAP**估计 θ_{MAP} ，我们通过最小化负对数后验分布来求解：

$$\theta_{\text{MAP}} \in \arg \min_{\theta} \{-\log p(\mathbf{Y}|\mathbf{X}, \theta) - \log p(\theta)\} \quad (9.26)$$

负对数后验关于 θ 的梯度为：

$$-\frac{d \log p(\theta|\mathbf{X}, \mathbf{Y})}{d\theta} = -\frac{d \log p(\mathbf{Y}|\mathbf{X}, \theta)}{d\theta} - \frac{d \log p(\theta)}{d\theta} \quad (9.27)$$

其中，右侧的第一项是式(9.11c)中的负对数似然的梯度。对于线性回归设置（式(9.13)），假设参数 θ 的先验为高斯分布 $p(\theta) = \mathcal{N}(0, b^2 I)$ ，我们得到负对数后验：

$$-\log p(\theta|\mathbf{X}, \mathbf{Y}) = \frac{1}{2\sigma^2} \|y - \Phi\theta\|^2 + \frac{1}{2b^2} \theta^\top \theta + \text{常数} \quad (9.28)$$

这里，第一项来自似然的贡献，第二项来自先验。负对数后验关于参数 θ 的梯度为：

$$-\frac{d \log p(\theta|\mathbf{X}, \mathbf{Y})}{d\theta} = \frac{1}{\sigma^2} (\theta^\top \Phi^\top \Phi - y^\top \Phi) + \frac{1}{b^2} \theta^\top \theta \quad (9.29)$$

我们通过将这个梯度设为 $\mathbf{0}^\top$ 来找到**MAP**估计 θ_{MAP} ：

$$\frac{1}{\sigma^2} (\theta_{\text{MAP}}^\top \Phi^\top \Phi - y^\top \Phi) + \frac{1}{b^2} \theta_{\text{MAP}}^\top \theta = \mathbf{0}^\top \quad (9.30a)$$

$$\Leftrightarrow \theta_{\text{MAP}}^\top \left(\frac{1}{\sigma^2} \Phi^\top \Phi + \frac{1}{b^2} I \right) = \frac{1}{\sigma^2} y^\top \Phi \quad (9.30\text{b})$$

$$\Leftrightarrow \theta_{\text{MAP}}^\top = y^\top \Phi \left(\Phi^\top \Phi + \frac{\sigma^2}{b^2} I \right)^{-1} \quad (9.30\text{c})$$

因此，MAP估计为（通过转置上式两边）：

$$\theta_{\text{MAP}} = \left(\Phi^\top \Phi + \frac{\sigma^2}{b^2} I \right)^{-1} \Phi^\top y \quad (9.31)$$

将式(9.31)中的MAP估计与式(9.19)中的最大似然估计进行比较，我们看到两种解之间的唯一区别是逆矩阵中的额外项 $\frac{\sigma^2}{b^2} I$ 。这个项确保了 $\Phi^\top \Phi + \frac{\sigma^2}{b^2} I$ 是对称且严格正定的（即其逆存在，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，我们在参数 θ 上放置一个高斯先验 $p(\theta) = \mathcal{N}(0, I)$ ，并根据式(9.31)确定MAP估计。在图9.7中，我们展示了6阶和8阶多项式的最大似然估计和MAP估计。先验（正则化器）对于低阶多项式没有显著作用，但对于高阶多项式，它使函数保持相对平滑。尽管MAP估计可以限制过拟合的影响，但它并不是解决这个问题的通用方法，因此我们需要一个更合理的方法来处理过拟合。

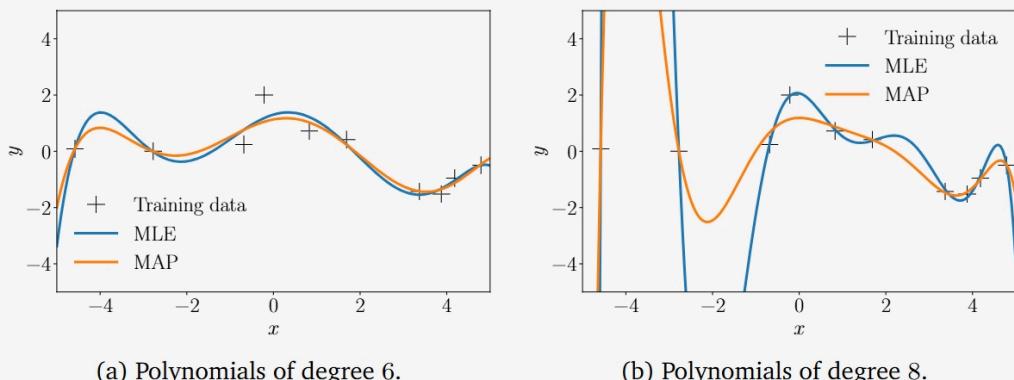


图9.7 多项式回归：最大似然和MAP估计。(a) 6阶多项式；(b) 8阶多项式。

9.2.4 MAP估计作为正则化

除了在参数 θ 上放置先验分布外，我们还可以通过正则化来减轻轻过拟合的影响。在正则化最小二乘法中，我们考虑损失函数：

$$\|y - \Phi\theta\|^2 + \lambda\|\theta\|_2^2 \quad (9.32)$$

其中，第一项是数据拟合项（也称为拟合误差项），与负对数似然成正比（见式(9.10b)）。第二项称为正则化项，正则化参数 $\lambda \geq 0$ 控制正则化的严格程度。

注释。在式(9.32)中，除了 Euclid 范数 $\|\cdot\|_2$ 外，我们还可以选择任意 p -范数 $\|\cdot\|_p$ 。在实践中，较小的 p 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 $p = 1$ ，正则化项称为**LASSO**（最小绝对收缩和选择算子），由Tibshirani (1996) 提出。

正则化项 $\lambda\|\theta\|_2^2$ 在式(9.32)中可以被解释为负高斯先验的对数。更具体地说，对于高斯先验 $p(\theta) = \mathcal{N}(0, b^2 I)$ ，我们得到负高斯先验的对数：

$$-\log p(\theta) = \frac{1}{2b^2} \|\theta\|_2^2 + \text{常数} \quad (9.33)$$

因此，当 $\lambda = \frac{1}{2b^2}$ 时，正则化项和负高斯先验的对数是相同的。鉴于正则化最小二乘法的损失函数（式(9.32)）由与负对数似然和负先验相关的项组成，当我们最小化这个损失时，得到的解与式(9.31)中的**MAP**估计非常相似。更具体地说，最小化正则化最小二乘法的损失函数得到：

$$\theta_{\text{RLS}} = (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top y \quad (9.34)$$

这与式(9.31)中的**MAP**估计完全相同，其中 $\lambda = \frac{\sigma^2}{b^2}$ ， σ^2 是噪声方差， b^2 是各向同性高斯先验 $p(\theta) = \mathcal{N}(0, b^2 I)$ 的方差。一个点估计是一个特定的参数值，与参数设置的分布不同。

到目前为止，我们已经讨论了使用最大似然和**MAP**估计进行参数估计，我们找到了优化目标函数（似然或后验）的点估计 θ^* 。我们发现最大似然和**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，我们将使用贝叶斯推断（第8.4节）来找到参数的后验分布，然后用它来进行预测。更具体地说，对于预测，我们将对所有合理的参数设置进行平均，而不是专注于一个点估计。

9.1 问题形式化

9.3 贝叶斯线性回归





9.3 贝叶斯线性回归

在前面的内容中，我们讨论了线性回归模型，其中我们通过最大似然估计或MAP估计来估计模型参数 θ 。我们发现，最大似然估计可能会导致严重的过拟合，尤其是在小数据情况下。MAP估计通过在参数上放置先验分布来缓解这一问题，起到了正则化的作用。贝叶斯线性回归（Bayesian Linear Regression）将参数先验的概念进一步推进，不再尝试计算参数的点估计，而是考虑参数的完整后验分布，并在进行预测时将其纳入考虑。这意味着我们不拟合任何参数，而是对所有合理的参数设置（根据后验分布）进行平均。

9.3.1 模型

在贝叶斯线性回归中，我们考虑以下模型：

$$\begin{aligned} \text{先验分布: } p(\theta) &= \mathcal{N}(\theta|m_0, S_0) \\ \text{似然函数: } p(y|x, \theta) &= \mathcal{N}(y|\phi(x)^\top \theta, \sigma^2) \end{aligned} \tag{9.35}$$

其中，我们明确地在 θ 上放置了一个高斯先验 $p(\theta) = \mathcal{N}(\theta|m_0, S_0)$ ，这使得参数向量成为一个随机变量。这允许我们写出对应的概率图模型，如图9.8所示，其中我们将高斯先验的参数明确表示出来。

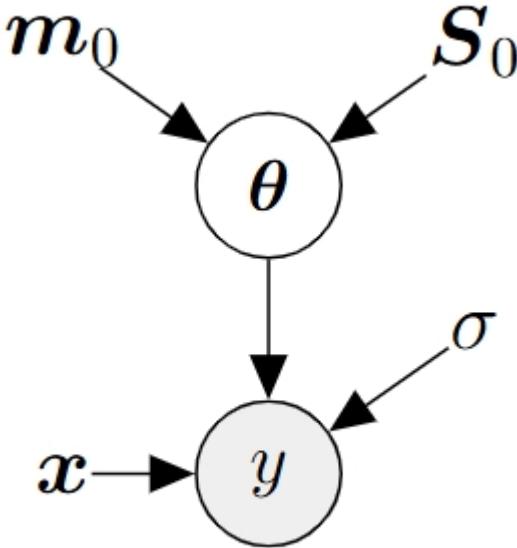


图9.8: 贝叶斯线性回归的概率图模型。

完整的概率模型，即观测和未观测随机变量 y 和 θ 的联合分布为：

$$p(y, \theta|x) = p(y|x, \theta)p(\theta) \quad (9.36)$$

9.3.2 先验预测

在实践中，我们通常对参数值 θ 本身并不感兴趣。相反，我们的关注点往往在于使用这些参数值进行的预测。在贝叶斯设置中，我们取参数分布，并在进行预测时对所有合理的参数设置进行平均。具体来说，为了在输入 x^* 处进行预测，我们对 θ 进行积分，得到：

$$p(y^*|x^*) = \int p(y^*|x^*, \theta)p(\theta)d\theta = \mathbb{E}_\theta[p(y^*|x^*, \theta)] \quad (9.37)$$

这可以被解释为对所有合理的参数 θ （根据先验分布 $p(\theta)$ ）的平均预测 $y^*|x^*, \theta$ 。注意，使用先验分布进行预测仅需要我们指定输入 x^* ，而不需要训练数据。在我们的模型（式(9.35)）中，我们选择了一个共轭（高斯）先验 $p(\theta) = \mathcal{N}(\theta|m_0, S_0)$ ，因此预测分布也是高斯的（并且可以以闭形式计算）：

$$p(y^*|x^*) = \mathcal{N}(y^*|\phi(x^*)^\top m_0, \phi(x^*)^\top S_0 \phi(x^*) + \sigma^2) \quad (9.38)$$

其中，我们利用了以下事实： (i) 由于共轭性（见第6.6节）和高斯分布的边缘化性质（见第6.5节），预测是高斯的； (ii) 高斯噪声是独立的，因此

$$\text{Var}[y^*] = \text{Var}_\theta[\phi(x^*)^\top \theta] + \text{Var}_\epsilon[\epsilon] \quad (9.3)$$

(iii) y^* 是 θ 的线性变换，因此我们可以利用 (6.50) 和 (6.51) 的规则，分别计算预测的均值和协方差。在式(9.38)中，预测方差中的 $\phi(x^*)^\top S_0 \phi(x^*)$ 项明确地考虑了与参数 θ 相关的不确定性，而 σ^2 是由于测量噪声导致的不确定性贡献。如果我们感兴趣的是预测无噪声函数值 $f(x^*) = \phi(x^*)^\top \theta$ ，而不是噪声干扰的目标 y^* ，我们得到：

$$p(f(x^*)) = \mathcal{N}(f(x^*) | \phi(x^*)^\top m_0, \phi(x^*)^\top S_0 \phi(x^*)) \quad (9.40)$$

这与式(9.38)的区别仅在于省略了噪声方差 σ^2 。

注释（函数分布）。由于我们可以使用一组样本 θ_i 来表示分布 $p(\theta)$ ，并且每个样本 θ_i 都产生一个函数 $f_i(\cdot) = \theta_i^\top \phi(\cdot)$ ，因此参数分布 $p(\theta)$ 诱导了一个函数分布 $p(f(\cdot))$ 。这里我们使用 (\cdot) 明确表示函数关系。

示例 9.7（先验函数分布） 考虑一个贝叶斯线性回归问题，其中多项式的阶数为5。我们选择参数先验 $p(\theta) = \mathcal{N}(\theta | 0, \frac{1}{4}I)$ 。图9.9展示了由该参数先验诱导的先验函数分布（阴影区域：深灰色表示67%置信区间；浅灰色表示95%置信区间），以及从该先验中采样得到的一些函数样本。函数样本是通过首先从参数先验 $p(\theta)$ 中采样一个参数向量 θ_i ，然后计算 $f_i(\cdot) = \theta_i^\top \phi(\cdot)$ 得到的。我们使用了200个输入位置 $x^* \in [-5, 5]$ ，并将特征函数 $\phi(\cdot)$ 应用于这些位置。图9.9中的不确定性（由阴影区域表示）仅由于参数不确定性引起，因为我们考虑的是无噪声的预测分布（式(9.40)）。

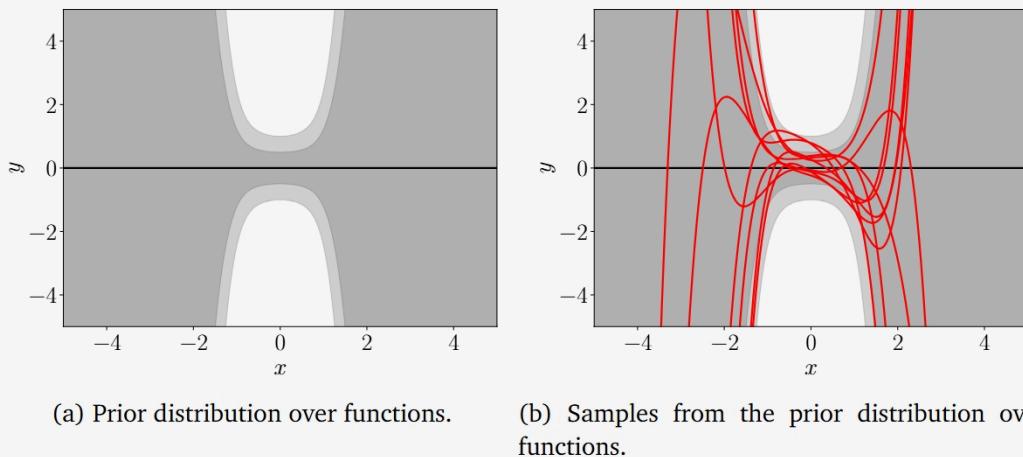


图9.9: 先验函数分布。(a) 由均值函数（黑线）和边缘不确定性（阴影区域）表示的分布，分别表示67%和95%置信区间；(b) 从先验函数分布中采样的样本，这些样本由参数先验的样本诱导。到目前为止，我们讨论了如何使用参

数先验 $p(\theta)$ 进行预测。然而，当我们有一个参数后验（给定一些训练数据 \mathbf{X}, \mathbf{Y} ）时，预测和推理的原则与式(9.37)相同——我们只需要将先验 $p(\theta)$ 替换为后验 $p(\theta|\mathbf{X}, \mathbf{Y})$ 。在接下来的内容中，我们将详细推导后验分布，然后使用它进行预测。

9.3.3 后验分布

给定一组训练输入 $\mathbf{x}_n \in \mathbb{R}^D$ 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：

$$p(\theta|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \theta)p(\theta)}{p(\mathbf{Y}|\mathbf{X})} \quad (9.41)$$

其中， \mathbf{X} 是训练输入的集合， \mathbf{Y} 是对应的训练目标值的集合。此外， $p(\mathbf{Y}|\mathbf{X}, \theta)$ 是似然函数， $p(\theta)$ 是参数先验，而

$$p(\mathbf{Y}|\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{X}, \theta)p(\theta)d\theta = \mathbb{E}_{\theta}[p(\mathbf{Y}|\mathbf{X}, \theta)] \quad (9.42)$$

是边际似然（证据），它与参数 θ 无关，并确保后验分布是归一化的，即它积分等于1。我们可以将边际似然视为在先验分布 $p(\theta)$ 下的期望似然。

定理 9.1 (参数后验) 在我们的模型（式(9.35)）中，参数后验（式(9.41)）可以以闭形式计算为：

$$p(\theta|\mathbf{X}, \mathbf{Y}) = \mathcal{N}(\theta|m_N, S_N) \quad (9.43a)$$

其中

$$S_N = (S_0^{-1} + \sigma^{-2}\Phi^\top\Phi)^{-1} \quad (9.43b)$$

$$m_N = S_N(S_0^{-1}m_0 + \sigma^{-2}\Phi^\top y) \quad (9.43c)$$

这里，下标 N 表示训练集的大小。

证明 贝叶斯定理告诉我们，后验 $p(\theta|\mathbf{X}, \mathbf{Y})$ 与似然 $p(\mathbf{Y}|\mathbf{X}, \theta)$ 和先验 $p(\theta)$ 的乘积成比例：

$$\text{后验 } p(\theta|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \theta)p(\theta)}{p(\mathbf{Y}|\mathbf{X})} \quad (9.44a)$$

$$\text{似然 } p(Y|X, \theta) = \mathcal{N}(y|\Phi\theta, \sigma^2 I) \quad (9.44a)$$

$$\text{先验 } p(\theta) = \mathcal{N}(\theta|m_0, S_0) \quad (9.44c)$$

与其考虑似然和先验的乘积，我们可以在对数空间中进行变换，并通过对数似然和对数先验的和来求解后验的均值和协方差。对数似然和对数先验的和为：

$$\log \mathcal{N}(y|\Phi\theta, \sigma^2 I) + \log \mathcal{N}(\theta|m_0, S_0) \quad (9.45a)$$

$$= -\frac{1}{2} (\sigma^{-2}(y - \Phi\theta)^\top(y - \Phi\theta) + (\theta - m_0)^\top S_0^{-1}(\theta - m_0)) + \text{常数} \quad (9.45b)$$

其中常数包含与 θ 无关的项。在接下来的推导中，我们忽略这些常数项。现在我们对式(9.45b)进行因式分解，得到：

$$\begin{aligned} & -\frac{1}{2} (\sigma^{-2}y^\top y - 2\sigma^{-2}y^\top \Phi\theta + \theta^\top \sigma^{-2}\Phi^\top \Phi\theta + \theta^\top S_0^{-1}\theta - 2m_0^\top S_0^{-1}\theta) \\ &= -\frac{1}{2} (\theta^\top (\sigma^{-2}\Phi^\top \Phi + S_0^{-1})\theta - 2(\sigma^{-2}\Phi^\top y + S_0^{-1}m_0)^\top \theta) + \text{常数} \end{aligned} \quad (9.46b)$$

其中，橙色部分是与 θ 线性相关的项，蓝色部分是与 θ 二次相关的项。观察式(9.46b)，我们发现这个表达式是关于 θ 的二次形式。未归一化的对数后验分布是负的二次形式这一事实表明，后验分布是高斯的，即：

$$p(\theta|X, Y) = \exp(\log p(\theta|X, Y)) \propto \exp(\log p(Y|X, \theta) + \log p(\theta)) \quad (9.47a)$$

$$\propto \exp\left(-\frac{1}{2} (\theta^\top (\sigma^{-2}\Phi^\top \Phi + S_0^{-1})\theta - 2(\sigma^{-2}\Phi^\top y + S_0^{-1}m_0)^\top \theta)\right) \quad (9.47b)$$

接下来的任务是将这个未归一化的高斯分布转换为与 $\mathcal{N}(\theta|m_N, S_N)$ 成比例的形式，即我们需要确定均值 m_N 和协方差矩阵 S_N 。为此，我们使用“补全平方”的方法。期望的对数后验形式为：

$$\log \mathcal{N}(\theta|m_N, S_N) = -\frac{1}{2}(\theta - m_N)^\top S_N^{-1}(\theta - m_N) + \text{常数} \quad (9.48a)$$

$$= -\frac{1}{2} (\theta^\top S_N^{-1}\theta - 2m_N^\top S_N^{-1}\theta + m_N^\top S_N^{-1}m_N) \quad (9.48b)$$

这里，我们将二次项 $(\theta - m_N)^\top S_N^{-1}(\theta - m_N)$ 分解为仅与 θ 二次相关的项（蓝色）、与 θ 线性相关的项（橙色）和常数项（黑色）。这使我们能够通过匹配式(9.46b)和式(9.48b)中的彩色表达式来确定 S_N 和 m_N ，从而得到：

$$S_N^{-1} = \sigma^{-2}\Phi^\top \Phi + S_0^{-1} \quad (9.49a)$$

$$\iff S_N = (\sigma^{-2} \Phi^\top \Phi + S_0^{-1})^{-1}$$

(9.49b)

以及

$$m_N^\top S_N^{-1} = (\sigma^{-2} \Phi^\top y + S_0^{-1} m_0)^\top \quad (9.50a)$$

$$\iff m_N = S_N(\sigma^{-2} \Phi^\top y + S_0^{-1} m_0) \quad (9.50b)$$

注释 (补全平方的一般方法) 如果我们有一个方程

$$x^\top A x - 2a^\top x + \text{常数}_1 \quad (9.51)$$

其中 A 是对称且正定的矩阵, 我们希望将其转换为以下形式:

$$(x - \mu)^\top \Sigma(x - \mu) + \text{常数}_2 \quad (9.52)$$

我们可以通过以下方式实现:

$$\Sigma := A \quad (9.53)$$

$$\mu := \Sigma^{-1} a \quad (9.54)$$

以及

$$\text{常数}_2 = \text{常数}_1 - \mu^\top \Sigma \mu$$

注释 由于 $p(\theta|X, Y) = \mathcal{N}(\theta|m_N, S_N)$, 因此 $\theta_{\text{MAP}} = m_N$ 。在我们的模型中, Φ 和 S_0 是已知的, 因此 S_N 和 m_N 可以直接计算得到。这表明, 贝叶斯线性回归的后验均值 m_N 与最大后验估计 θ_{MAP} 是一致的。然而, 贝叶斯方法不仅提供了一个点估计, 还提供了一个完整的后验分布, 这使得我们可以对参数的不确定性进行量化。这种不确定性在预测新数据时尤其重要, 因为它允许我们评估模型预测的置信度。接下来, 我们将利用这个后验分布来进行预测。

9.3.4 后验预测

在式(9.37)中, 我们使用参数先验 $p(\theta)$ 在测试输入 x^* 处计算预测分布 $p(y^*|x^*)$ 。从原理上讲, 使用参数后验 $p(\theta|X, Y)$ 进行预测与使用先验并无本质区别, 因为在我们的共轭模型中, 先验和后验都是高斯分布 (只是参数不同)。因此, 按照与第 9.3.2 节相同的推理, 我们得到 (后验) 预测分布:

$$p(y^*|X, Y, x^*) = \int p(y^*|x^*, \theta)p(\theta|X, Y)d\theta \quad (9.57a)$$

$$= \int \mathcal{N}(y^* | \phi(x^*)^\top \theta, \sigma^2) \mathcal{N}(\theta | m_N, S_N) d\theta \quad (9.57b)$$

$$= \mathcal{N}(y^* | \phi(x^*)^\top m_N, \phi(x^*)^\top S_N \phi(x^*) + \sigma^2) \quad (9.57c)$$

其中， $\phi(x^*)^\top S_N \phi(x^*)$ 反映了与参数 θ 相关的后验不确定性。注意， S_N 依赖于训练输入，通过 Φ 体现；见式(9.43b)。预测均值 $\phi(x^*)^\top m_N$ 与使用MAP估计 θ_{MAP} 进行预测的结果一致。

注释（边际似然与后验预测分布） 通过替换式(9.57a)中的积分，预测分布也可以等价地表示为 $\mathbb{E}_{\theta|X,Y}[p(y^*|x^*, \theta)]$ ，其中期望是相对于参数后验 $p(\theta|X, Y)$ 取的。以这种方式写出后验预测分布，突显了它与边际似然（式(9.42)）之间的密切相似性。边际似然与后验预测分布之间的关键区别在于：(i) 边际似然可以被认为是在先验下对训练目标 y 的预测，而不是对测试目标 y^* 的预测；(ii) 边际似然是对参数先验的平均，而不是对参数后验的平均。

注释（无噪声函数值的均值和方差） 在许多情况下，我们感兴趣的不是带噪声观测 y^* 的预测分布 $p(y^*|X, Y, x^*)$ ，而是无噪声函数值 $f(x^*) = \phi(x^*)^\top \theta$ 的分布。我们可以通过利用均值和方差的性质来确定相应的矩：

$$\mathbb{E}[f(x^*)|X, Y] = \mathbb{E}_\theta[\phi(x^*)^\top \theta|X, Y] = \phi(x^*)^\top \mathbb{E}_\theta[\theta|X, Y] = \phi(x^*)^\top m_N \quad (9.58)$$

$$\text{Var}_\theta[f(x^*)|X, Y] = \text{Var}_\theta[\phi(x^*)^\top \theta|X, Y] = \phi(x^*)^\top \text{Var}_\theta[\theta|X, Y] \phi(x^*) \quad (9.59)$$

我们看到，预测均值与带噪声观测的预测均值相同，因为噪声的均值为0。预测方差仅在包含 σ^2 时有所不同，这是测量噪声的方差。当我们预测带噪声的函数值时，需要将 σ^2 作为不确定性的一个来源包括在内，但对于无噪声预测，这个项是不需要的。这里，唯一的剩余不确定性来自于参数后验。

注释（函数分布） 由于我们对参数 θ 进行了积分，这诱导了一个函数分布：如果我们从参数后验 $p(\theta|X, Y)$ 中采样 θ_i ，我们得到一个函数实现 $\theta_i^\top \phi(\cdot)$ 。这个函数分布的均值函数，即所有期望函数值 $\mathbb{E}_\theta[f(\cdot)|\theta, X, Y]$ 的集合，是 $m_N^\top \phi(\cdot)$ 。函数 $f(\cdot)$ 的（边缘）方差由 $\phi(\cdot)^\top S_N \phi(\cdot)$ 给出。

示例 9.8（后验函数分布）

让我们重新审视多项式阶数为5的贝叶斯线性回归问题。我们选择参数先验 $p(\theta) = \mathcal{N}(0, \frac{1}{4}I)$ 。图9.9展示了由参数先验诱导的先验函数分布，并从该先验中采样得到的函数样本。



图9.10展示了通过贝叶斯线性回归得到的后验函数分布。训练数据集如图(a)所示；图(b)展示了后验函数分布，包括通过最大似然和MAP估计得到的函数。MAP估计也对应于贝叶斯线性回归设置中的后验均值函数。图(c)展示了从后验函数分布中采样的函数样本。

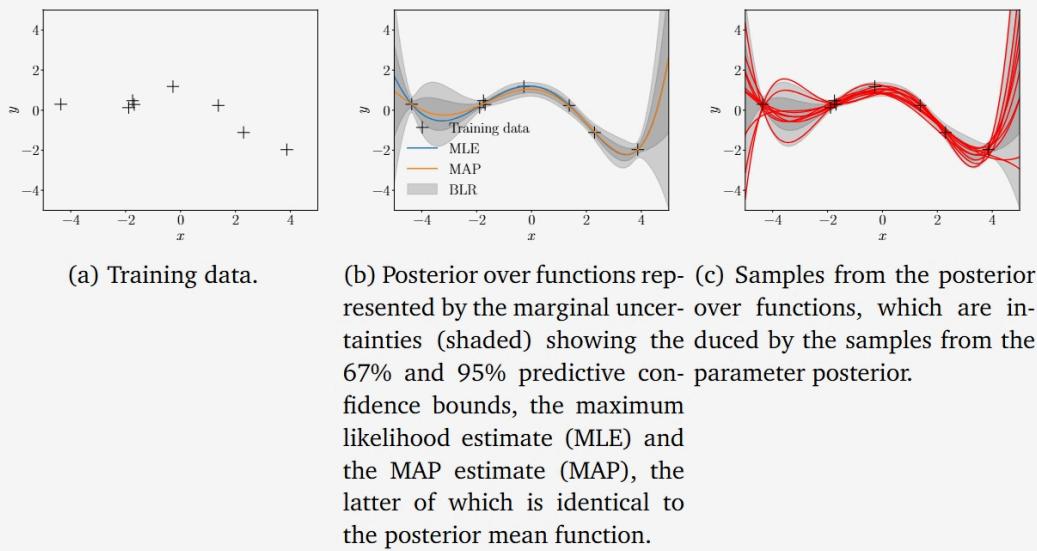
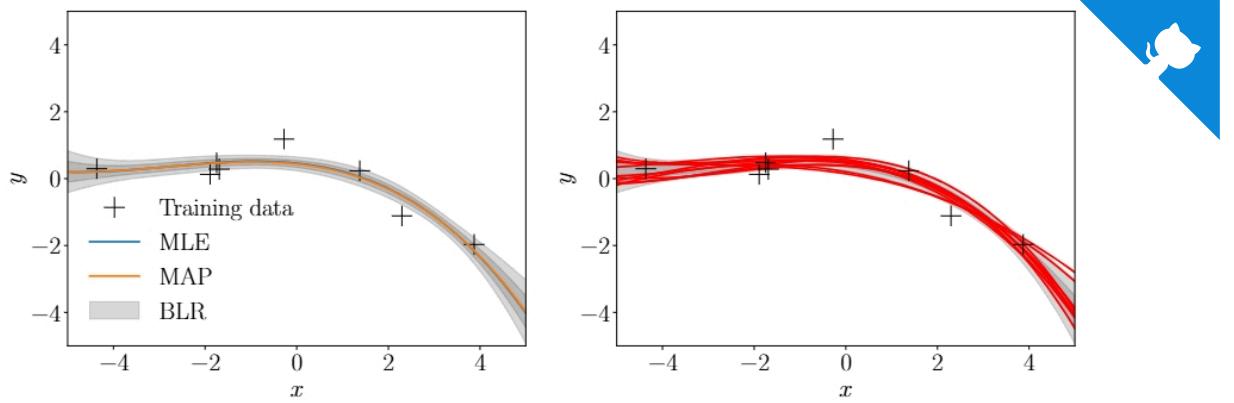
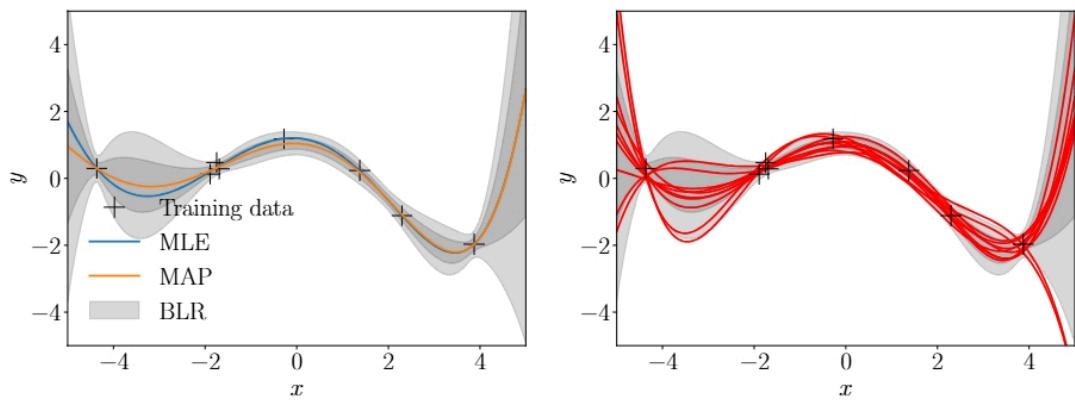


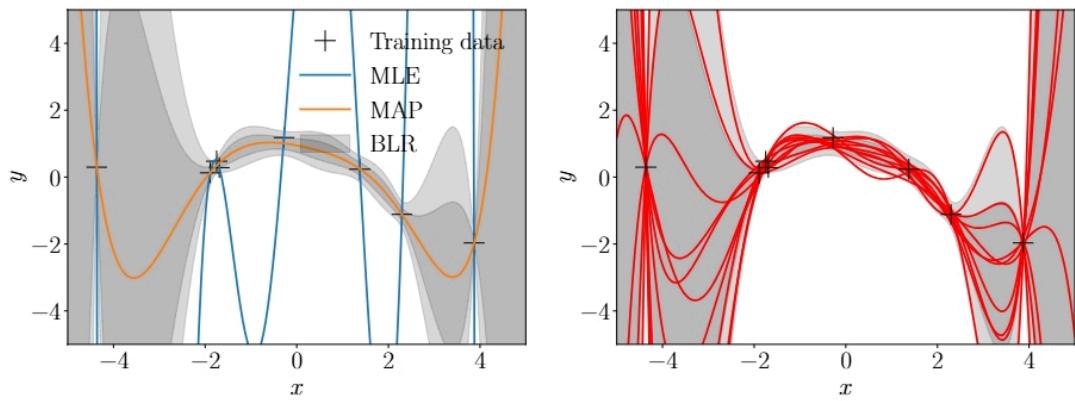
图9.10 贝叶斯线性回归和后验函数分布。(a) 训练数据；(b) 后验函数分布，包括通过最大似然和MAP估计得到的函数。MAP估计也对应于贝叶斯线性回归设置中的后验均值函数。(c) 从后验函数分布中采样的函数样本。



(a) Posterior distribution for polynomials of degree $M = 3$ (left) and samples from the posterior over functions (right).



(b) Posterior distribution for polynomials of degree $M = 5$ (left) and samples from the posterior over functions (right).



(c) Posterior distribution for polynomials of degree $M = 7$ (left) and samples from the posterior over functions (right).

图9.11 不同多项式阶数的贝叶斯线性回归。左侧图：阴影区域表示67%（深灰色）和95%（浅灰色）预测置信区间。贝叶斯线性回归模型的均值与MAP估计一致。右侧图：从后验函数分布中采样的函数样本。

图9.11展示了不同多项式阶数 M 下的后验函数分布。左侧图展示了最大似然函数 $\theta_{\text{ML}}^\top \phi(\cdot)$ 、MAP函数 $\theta_{\text{MAP}}^\top \phi(\cdot)$ （与后验均值函数相同），以及由贝叶斯线性回归

得到的67%和95%预测置信区间，用阴影区域表示。右侧图展示了从后验函数分布中采样的函数样本：我们从参数后验中采样参数 θ_i ，并计算函数 $\phi(x^*)^\top \theta_i$ ，这是在后验分布下函数的一个实现。对于低阶多项式，参数后验对参数的约束较弱：采样得到的函数几乎相同。当我们通过增加参数数量使模型更灵活（即得到更高阶的多项式）时，这些参数没有被后验充分约束，采样得到的函数可以很容易地区分开来。我们也可以从对应的左侧图中看到，不确定性在边界处尤其增大。尽管对于7阶多项式，MAP估计得到了一个合理的拟合，但贝叶斯线性回归模型还告诉我们，后验不确定性是巨大的。当我们将这些预测用于决策系统时，这种信息可能至关重要，因为错误的决策可能会产生严重的后果（例如在强化学习或机器人学中）。

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在贝叶斯模型选择中的重要性。接下来，我们将计算贝叶斯线性回归的边际似然，其中参数具有共轭高斯先验，这正是我们在本章中讨论的设置。回顾一下，我们考虑以下生成过程：

$$\theta \sim \mathcal{N}(m_0, S_0) \quad (9.60a)$$

$$y_n | x_n, \theta \sim \mathcal{N}(x_n^\top \theta, \sigma^2), \quad n = 1, \dots, N \quad (9.60b)$$

边际似然是

$$p(Y|X) = \int p(Y|X, \theta)p(\theta)d\theta \quad (9.61a)$$

$$= \int \mathcal{N}(y|X\theta, \sigma^2 I)\mathcal{N}(\theta|m_0, S_0)d\theta \quad (9.61b)$$

其中，我们对模型参数 θ 进行了积分。我们分两步计算边际似然：首先，我们证明边际似然是高斯分布（作为 y 的分布）；其次，我们计算这个高斯分布的均值和协方差矩阵。

- 1. 边际似然是高斯分布：**从第6.5.2节中，我们知道 (i) 两个高斯随机变量的乘积是一个（未归一化的）高斯分布，以及 (ii) 高斯随机变量的线性变换也是高斯分布的。在式(9.61b)中，我们需要一个线性变换，将 $\mathcal{N}(y|X\theta, \sigma^2 I)$ 转换为 $\mathcal{N}(\theta|\mu, \Sigma)$ 的形式，其中 μ 和 Σ 是某些参数。一旦完成这一转换，积分就可以用闭形式求解。结果是两个高斯分布乘积的归一化常数。归一化常数本身具有高斯形状；见式(6.76)。

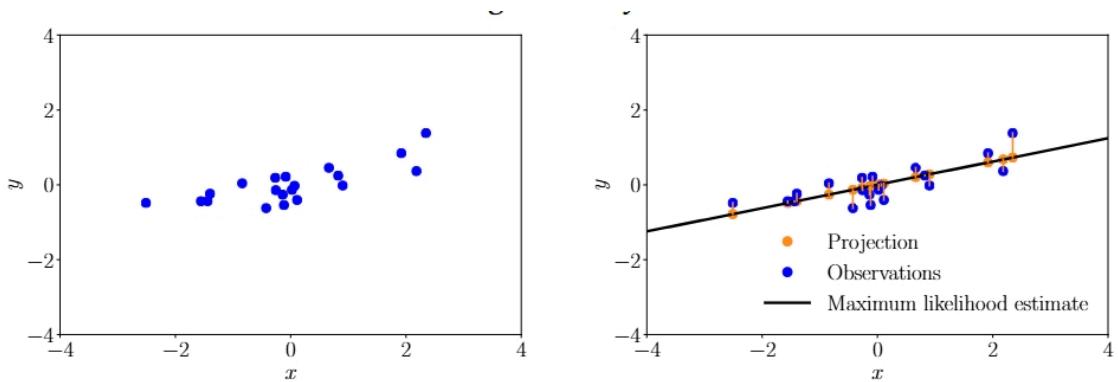
2. 均值和协方差矩阵：我们通过利用第6.4.4节中关于随机变量的仿射变换的均值和协方差的标准结果来计算边际似然的均值和协方差矩阵。边际似然的均值计算如下：

$$\mathbb{E}[Y|X] = \mathbb{E}_{\theta,\epsilon}[X\theta + \epsilon] = X\mathbb{E}_\theta[\theta] = Xm_0 \quad (9.62)$$

注意， $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ 是一个独立同分布的随机变量向量。协方差矩阵为：

$$\text{Cov}[Y|X] = \text{Cov}_{\theta,\epsilon}[X\theta + \epsilon] = \text{Cov}_\theta[X\theta] + \sigma^2 I \quad (9.63a)$$

$$= X\text{Cov}_\theta[\theta]X^\top + \sigma^2 I = XS_0X^\top + \sigma^2 I \quad (9.63b)$$



(a) Regression dataset consisting of noisy observations y_n (blue) of function values $f(x_n)$ at input locations x_n .

(b) The orange dots are the projections of the noisy observations (blue dots) onto the line $\theta_{ML}x$. The maximum likelihood solution to a linear regression problem finds a subspace (line) onto which the overall projection error (orange lines) of the observations is minimized.

图 9.12: 最小二乘的几何解释。(a) 数据集; (b) 最大似然解的投影解释。

因此，边际似然是

$$p(Y|X) = (2\pi)^{-\frac{N}{2}} \det(XS_0X^\top + \sigma^2 I)^{-\frac{1}{2}} \quad (9.64a)$$

$$\begin{aligned} &\cdot \exp\left(-\frac{1}{2}(y - Xm_0)^\top (XS_0X^\top + \sigma^2 I)^{-1}(y - Xm_0)\right) \\ &= \mathcal{N}(y|Xm_0, XS_0X^\top + \sigma^2 I) \end{aligned} \quad (9.64b)$$

鉴于与后验预测分布（见前面关于边际似然和后验预测分布的注释）的密切联系，边际似然的形式并不令人意外。



< 上一章节

下一章节 >

9.2 参数估计

9.4 最大似然作为正交投影



9.4 最大似然作为正交投影

在经过大量代数运算推导出最大似然估计和MAP估计之后，我们现在将为最大似然估计提供一个几何解释。让我们考虑一个简单的线性回归设置：

$$y = x\theta + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \quad (9.65)$$

其中，我们考虑从原点通过的线性函数 $f : \mathbb{R} \rightarrow \mathbb{R}$ （为了清晰起见，这里省略了特征）。参数 θ 决定了直线的斜率。图9.12(a)展示了一个一维数据集。给定一个训练数据集 $\{(x_1, y_1), \dots, (x_N, y_N)\}$ ，回忆第9.2.1节中的结果，我们得到斜率参数的最大似然估计为：

$$\theta_{\text{ML}} = (X^\top X)^{-1} X^\top y = \frac{X^\top y}{X^\top X} \in \mathbb{R} \quad (9.66)$$

其中， $X = [x_1, \dots, x_N]^\top \in \mathbb{R}^N$ ， $y = [y_1, \dots, y_N]^\top \in \mathbb{R}^N$ 。这意味着对于训练输入 X ，我们得到训练目标的最佳（最大似然）重构为：

$$X\theta_{\text{ML}} = X \left(\frac{X^\top y}{X^\top X} \right) = \frac{XX^\top}{X^\top X} y \quad (9.67)$$

即，我们得到了 y 和 $X\theta$ 之间最小二乘误差的近似值。由于我们正在寻找 $y = X\theta$ 的解，因此我们可以将线性回归视为求解线性方程组的问题。因此，我们可以联系到我们在第2章和第3章中讨论的线性代数和解析几何的概念。仔细观察式(9.67)，我们发现最大似然估计 θ_{ML} 在我们的例子（式(9.65)）中实际上是对 y 进行正交投影，将其投影到由 X 张成的一维子空间上。回忆第3.8节中关于正交投影的结果，我们识别出 $\frac{XX^\top}{X^\top X}$ 是投影矩阵， θ_{ML} 是投影到 \mathbb{R}^N 中由 X 张成的一维子空间上的坐标，而 $X\theta_{\text{ML}}$ 是 y 到这个子空间的正交投影。因此，最大似然解还通过正交投影提供了一个几何最优解，通过找到子空间中“最接近”对应观测值 y 的向量，其中“最接近”意味着函数值 y_n 与 $x_n\theta$ 之间的最小（平方）距离。这是通过正交投影实现的。图9.12(b)展示了将噪声观测值投影到最小化原始数据集及其投影（注意 x 坐标是固定的）之间的平方距离的子空间上，这对应于最大似然解。



图9.12 图9.12 最小二乘法的几何解释。(a) 数据集; (b) 最大似然解被解释为一个投影。



在一般线性回归情况下，其中

$$y = \phi(x)^\top \theta + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \quad (9.68)$$

具有向量值特征 $\phi(x) \in \mathbb{R}^K$ ，我们同样可以将最大似然结果

$$y \approx \Phi \theta_{\text{ML}}, \quad \theta_{\text{ML}} = (\Phi^\top \Phi)^{-1} \Phi^\top y \quad (9.70)$$

解释为投影到由特征矩阵 Φ 的列张成的 K 维子空间 \mathbb{R}^N 上；见第3.8.2节。如果我们在构建特征矩阵 Φ 时使用的特征函数 ϕ_k 是正交的（见第3.7节），那么我们得到一个特殊情况，其中 Φ 的列形成了一个正交基（见第3.5节），使得 $\Phi^\top \Phi = I$ 。这将导致投影

$$\Phi(\Phi^\top \Phi)^{-1} \Phi^\top y = \Phi \Phi^\top y = \left(\sum_{k=1}^K \phi_k \phi_k^\top \right) y \quad (9.71)$$

因此，最大似然投影仅仅是将 y 投影到各个基向量 ϕ_k 上的和，即 Φ 的列。此外，由于基的正交性，不同特征之间的耦合消失了。在信号处理中，许多流行的基函数，如小波和傅里叶基，都是正交基函数。

注释。当基不是正交的，可以使用格拉姆-施密特过程（见第3.8.3节和Strang, 2003）将一组线性独立的基函数转换为正交基。

< 上一章节

下一章节 >

9.3 贝叶斯线性回归

9.5 拓展阅读



9.5 拓展阅读

在本章中，我们讨论了具有高斯似然和共轭高斯先验的线性回归模型的参数。这使得我们可以进行闭式贝叶斯推断。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，在二分类设置中，我们观察到的只有两个可能的（分类）结果，高斯似然在这种情况下是不合适的。相反，我们可以选择伯努利似然，它将返回预测标签为1（或0）的概率。我们推荐Barber (2012)、Bishop (2006) 和 Murphy (2012) 的书籍，以深入了解分类问题。

另一个例子是计数数据。计数是非负整数，在这种情况下，二项分布或泊松分布比高斯分布是更好的选择。所有这些例子都属于广义线性模型（Generalized Linear Models, GLM）的范畴，这是线性回归的一种灵活推广，允许响应变量具有高斯分布之外的误差分布。广义线性模型通过一个平滑且可逆的函数 $\sigma(\cdot)$ 将线性模型与观测值联系起来，该函数可能是非线性的，使得 $y = \sigma(f(x))$ ，其中 $f(x) = \theta^\top \phi(x)$ 是式(9.13)中的线性回归模型。因此，我们可以将广义线性模型视为函数复合 $y = \sigma \circ f$ ，其中 f 是线性回归模型， σ 是激活函数。需要注意的是，尽管我们在这里讨论的是“广义线性模型”，但输出 y 不再是参数 θ 的线性函数。在逻辑回归中，我们选择逻辑函数 $\sigma(f) = \frac{1}{1+\exp(-f)} \in [0, 1]$ ，它可以被解释为伯努利随机变量 $y \in \{0, 1\}$ 观测值为1的概率。函数 $\sigma(\cdot)$ 被称为转移函数或激活函数，其逆被称为典型链接函数。从这个角度来看，普通线性回归的激活函数仅仅是恒等函数。

此外，很明显广义线性模型是（深度）前馈神经网络的构建块：如果我们考虑一个广义线性模型 $y = \sigma(Ax + b)$ ，其中 A 是权重矩阵， b 是偏置向量，我们将这个广义线性模型识别为一个具有激活函数 $\sigma(\cdot)$ 的单层神经网络。我们可以通过以下方式递归组合这些函数：

$$x_{k+1} = f_k(x_k) \quad f_k(x_k) = \sigma_k(A_k x_k + b_k) \quad (9.72)$$

其中 $k = 0, \dots, K - 1$ ， x_0 是输入特征， $x_K = y$ 是观测输出，使得 $f_{K-1} \circ \dots \circ f_0$ 是一个 K 层深度神经网络。因此，这个深度神经网络的构建块是式(9.72)中定义的广义线性模型。关于GLM和深度网络之间关系的详细介绍，可以参考 <https://tinyurl.com/glm-dnn>。

神经网络 (Bishop, 1995; Goodfellow et al., 2016) 比线性回归模型具有更高的表达能力和灵活性。然而，最大似然参数估计是一个非凸优化问题，而在完全贝叶斯设置

中对参数进行边缘化是解析上不可行的。我们在前面的内容中简要提到，一个参数分布可以诱导一个函数分布。高斯过程（Gaussian Process, GP）是一种回归模型，其中函数分布的概念是核心。高斯过程通过利用核技巧（Schölkopf and Smola, 2002），直接在函数空间上放置分布，而无需通过参数的“绕道”。核技巧允许我们通过查看对应的输入 x_i, x_j 来计算两个函数值 $f(x_i), f(x_j)$ 之间的内积。高斯过程与贝叶斯线性回归和支撑向量回归密切相关，也可以被解释为贝叶斯神经网络的一个单隐藏层版本，其中隐藏单元的数量趋于无穷大（Neal, 1996; Williams, 1997）。关于高斯过程的优秀介绍可以参考MacKay (1998) 和Rasmussen and Williams (2006)。

我们在本章的讨论中专注于高斯参数先验，因为它们允许我们在线性回归模型中进行闭式推理。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。考虑一个输入 $\mathbf{x} \in \mathbb{R}^D$ 的设置，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，我们可以选择一个强制稀疏性的参数先验，即一个试图将尽可能多的参数设置为0的先验（变量选择）。这个先验提供了一个比高斯先验更强的正则化器，通常可以提高模型的预测精度和可解释性。拉普拉斯先验是一个经常用于此目的的例子。具有拉普拉斯先验的线性回归模型等同于具有L1正则化的线性回归（LASSO）（Tibshirani, 1996）。拉普拉斯分布的峰值在零处非常尖锐（其一阶导数不连续），并且它将概率质量更集中在零附近，这比高斯分布更倾向于使参数为0。因此，非零参数对于回归问题是相关的，这也是我们所说的“变量选择”的原因。

< 上一章节

9.4 最大似然作为正交投影

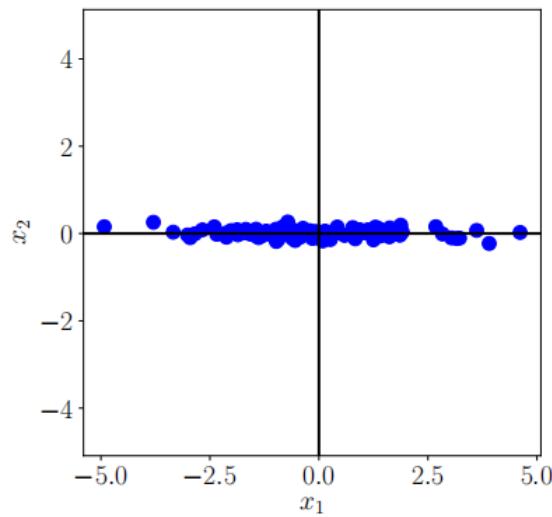


第10章 主成分分析与降维

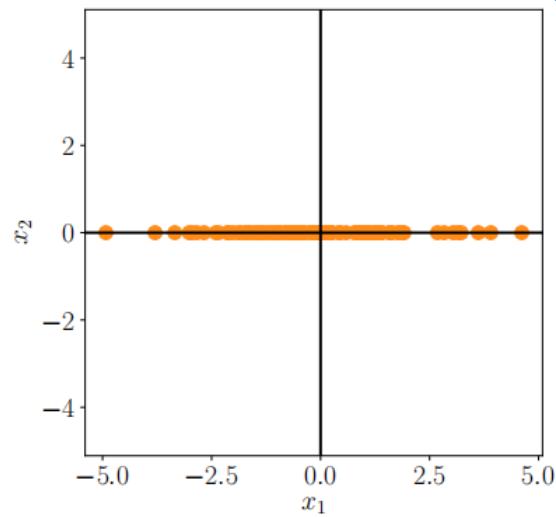
直接处理高维数据（如图像）会带来一些困难：难以分析、解释困难、几乎不可能可视化，而且（从实际角度来看）数据向量的存储成本可能很高。然而，高维数据往往具有我们可以利用的特性。例如，高维数据通常是冗余的，即许多维度是多余的，可以通过其他维度的组合来解释。此外，高维数据中的维度往往相互关联，从而使数据具有固有的低维结构。降维利用这种结构和相关性，使我们能够使用更紧凑的数据表示方式，理想情况下不会丢失信息。我们可以将降维视为一种压缩技术，类似于jpeg或mp3，这些是图像和音乐的压缩算法。

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由皮尔逊（Pearson, 1901）和霍特林（Hotelling, 1933）提出，至今已有100多年的历史，仍然是数据压缩和数据可视化最常用的技术之一。它还用于识别高维数据的简单模式、潜在因素和结构。在信号处理领域，PCA也被称为Karhunen-Loève变换。在本章中，我们将从基本原理出发推导PCA，利用我们对基和基变换（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。

降维通常利用高维数据（如图像）的一个特性，即它通常位于低维子空间上。图10.1给出了一个二维示例。尽管图10.1(a)中的数据并不完全位于一条直线上，但数据在 x_2 方向上的变化不大，因此我们可以将其视为几乎无损地位于一条直线上；见图10.1(b)。为了描述图10.1(b)中的数据，仅需要 x_1 坐标，且数据位于 \mathbb{R}^2 的一个一维子空间中。



(a) Dataset with x_1 and x_2 coordinates.



(b) Compressed dataset where only the x_1 coordinate is relevant.

图10.1说明：降维。**(a)**原始数据集在x2方向上变化不大。**(b)**来自**(a)**的数据可以单独用x1-坐标来表示，几乎没有损失。

< 上一章节

下一章节 >

第九章 线性回归

第十一章 密度估计和混合Gauss模型



10.1 问题设定

在PCA（主成分分析）中，我们关注的是找到数据点 \mathbf{x}_n 的投影 $\tilde{\mathbf{x}}_n$ ，这些投影应尽可能与原始数据点相似，但具有显著降低的内在维度。图10.1给出了这种情况的一个示意图。

更具体地说，我们考虑一个独立同分布的数据集 $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ，其中 $\mathbf{x}_n \in \mathbb{R}^D$ ，均值为0，且具有数据协方差矩阵（6.42）

(10.1)

$$S = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^\top.$$

此外，我们假设存在一个低维压缩表示（编码）

(10.2)

$$\mathbf{z}_n = \mathbf{B}^\top \mathbf{x}_n \in \mathbb{R}^M$$

其中， \mathbf{x}_n 的投影矩阵定义为

(10.3)

$$\mathbf{B} := [\mathbf{b}_1, \dots, \mathbf{b}_M] \in \mathbb{R}^{D \times M}.$$

我们假设 \mathbf{B} 的列是标准正交的（根据定义3.7），即 $\mathbf{b}_i^\top \mathbf{b}_j = 0$ 当且仅当 $i \neq j$ ，且 $\mathbf{b}_i^\top \mathbf{b}_i = 1$ 。我们寻找一个 M 维子空间 $\mathbf{U} \subseteq \mathbb{R}^D$ ，其中 $\dim(\mathbf{U}) = M < D$ ，以便将数据投影到这个子空间上。我们用 $\tilde{\mathbf{x}}_n \in \mathbf{U}$ 表示投影后的数据，其坐标（相对于 \mathbf{U} 的基向量 $\mathbf{b}_1, \dots, \mathbf{b}_M$ ）由 \mathbf{z}_n 给出。我们的目标是找到 $\bar{\mathbf{x}}_n \in \mathbb{R}^D$ （或等价地，找到编码 \mathbf{z}_n 和基向量 $\mathbf{b}_1, \dots, \mathbf{b}_M$ ），使得它们由于压缩而与原始数据 \mathbf{x}_n 尽可能相似。

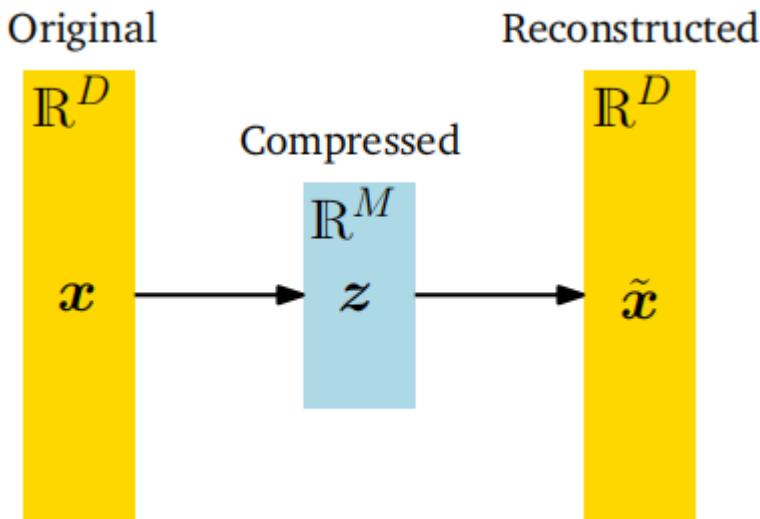


图10.2PCA。在PCA中，我们找到了原始数据 x 的压缩版本 z 。压缩后的数据可以重建成 \tilde{x} ，它存在于原始数据空间中，但具有比 x 在内的低维表示。

示例10.1（坐标表示/编码）

考虑 \mathbb{R}^2 ，其标准基为 $e_1 = [1, 0]^\top, e_2 = [0, 1]^\top$ 。根据第2章的内容，我们知道 $x \in \mathbb{R}^2$ 可以表示为这些基向量的线性组合，例如

(10.4)

$$\begin{bmatrix} 5 \\ 3 \end{bmatrix} = 5e_1 + 3e_2.$$

然而，当我们考虑形式为

(10.5)

$$\tilde{x} = \begin{bmatrix} 0 \\ z \end{bmatrix} \in \mathbb{R}^2, \quad z \in \mathbb{R},$$

的向量时，它们总可以表示为 $0e_1 + ze_2$ 。为了表示这些向量，只需记住/存储 \tilde{x} 相对于 e_2 向量的坐标/编码 z 。

更准确地说， \tilde{x} 向量的集合（具有标准的向量加法和标量乘法）形成了一个向量子空间 U （参见第2.4节），其中 $\dim(U) = 1$ ，因为 $U = \text{span}[e_2]$ 。

在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。在10.3中，我们将给出PCA的另一种推导，即最小化原始数据 x_n 与其投影 \tilde{x}_n 之间的平方重构误差 $\|x_n - \tilde{x}_n\|^2$ 。

图10.2展示了我们在主成分分析（PCA）中考虑的设置，其中 z 代表压缩数据 \tilde{x} 的低维表示，并扮演瓶颈的角色，控制着 x 和 \tilde{x} 之间可以流动的信息量。在PCA中，我们考虑原始数据 x 与其低维编码 z 之间的线性关系，使得 $z = B^\top x$ 且 $\tilde{x} = Bz$ ，其中 B 是一个合适的矩阵。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一对操作。由 B 表示的线性映射可以视为解码器，它将低维编码 $z \in \mathbb{R}^M$ 映射回原始数据空间 \mathbb{R}^D 。类似地， B^\top 可以视为编码器，它将原始数据 x 编码为低维（压缩）编码 z 。

在本章中，我们将使用**MNIST**数字数据集作为反复出现的示例，该数据集包含60,000个手写数字0到9的示例。每个数字都是大小为 28×28 的灰度图像，即它包含784个像素，因此我们可以将该数据集中的每个图像解释为向量 $x \in \mathbb{R}^{784}$ 。这些数字的一些示例如图10.3所示。



图10.3来自**MNIST**数据集的手写数字示例。<http://yann.lecun.com/exdb/mnist/>.

下一章节 >

10.2 最大方差视角



10.2 最大方差视角

图10.1给出了一个二维数据集如何使用单个坐标来表示的例子。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，所以压缩后的数据与图10.1(a)中的原始数据相似。我们也可以选择忽略 x_1 坐标，但那样压缩后的数据将与原始数据非常不同，数据中的大量信息将会丢失。

如果我们将数据中的信息量解释为数据集“填充空间”的程度，那么我们可以通过观察数据的散布来描述数据中包含的信息。从第6.4.1节我们知道，方差是数据散布的一个指标，我们可以将PCA推导为一种降维算法，它通过最大化数据低维表示中的方差来尽可能保留信息。图10.4对此进行了说明。

考虑到第10.1节中讨论的设置，我们的目标是找到一个矩阵 B （见（10.3）），当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，该矩阵能够尽可能多地保留信息。在数据压缩后保留大部分信息，等价于在低维编码中捕获最大的方差量（Hotelling, 1933）。

备注。（数据中心化）对于（10.1）中的数据协方差矩阵，我们假设数据已经中心化。我们可以不失一般性地做出这个假设：假设 μ 是数据的均值。利用我们在第6.4.4节中讨论的方差的性质，我们得到

$$\mathbf{V}_z[z] = \mathbf{V}_x[B^\top(x - \mu)] = \mathbf{V}_x[B^\top x - B^\top \mu] = \mathbf{V}_x[B^\top x],$$

(10.6)

即，低维编码的方差不依赖于数据的均值。因此，在本节的其余部分中，我们不失一般性地假设数据的均值为 $\mathbf{0}$ 。在这个假设下，由于 $\mathbb{E}_z[z] = \mathbb{E}_x[B^\top x] = B^\top \mathbb{E}_x[x] = \mathbf{0}$ ，低维编码的均值也为 $\mathbf{0}$ 。

10.2.1 最大方差方向

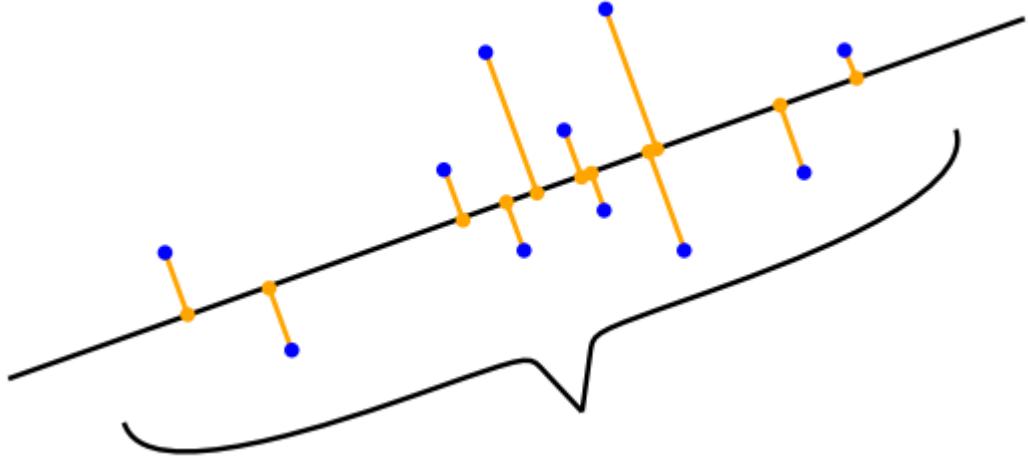


图10.4 PCA找到了一个低维子空间（线），当数据（蓝色）投影到这个子空间（橙色）时，它保持尽可能多的方差（数据的分布）。

我们使用顺序方法来最大化低维编码的方差。首先，我们寻找一个单独的向量 $\mathbf{b}_1 \in \mathbb{R}^D$ ，该向量能够最大化投影数据的方差，即我们旨在最大化 $\mathbf{z} \in \mathbb{R}^M$ 的第一个坐标 z_1 的方差，使得

(10.7)

$$V_1 := \text{V}[z_1] = \frac{1}{N} \sum_{n=1}^N z_{1n}^2$$

达到最大，其中我们利用了数据的独立同分布（i.i.d.）假设，并将 z_{1n} 定义为 $\mathbf{x}_n \in \mathbb{R}^D$ 的低维表示 $\mathbf{z}_n \in \mathbb{R}^M$ 的第一个坐标。注意， z_n 的第一个分量由下式给出：

$$z_{1n} = \mathbf{b}_1^\top \mathbf{x}_n ,$$

(10.8)

即，它是 \mathbf{x}_n 在由 \mathbf{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 (10.8) 代入 (10.7)，得到

(10.9a)

$$V_1 = \frac{1}{N} \sum_{n=1}^N (\mathbf{b}_1^\top \mathbf{x}_n)^2 = \frac{1}{N} \sum_{n=1}^N \mathbf{b}_1^\top \mathbf{x}_n \mathbf{x}_n^\top \mathbf{b}_1$$



$$= \mathbf{b}_1^\top \left(\frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^\top \right) \mathbf{b}_1 = \mathbf{b}_1^\top \mathbf{S} \mathbf{b}_1 ,$$

(10.9b)

其中 \mathbf{S} 是在(10.1)中定义的数据协方差矩阵。在(10.9a)中，我们使用了两个向量的点积关于其参数是对称的这一事实，即 $\mathbf{b}_1^\top \mathbf{x}_n = \mathbf{x}_n^\top \mathbf{b}_1$ 。

注意到，任意增加向量 \mathbf{b}_1 的幅度都会增加 V_1 ，即一个长度为两倍的 \mathbf{b}_1 向量可能导致 V_1 潜在地增加到四倍。因此，我们将所有解的范数限制为 $\|\mathbf{b}_1\|^2 = 1$ ，这导致了一个约束优化问题，我们在这个问题中寻求数据变化最大的方向。

在将解空间限制为单位向量的条件下，指向最大方差方向的向量 \mathbf{b}_1 可以通过以下约束优化问题找到：

(10.10)

$$\begin{aligned} & \max_{\mathbf{b}_1} \mathbf{b}_1^\top \mathbf{S} \mathbf{b}_1 \\ & \text{subject to } \|\mathbf{b}_1\|^2 = 1 \end{aligned}$$

根据第7.2节，我们得到Lagrange 函数

$$\mathcal{L}(\mathbf{b}_1, \lambda) = \mathbf{b}_1^\top \mathbf{S} \mathbf{b}_1 + \lambda_1 (1 - \mathbf{b}_1^\top \mathbf{b}_1)$$

(10.11)

来解决这个约束优化问题。 \mathcal{L} 关于 \mathbf{b}_1 和 λ_1 的偏导数分别为

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}_1} = 2\mathbf{b}_1^\top \mathbf{S} - 2\lambda_1 \mathbf{b}_1^\top, \quad \frac{\partial \mathcal{L}}{\partial \lambda_1} = 1 - \mathbf{b}_1^\top \mathbf{b}_1 ,$$

(10.12)

分别设置这些偏导数为0，我们得到关系式

$$\mathbf{S} \mathbf{b}_1 = \lambda_1 \mathbf{b}_1 ,$$

(10.13)

$$\mathbf{b}_1^\top \mathbf{b}_1 = 1 .$$

(10.14)

通过与特征值分解的定义（第4.4节）进行比较，我们发现 \mathbf{b}_1 是数据协方差矩阵 \mathbf{S} 的一个特征向量，而Lagrange乘子 λ_1 则扮演了相应特征值的角色。这个特征向量属性(10.13)允许我们将方差目标(10.10)重写为

$$V_1 = \mathbf{b}_1^\top \mathbf{S} \mathbf{b}_1 = \lambda_1 \mathbf{b}_1^\top \mathbf{b}_1 = \lambda_1 , \quad (10.15)$$

即，将数据投影到一维子空间上的方差等于与该子空间所跨越的基向量 \mathbf{b}_1 相关联的特征值。因此，为了最大化低维编码的方差，我们选择数据协方差矩阵中最大特征值所关联的基向量。这个特征向量被称为第一主成分。我们可以通过将坐标 z_{1n} 映射回数据空间来确定主成分 \mathbf{b}_1 在原始数据空间中的效果/贡献，这给了我们投影后的数据点在原始数据空间中。

$$\tilde{\mathbf{x}}_n = \mathbf{b}_1 z_{1n} = \mathbf{b}_1 \mathbf{b}_1^\top \mathbf{x}_n \in \mathbb{R}^D \quad (10.16)$$

备注：尽管 $\tilde{\mathbf{x}}_n$ 是一个 D 维向量，但它仅需要关于基向量 $\mathbf{b}_1 \in \mathbb{R}^D$ 的一个坐标 z_{1n} 来表示。

10.2.2 最大方差M维子空间

假设我们已经找到了与最大的 $m - 1$ 个特征值相关联的 \mathbf{S} 的 $m - 1$ 个特征向量，即前 $m - 1$ 个主成分。由于 \mathbf{S} 是对称的，根据谱定理（定理4.15），我们可以使用这些特征向量来构造一个 $(m - 1)$ 维子空间的正交特征基，该子空间位于 \mathbb{R}^D 中。一般来说，第 m 个主成分可以通过从数据中减去前 $m - 1$ 个主成分 $\mathbf{b}_1, \dots, \mathbf{b}_{m-1}$ 的影响来找到，从而尝试找到能够压缩剩余信息的主成分。然后我们得到新的数据矩阵

$$\hat{\mathbf{X}} := \mathbf{X} - \sum_{i=1}^{m-1} \mathbf{b}_i \mathbf{b}_i^\top \mathbf{X} = \mathbf{X} - \mathbf{B}_{m-1} \mathbf{X} ,$$

(10.17)

其中 $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ 包含作为列向量的数据点，而 $\mathbf{B}_{m-1} := \sum_{i=1}^{m-1} \mathbf{b}_i \mathbf{b}_i^\top$ 是一个投影矩阵，它将数据投影到由 $\mathbf{b}_1, \dots, \mathbf{b}_{m-1}$ 所跨越的子空间上。备注（符号）。在本章中，我们没有遵循将数据 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 作为数据矩阵的行的惯例，而是将它们定义为 \mathbf{X} 的列。这意味着我们的数据矩阵 \mathbf{X} 是一个 $D \times N$ 矩阵，而

不是传统的 $N \times D$ 矩阵。我们选择这样做的原因是代数运算可以平滑地进行，而无需转置矩阵或将向量重新定义为左乘矩阵的行向量。

为了找到第 m 个主成分，我们最大化方差

(10.18)

$$V_m = \text{V}[z_m] = \frac{1}{N} \sum_{n=1}^N z_{mn}^2 = \frac{1}{N} \sum_{n=1}^N (\mathbf{b}_m^\top \mathbf{x}_n)^2 = \mathbf{b}_m^\top \hat{\mathbf{S}} \mathbf{b}_m ,$$

受约束于 $\|\mathbf{b}_m\|^2 = 1$ ，其中我们遵循了与(10.9b)中相同的步骤，并将 $\hat{\mathbf{S}}$ 定义为变换数据集 $\hat{\mathcal{X}} := \{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_N\}$ 的数据协方差矩阵。与之前单独查看第一个主成分时一样，我们解决了一个约束优化问题，并发现最优解 \mathbf{b}_m 是 $\hat{\mathbf{S}}$ 的特征向量，该特征向量与 $\hat{\mathbf{S}}$ 的最大特征值相关联。

结果证明， \mathbf{b}_m 也是 \mathbf{S} 的特征向量。更一般地说， \mathbf{S} 和 $\hat{\mathbf{S}}$ 的特征向量集是相同的。由于 \mathbf{S} 和 $\hat{\mathbf{S}}$ 都是对称的，我们可以找到特征向量的正交归一基（谱定理4.15），即 \mathbf{S} 和 $\hat{\mathbf{S}}$ 都存在 D 个不同的特征向量。接下来，我们证明 \mathbf{S} 的每个特征向量都是 $\hat{\mathbf{S}}$ 的特征向量。假设我们已经找到了 $\hat{\mathbf{S}}$ 的特征向量 $\mathbf{b}_1, \dots, \mathbf{b}_{m-1}$ 。考虑 $\tilde{\mathbf{S}}$ （这里应该是 \mathbf{S} 的笔误）的一个特征向量 \mathbf{b}_i ，即 $\mathbf{S}\mathbf{b}_i = \lambda_i \mathbf{b}_i$ 。一般来说，

$$\begin{aligned} \hat{\mathbf{S}}\mathbf{b}_i &= \frac{1}{N} \hat{\mathbf{X}} \hat{\mathbf{X}}^\top \mathbf{b}_i = \frac{1}{N} (\mathbf{X} - \mathbf{B}_{m-1} \mathbf{X})(\mathbf{X} - \mathbf{B}_{m-1} \mathbf{X})^\top \mathbf{b}_i \\ &= (\mathbf{S} - \mathbf{S}\mathbf{B}_{m-1} - \mathbf{B}_{m-1}\mathbf{S} + \mathbf{B}_{m-1}\mathbf{S}\mathbf{B}_{m-1})\mathbf{b}_i . \end{aligned}$$

我们区分两种情况。如果 $i \geq m$ ，即 \mathbf{b}_i 是前 $m-1$ 个主成分之外的特征向量，那么 \mathbf{b}_i 与前 $m-1$ 个主成分正交，且 $\mathbf{B}_{m-1}\mathbf{b}_i = \mathbf{0}$ 。如果 $i < m$ ，即 \mathbf{b}_i 是前 $m-1$ 个主成分之一，那么 \mathbf{b}_i 是主子空间的基础向量，该子空间是 \mathbf{B}_{m-1} 投影的目标。由于 $\mathbf{b}_1, \dots, \mathbf{b}_{m-1}$ 是该主子空间的正交归一基（ONB），我们得到 $\mathbf{B}_{m-1}\mathbf{b}_i = \mathbf{b}_i$ 。这两种情况可以总结如下：

$$\mathbf{B}_{m-1}\mathbf{b}_i = \mathbf{b}_i \quad \text{如果 } i < m , \quad \mathbf{B}_{m-1}\mathbf{b}_i = \mathbf{0} \quad \text{如果 } i \geq m .$$

(10.20)

在 $i \geq m$ 的情况下，将(10.20)代入(10.19b)，我们得到 $\hat{\mathbf{S}}\mathbf{b}_i = (\mathbf{S} - \mathbf{B}_{m-1}\mathbf{S})\mathbf{b}_i = \mathbf{S}\mathbf{b}_i = \lambda_i \mathbf{b}_i$ ，即 \mathbf{b}_i 也是 $\hat{\mathbf{S}}$ 的特征向量，对应的特征值为 λ_i 。具体来说，

$$\hat{\mathbf{S}}\mathbf{b}_m = \mathbf{S}\mathbf{b}_m = \lambda_m \mathbf{b}_m .$$

(10.21)

方程 (10.21) 表明 b_m 不仅是 S 的特征向量，也是 \hat{S} 的特征向量。具体来说， λ_m 是 \hat{S} 的最大特征值，并且是 S 的第 m 大特征值，两者都与特征向量 b_m 相关联。

在 $i < m$ 的情况下，将 (10.20) 代入 (10.19b)，我们得到

(10.22)

$$\hat{S}b_i = (S - SB_{m-1} - B_{m-1}S + B_{m-1}SB_{m-1})b_i = 0 = 0b_i$$

这意味着 b_1, \dots, b_{m-1} 也是 \hat{S} 的特征向量，但它们与特征值 0 相关联，因此 b_1, \dots, b_{m-1} 构成了 \hat{S} 的零空间。总的来说， S 的每个特征向量也是 \hat{S} 的特征向量。但是，如果 S 的特征向量是 $(m-1)$ 维主子空间的一部分，那么 \hat{S} 的相关特征值为 0。

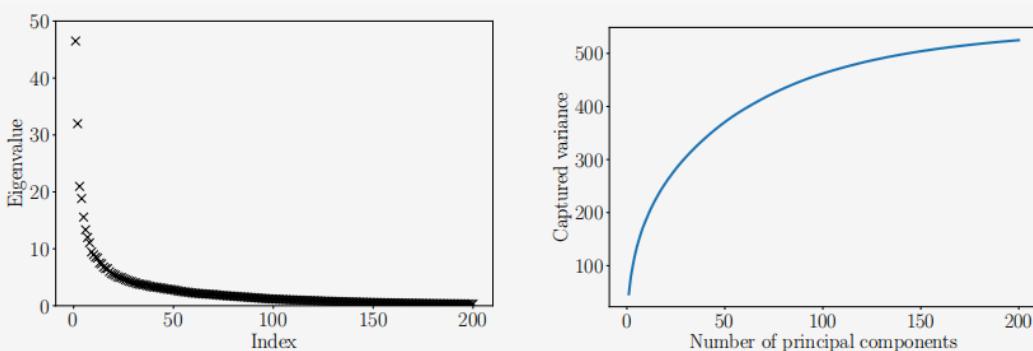
利用关系 (10.21) 和 $\mathbf{b}_m^\top \mathbf{b}_m = 1$ ，数据投影到第 m 个主成分上的方差是

$$V_m = \mathbf{b}_m^\top S \mathbf{b}_m \stackrel{(10.21)}{=} \lambda_m \mathbf{b}_m^\top \mathbf{b}_m = \lambda_m.$$

(10.23)

这意味着当数据投影到 M 维子空间时，其方差等于数据协方差矩阵对应特征向量的特征值之和。

例10.2 数字8的特征值



(a) Eigenvalues (sorted in descending order) of the data covariance matrix of all digits “8” in the MNIST training set.
 (b) Variance captured by the principal components.

图10.5 MNIST“8”的训练数据的属性。(a)特征值按降序排列；(b)由与最大特征值相关联的主成分捕获的方差。



我们取 MNIST 训练数据中所有的“8”数字，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。我们看到只有少数几个特征值显著不同于 0。因此，当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几个主成分捕获，如图 10.5(b) 所示。

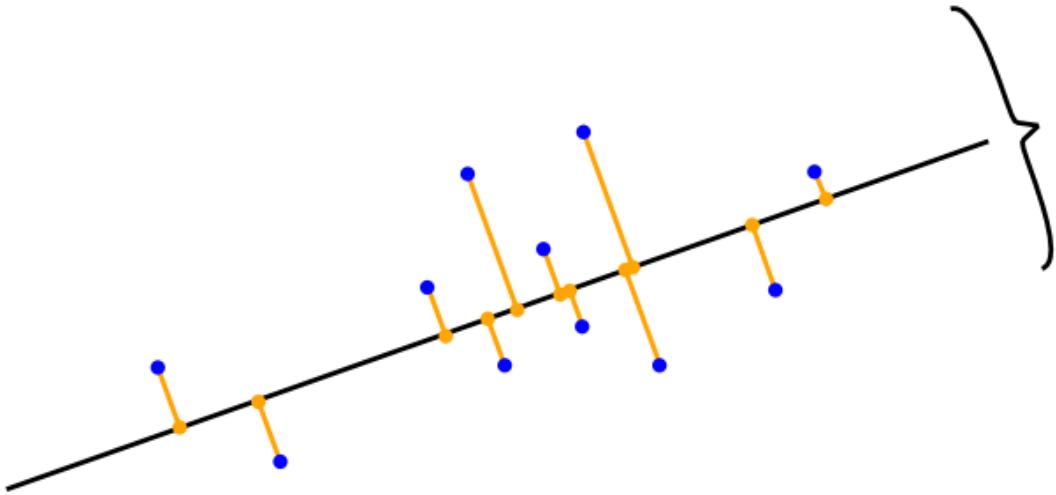


图10.6投影方法说明：找到一个子空间（线），尽量减少投影（橙色）和原始（蓝色）数据之间的差向量的长度。

总的来说，为了找到 \mathbb{R}^D 中的一个 M 维子空间，该子空间尽可能多地保留信息，PCA 告诉我们选择矩阵 B （在 (10.3) 中）的列作为数据协方差矩阵 S 的 M 个特征向量，这些特征向量与 M 个最大特征值相关联。PCA 用前 M 个主成分可以捕获的最大方差是

(10.24)

$$V_M = \sum_{m=1}^M \lambda_m,$$

其中 λ_m 是数据协方差矩阵 S 的 M 个最大特征值。因此，通过 PCA 进行数据压缩时损失的方差是

(10.25)

Instead of these absolute quantities, we can define the relative variance captured as $\frac{V}{M} V_D$, and the relative variance lost by compression as $1 - \frac{V_M}{V_D}$.



< 上一章节

下一章节 >

10.1 问题设定

10.3 投影视角



10.3 投影视角

接下来，我们将推导主成分分析（PCA）作为一种直接最小化平均重构误差的算法。这一视角使我们能够将PCA解释为实现最优线性自编码器的方法。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，以便尽可能多地保留信息。接下来，我们将关注原始数据 x_n 与其重构 \tilde{x}_n 之间的差向量，并最小化这一距离，以便 x_n 和 \tilde{x}_n 尽可能接近。图10.6展示了这一设置。

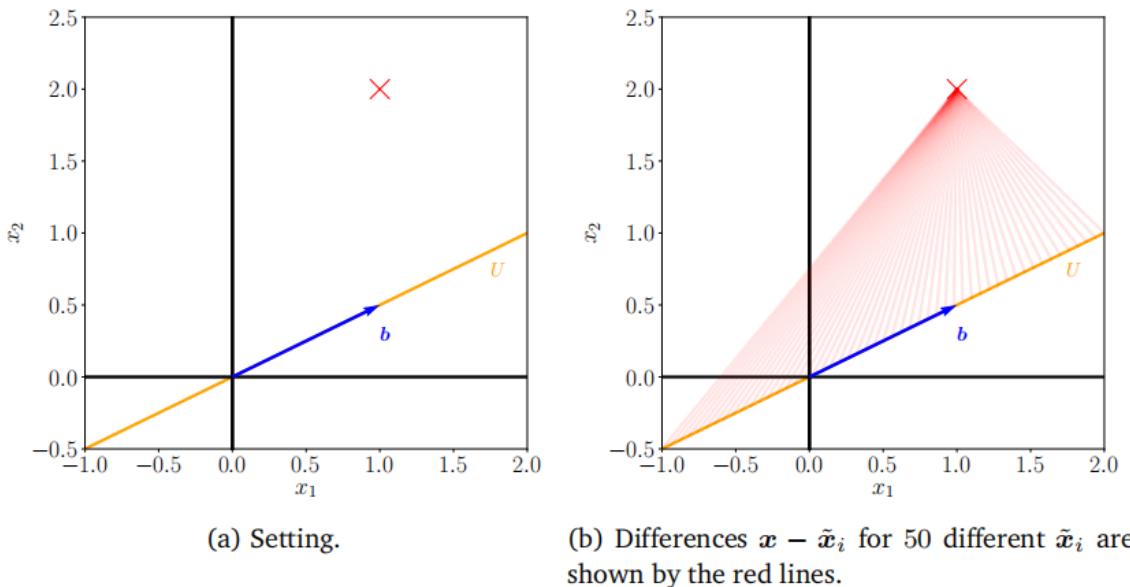


图10.7简化的投影设置。(a) 向量 $x \in \mathbb{R}^2$ (红十字) 应投影到由 b 跨越的一维子空间 $U \subseteq \mathbb{R}^2$ 上。(b) 表示 x 和一些候选向量 \tilde{x} 之间的差分向量。

10.3.1 设定与目标

假设有一个（有序的）正交归一基（ONB） $B = (\mathbf{b}_1, \dots, \mathbf{b}_D)$ 在 \mathbb{R}^D 上，即当且仅当 $i = j$ 时， $\mathbf{b}_i^\top \mathbf{b}_j = 1$ ，否则为0。从第2.5节我们知道，对于 \mathbb{R}^D 的一个基 $(\mathbf{b}_1, \dots, \mathbf{b}_D)$ ，任何 $x \in \mathbb{R}^D$ 都可以表示为该基向量的线性组合，即



$$\mathbf{x} = \sum_{d=1}^D \zeta_d \mathbf{b}_d = \sum_{m=1}^M \zeta_m \mathbf{b}_m + \sum_{j=M+1}^D \zeta_j \mathbf{b}_j$$

(10.26)

其中, $\zeta_d \in \mathbb{R}$ 是适当的坐标。

我们感兴趣的是找到向量 $\tilde{\mathbf{x}} \in \mathbb{R}^D$, 这些向量位于较低维度的子空间 $U \subseteq \mathbb{R}^D$ 中, 且 $\dim(U) = M$, 使得

(10.27)

$$\bar{\mathbf{x}} = \sum_{m=1}^M z_m \mathbf{b}_m \in U \subseteq \mathbb{R}^D$$

尽可能接近 \mathbf{x} 。请注意, 此时我们需要假设 $\tilde{\mathbf{x}}$ 的坐标 z_m 与 \mathbf{x} 的坐标 ζ_m 不相同。接下来, 我们将使用这种 $\tilde{\mathbf{x}}$ 的表示方式, 来找到最优的坐标 z 和基向量 $\mathbf{b}_1, \dots, \mathbf{b}_M$, 使得 $\tilde{\mathbf{x}}$ 尽可能接近原始数据点 \mathbf{x} , 即我们的目标是最小化 (Euclid) 距离 $\|\mathbf{x} - \bar{\mathbf{x}}\|$ 。图 10.7 展示了这一设定。

不失一般性, 我们假设数据集 $\mathcal{X} = \{x_1, \dots, x_N\}$, 其中 $x_n \in \mathbb{R}^D$, 以 0 为中心, 即 $E[X] = \mathbf{0}$ 。如果不假设均值为 $\mathbf{0}$, 我们也能得到相同的解, 但表示会更加复杂。

我们感兴趣的是找到 χ 到较低维度子空间 $U \subseteq \mathbb{R}^D$ (其中 $\dim(U) = M$) 的最佳线性投影, 该子空间具有正交归一基向量 $\mathbf{b}_1, \dots, \mathbf{b}_M$ 。我们将这个子空间 U 称为主子空间。数据点的投影表示为

(10.28)

$$\tilde{x}_n := \sum_{m=1}^M z_{mn} \mathbf{b}_m = B z_n \in \mathbb{R}^D,$$

其中 $z_n := [z_{1n}, \dots, z_{Mn}]^\top \in \mathbb{R}^M$ 是 \tilde{x}_n 相对于基 $(\mathbf{b}_1, \dots, \mathbf{b}_M)$ 的坐标向量。更具体地说, 我们希望 \tilde{x}_n 尽可能接近 x_n 。

在后续中, 我们使用的相似度度量是 \mathbf{x} 和 $\tilde{\mathbf{x}}$ 之间的平方距离 (Euclid 范数) $\|\mathbf{x} - \tilde{\mathbf{x}}\|^2$ 。因此, 我们将目标定义为最小化平均平方 Euclid 距离 (重构误差) (Pearson, 1901)

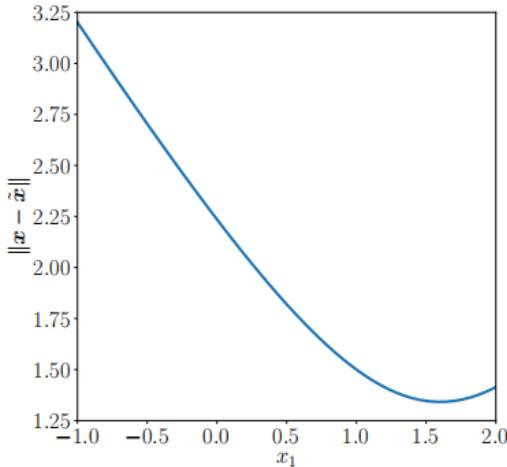


$$J_M := \frac{1}{N} \sum_{n=1}^N \| \mathbf{x}_n - \bar{\mathbf{x}}_n \|^2,$$

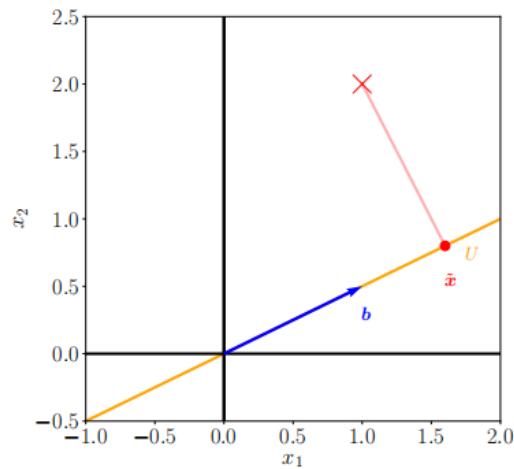
(10.29)

其中我们明确指出，我们将数据投影到的子空间的维度是 M 。为了找到这种最优线性投影，我们需要找到主子空间的正交归一基以及在该基下投影的坐标 $\mathbf{z}_n \in \mathbb{R}^M$ 。

为了找到坐标 \mathbf{z}_n 和主子空间的正交归一基，我们采用两步法。首先，我们针对给定的正交归一基 $(\mathbf{b}_1, \dots, \mathbf{b}_M)$ 优化坐标 \mathbf{z}_n ；其次，我们找到最优的正交归一基。



(a) Distances $\| \mathbf{x} - \tilde{\mathbf{x}} \|$ for some $\tilde{\mathbf{x}} = z_1 \mathbf{b} \in U = \text{span}[\mathbf{b}]$; see panel (b) for the setting.



(b) The vector $\tilde{\mathbf{x}}$ that minimizes the distance in panel (a) is its orthogonal projection onto U . The coordinate of the projection $\tilde{\mathbf{x}}$ with respect to the basis vector \mathbf{b} that spans U is the factor we need to scale \mathbf{b} in order to “reach” $\tilde{\mathbf{x}}$.

图10.8向量 $\mathbf{x} \in \mathbb{R}^2$ 在一维子空间上的最优投影（从图10.7开始的延续）。(a) 距离 $\|\mathbf{x} - \tilde{\mathbf{x}}\|$ 为一些 $\mathbf{x} \in U$ 。(b) 正交投影和最优坐标。

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_{1n}, \dots, z_{Mn} ，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。从几何上讲，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，这种表示使 $\tilde{\mathbf{x}} - \mathbf{x}$ 之间的距离最小化。从图 10.7(b) 可以清楚地看出，这将是正交投影，接下来我们将确切地展示这一点。

我们假设 $U \subseteq \mathbb{R}^D$ 的一个标准正交基 (ONB) 为 $(\mathbf{b}_1, \dots, \mathbf{b}_M)$ 。为了找到关于这个基的最优坐标 z_m , 我们需要偏导数

(10.30a)

$$\begin{aligned}\frac{\partial J_M}{\partial z_{in}} &= \frac{\partial J_M}{\partial \tilde{x}_n} \frac{\partial \tilde{x}_n}{\partial z_{in}}, \\ \frac{\partial J_M}{\partial \tilde{x}_n} &= -\frac{2}{N}(\mathbf{x}_n - \tilde{\mathbf{x}}_n)^\top \in \mathbb{R}^{1 \times D},\end{aligned}$$

(10.30b)

(10.30c)

$$\frac{\partial \tilde{\mathbf{x}}_n}{\partial z_{in}} \stackrel{(10.28)}{=} \frac{\partial}{\partial z_{in}} \left(\sum_{m=1}^M z_{mn} \mathbf{b}_m \right) = \mathbf{b}_i$$

对于 $i = 1, \dots, M$, 我们得到

$$\frac{\partial J_M}{\partial z_{in}} \stackrel{(10.30b)}{=} -\frac{2}{N}(\mathbf{x}_n - \bar{\mathbf{x}}_n)^\top \mathbf{b}_i \stackrel{(10.28)}{=} -\frac{2}{N} \left(\mathbf{x}_n - \sum_{m=1}^M z_{mn} \mathbf{b}_m \right)^\top \mathbf{b}_i$$

(10.31a)

$$\stackrel{\text{ONB}}{=} -\frac{2}{N}(\mathbf{x}_n^\top \mathbf{b}_i - z_{in} \mathbf{b}_i^\top \mathbf{b}_i) = -\frac{2}{N}(\mathbf{x}_n^\top \mathbf{b}_i - z_{in}).$$

(10.31b)

由于 $\mathbf{b}_i^\top \mathbf{b}_i = 1$ 。将偏导数设为 0 立即得到最优坐标

$$z_{in} = \mathbf{x}_n^\top \mathbf{b}_i = \mathbf{b}_i^\top \mathbf{x}_n$$

(10.32)

对于 $i = 1, \dots, M$ 和 $n = 1, \dots, N$ 。这意味着投影 \tilde{x}_n 的最优坐标 z_{in} 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的坐标 (见第 3.8 节)。因此:

- \mathbf{x}_n 的最优线性投影 \tilde{x}_n 是正交投影
- \tilde{x}_n 相对于基 $(\mathbf{b}_1, \dots, \mathbf{b}_M)$ 的坐标是 \mathbf{x}_n 在主子空间上的正交投影的坐标
- 正交投影是在给定目标 (10.29) 下的最佳线性映射

- x 在 (10.26) 中的坐标 ζ_m 和 \tilde{x} 在 (10.27) 中的坐标 z_m 对于 $m = 1, \dots, M$ 必须相同，因为 $U^\perp = \text{span}[\mathbf{b}_{M+1}, \dots, \mathbf{b}_D]$ 是 $U = \text{span}[\mathbf{b}_1, \dots, \mathbf{b}_M]$ 的正交补（见第 3.6 节）

注记（具有标准正交基向量的正交投影）。让我们简要回顾一下第3.8节中的正交投影。如果 $(\mathbf{b}_1, \dots, \mathbf{b}_D)$ 是 \mathbb{R}^D 的一个标准正交基，那么

$$\tilde{\mathbf{x}} = \mathbf{b}_j (\mathbf{b}_j^\top \mathbf{b}_j)^{-1} \mathbf{b}_j^\top \mathbf{x} = \mathbf{b}_j \mathbf{b}_j^\top \mathbf{x} \in \mathbb{R}^D$$

(10.33)

是向量 \mathbf{x} 在第 j 个基向量所张成的子空间上的正交投影，并且 $z_j = \mathbf{b}_j^\top \mathbf{x}$ 是该投影相对于基向量 \mathbf{b}_j （该基向量张成该子空间）的坐标，因为 $z_j \mathbf{b}_j = \tilde{\mathbf{x}}$ 。图10.8(b)展示了这种设置。

更一般地，如果我们想要将向量投影到 \mathbb{R}^D 的一个 M 维子空间上，我们可以得到向量 x 在由标准正交基向量 $\mathbf{b}_1, \dots, \mathbf{b}_M$ 所张成的 M 维子空间上的正交投影为

(10.34)

$$\tilde{\mathbf{x}} = \mathbf{B} (\underbrace{\mathbf{B}^\top \mathbf{B}}_{=I})^{-1} \mathbf{B}^\top \mathbf{x} = \mathbf{B} \mathbf{B}^\top \mathbf{x} ,$$

其中我们定义了 $\mathbf{B} := [\mathbf{b}_1, \dots, \mathbf{b}_M] \in \mathbb{R}^{D \times M}$ 。关于有序基 $(\mathbf{b}_1, \dots, \mathbf{b}_M)$ 的该投影的坐标是 $z := \mathbf{B}^\top \mathbf{x}$ ，如第3.8节所述。

我们可以将这些坐标视为投影向量在由 $(\mathbf{b}_1, \dots, \mathbf{b}_M)$ 定义的新坐标系中的表示。注意，虽然 $\tilde{\mathbf{x}} \in \mathbb{R}^D$ ，但我们只需要 M 个坐标 z_1, \dots, z_M 来表示这个向量；关于基向量 $(\mathbf{b}_{M+1}, \dots, \mathbf{b}_D)$ 的其他 $D - M$ 个坐标总是 0。

到目前为止，我们已经证明了对于给定的标准正交基（ONB），我们可以通过将向量正交投影到主子空间上来找到 $\tilde{\mathbf{x}}$ 的最优坐标。接下来，我们将确定什么是最好的基。

10.3.3 寻找主子空间的基

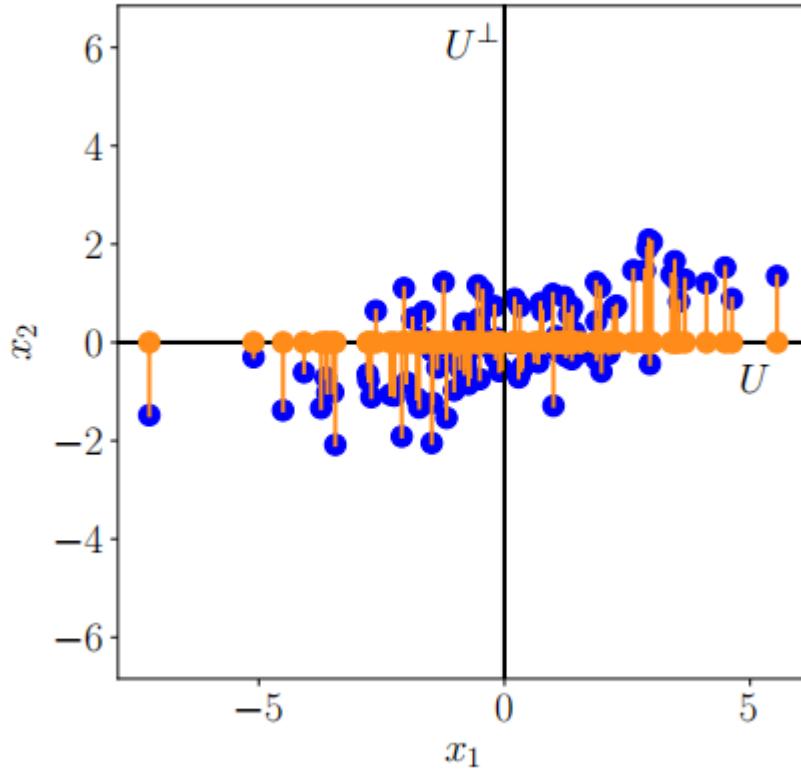


图10.9正交投影和位移向量。当将数据点 \mathbf{x}_n （蓝色）投影到子空间 U_1 上时，我们得到了 $\tilde{\mathbf{x}}_n$ （橙色）。位移向量 $\tilde{\mathbf{x}}_n - \mathbf{x}_n$ 完全位于 U_1 的正交补体 U_2 中

为了确定主子空间的基向量 $\mathbf{b}_1, \dots, \mathbf{b}_M$ ，我们使用到目前为止的结果重新表述损失函数（10.29），这将有助于我们更容易地找到基向量。为了重新表述损失函数，我们利用之前的结果得到

$$\tilde{\mathbf{x}}_n = \sum_{m=1}^M z_{mn} \mathbf{b}_m \stackrel{(10.32)}{=} \sum_{m=1}^M (\mathbf{x}_n^\top \mathbf{b}_m) \mathbf{b}_m .$$

(10.35)

现在我们利用点积的对称性，得到

(10.36)

$$\tilde{\mathbf{x}}_n = \left(\sum_{m=1}^M \mathbf{b}_m \mathbf{b}_m^\top \right) \mathbf{x}_n .$$

由于我们一般可以将原始数据点 \mathbf{x}_n 表示为所有基向量的线性组合，因此有



$$\mathbf{x}_n = \sum_{d=1}^D z_{dn} \mathbf{b}_d \stackrel{(10.32)}{=} \sum_{d=1}^D (\mathbf{x}_n^\top \mathbf{b}_d) \mathbf{b}_d = \left(\sum_{d=1}^D \mathbf{b}_d \mathbf{b}_d^\top \right) \mathbf{x}_n \quad (10.37)$$

$$= \left(\sum_{m=1}^M \mathbf{b}_m \mathbf{b}_m^\top \right) \mathbf{x}_n + \left(\sum_{j=M+1}^D \mathbf{b}_j \mathbf{b}_j^\top \right) \mathbf{x}_n, \quad (10.37)$$

其中我们将包含 D 项的求和拆分为 M 项和 $D - M$ 项的求和。根据这个结果，我们发现位移向量 $\mathbf{x}_n - \tilde{\mathbf{x}}_n$ ，即原始数据点与其投影之间的差向量是

(10.38a)

$$\begin{aligned} \mathbf{x}_n - \bar{\mathbf{x}}_n &= \left(\sum_{j=M+1}^D \mathbf{b}_j \mathbf{b}_j^\top \right) \mathbf{x}_n \\ &= \sum_{j=M+1}^D (\mathbf{x}_n^\top \mathbf{b}_j) \mathbf{b}_j. \end{aligned}$$

(10.38b)

这意味着这个差正好是数据点在主子空间正交补上的投影：我们识别出(10.38a)中的矩阵 $\sum_{j=M+1}^D \mathbf{b}_j \mathbf{b}_j^\top$ 是执行这个投影的投影矩阵。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。

注记（低秩近似）。在(10.38a)中，我们看到将 x 投影到 \tilde{x} 的投影矩阵由

(10.39)

$$\sum_{m=1}^M \mathbf{b}_m \mathbf{b}_m^\top = \mathbf{B} \mathbf{B}^\top.$$

给出。由于它是由秩一矩阵 $\mathbf{b}_m \mathbf{b}_m^\top$ 的和构成的，我们可以看到 $\mathbf{B} \mathbf{B}^\top$ 是对称的且秩为 M 。因此，平均平方重建误差也可以写为

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \bar{\mathbf{x}}_n\|^2 &= \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \mathbf{B} \mathbf{B}^\top \mathbf{x}_n\|^2 \\ &= \frac{1}{N} \sum_{n=1}^N \|(\mathbf{I} - \mathbf{B} \mathbf{B}^\top) \mathbf{x}_n\|^2. \end{aligned}$$

(10.40a)

寻找正交归一基向量 $\mathbf{b}_1, \dots, \mathbf{b}_M$, 以最小化原始数据 \mathbf{x}_n 与其投影 $\tilde{\mathbf{x}}_n$ 之间的差异, 等价于找到单位矩阵 \mathbf{I} 的最佳秩 M 近似 $\mathbf{B}\mathbf{B}^\top$ (参见第4.6节)。

(10.40b)

寻找正交归一化基向量 $\mathbf{b}_1, \dots, \mathbf{b}_M$, 这些向量能最小化原始数据 \mathbf{x}_n 与它们投影 $\tilde{\mathbf{x}}_n$ 之间的差异, 这等价于找到单位矩阵 \mathbf{I} 的最佳秩 M 近似 $\mathbf{B}\mathbf{B}^\top$ (参见第4.6节)。

现在我们有了所有工具来重新表述损失函数 (10.29)。

(10.41)

$$J_M = \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \bar{\mathbf{x}}_n\|^2 \stackrel{(10.38b)}{=} \frac{1}{N} \sum_{n=1}^N \left\| \sum_{j=M+1}^D (\mathbf{b}_j^\top \mathbf{x}_n) \mathbf{b}_j \right\|^2.$$

我们现在明确计算平方范数, 并利用 \mathbf{b}_j 形成正交归一基 (ONB) 的事实, 得到

(10.42a)

$$\begin{aligned} J_M &= \frac{1}{N} \sum_{n=1}^N \sum_{j=M+1}^D (\mathbf{b}_j^\top \mathbf{x}_n)^2 = \frac{1}{N} \sum_{n=1}^N \sum_{j=M+1}^D \mathbf{b}_j^\top \mathbf{x}_n \mathbf{b}_j^\top \mathbf{x}_n \\ &= \frac{1}{N} \sum_{n=1}^N \sum_{j=M+1}^D \mathbf{b}_j^\top \mathbf{x}_n \mathbf{x}_n^\top \mathbf{b}_j, \end{aligned}$$

(10.42b)

在最后一步中, 我们利用了点积的对称性来写作 $\mathbf{b}_j^\top \mathbf{x}_n = \mathbf{x}_n^\top \mathbf{b}_j$ 。现在我们交换求和顺序, 得到

$$\begin{aligned} J_M &= \sum_{j=M+1}^D \mathbf{b}_j^\top \underbrace{\left(\frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^\top \right)}_{=: \mathbf{S}} \mathbf{b}_j = \sum_{j=M+1}^D \mathbf{b}_j^\top \mathbf{S} \mathbf{b}_j \quad (10.43a) \\ &= \sum_{j=M+1}^D \text{tr}(\mathbf{b}_j^\top \mathbf{S} \mathbf{b}_j) = \sum_{j=M+1}^D \text{tr}(\mathbf{S} \mathbf{b}_j \mathbf{b}_j^\top) = \text{tr} \left(\underbrace{\left(\sum_{j=M+1}^D \mathbf{b}_j \mathbf{b}_j^\top \right)}_{\text{投影矩阵}} \mathbf{S} \right), \end{aligned}$$



这里我们利用了迹算子 $\text{tr}(\cdot)$ (见(4.18)) 的线性性质以及对其参数循环置换的不变性。由于我们假设数据集是居中的, 即 $\mathbb{E}[\mathcal{X}] = \mathbf{0}$, 我们将 S 识别为数据协方差矩阵。由于 (10.43b) 中的投影矩阵是秩一矩阵 $b_j b_j^\top$ 的和, 因此它自身的秩为 $D - M$ 。

方程 (10.43a) 表明, 我们可以将平均平方重构误差等价地表述为数据协方差矩阵投影到主子空间正交补上的矩阵。因此, 最小化平均平方重构误差等价于最小化当我们忽略某个子空间 (即主子空间的正交补) 时数据的方差。等价地, 我们最大化保留在主子空间中的投影的方差, 这立即将投影损失与第10.2节中讨论的最大方差PCA公式联系起来。但这也意味着我们将获得与最大方差视角相同的解。因此, 我们省略了与第10.2节中给出的推导相同的部分, 并根据投影视角总结了前面的结果。

投影到 M 维主子空间上的平均平方重构误差为

(10.44)

$$J_M = \sum_{j=M+1}^D \lambda_j,$$

其中 λ_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。

Consequently, this means that the basis of the principal subspace comprises the eigenvectors b_1, \dots, b_M that are associated with the largest M eigenvalues of the data covariance matrix.

例 10.3 (MNIST 数字嵌入)

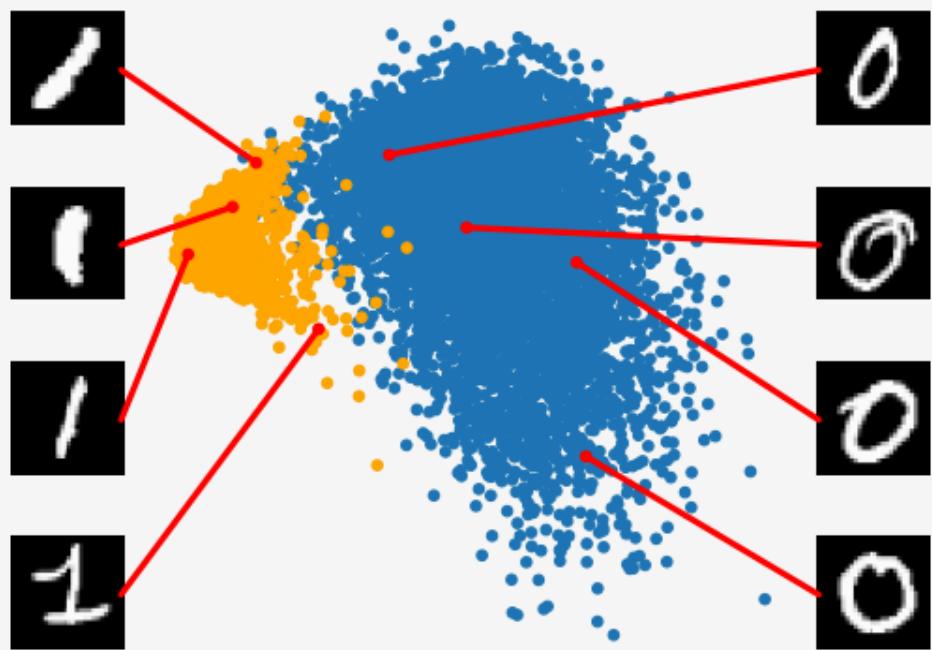


图10.10使用PCA将MNIST数字0（蓝色）和1（橙色）嵌入到二维主子空间中。主子空间中的数字“0”和“1”的四个嵌入用红色突出显示，并附有它们对应的原始数字。

图 10.10 展示了 MNIST 数字“0”和“1”的训练数据，这些数据被嵌入到由前两个主成分构成的向量子空间中。我们观察到“0”（蓝色点）和“1”（橙色点）之间相对清晰的分离，并且可以看到每个单独集群内的变化。在主成分子空间中，数字“0”和“1”的四个嵌入示例被用红色突出显示，并附有它们对应的原始数字。该图表明，“0”集合内的变化显著大于“1”集合内的变化。

< 上一章节

下一章节 >

10.2 最大方差视角

10.4 特征向量计算和低秩近似



10.4 特征向量计算和低秩近似

在前面的章节中，我们获得了主成分子空间的基础，即与数据协方差矩阵最大特征值相关联的特征向量

(10.45)

$$S = \frac{1}{N} \sum_{n=1}^N x_n x_n^\top = \frac{1}{N} X X^\top,$$
$$X = [x_1, \dots, x_N] \in \mathbb{R}^{D \times N}.$$

(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）。为了得到 S 的特征值（以及对应的特征向量），我们可以采用两种方法：

- 我们进行特征分解（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，并且可以分解为 $X X^\top$ （忽略因子 $\frac{1}{N}$ ），因此 S 的特征值是 X 的奇异值的平方。

更具体地说， X 的奇异值分解（SVD）由下式给出：

(10.47)

$$\underbrace{X}_{D \times N} = \underbrace{U}_{D \times D} \underbrace{\Sigma}_{D \times N} \underbrace{V^\top}_{N \times N},$$

其中 $U \in \mathbb{R}^{D \times D}$ 和 $V^\top \in \mathbb{R}^{N \times N}$ 是正交矩阵， $\Sigma \in \mathbb{R}^{D \times N}$ 是一个矩阵，其非零元素是奇异值 σ_i ($i \geq 0$)。由此可得

(10.48)

$$S = \frac{1}{N} X X^\top = \frac{1}{N} U \Sigma \underbrace{V^\top V^\top}_{=I_N} \Sigma^\top U^\top = \frac{1}{N} U \Sigma \Sigma^\top U^\top.$$

根据第 4.5 节的结果，我们得到 U 的列是 $X X^\top$ （因此也是 S ）的特征向量。此外， S 的特征值 λ_d 与 X 的奇异值之间的关系为



$$\lambda_d = \frac{\sigma_d^2}{N}.$$

(10.49)

S 的特征值与 X 的奇异值之间的这种关系，将最大方差观点（第 10.2 节）与奇异值分解联系起来。

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建误差），PCA选择在 (10.48) 中的 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，这样我们就可以将 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）提供了一种直接估计低维表示的方法。考虑最佳秩- M 近似

(10.50)

$$\bar{X}_M := \operatorname{argmin}_{\operatorname{rk}(A) \leq M} \|X - A\|_2 \in \mathbb{R}^{D \times N}$$

其中 X 的 $\|\cdot\|_2$ 是在 (4.93) 中定义的谱范数。Eckart-Young 定理指出， \tilde{X}_M 是通过在 SVD 中截断前 M 个奇异值得到的。换句话说，我们得到

(10.51)

$$\tilde{X}_M = \underbrace{U_M}_{D \times M} \underbrace{\Sigma_M}_{M \times M} \underbrace{V_M^\top}_{M \times N} \in \mathbb{R}^{D \times N}$$

其中， $U_M := [u_1, \dots, u_M] \in \mathbb{R}^{D \times M}$ 和 $V_M := [v_1, \dots, v_M] \in \mathbb{R}^{N \times M}$ 是正交矩阵， $\Sigma_M \in \mathbb{R}^{M \times M}$ 是对角矩阵，其对角线上的元素是 X 的 M 个最大奇异值。

10.4.2 实际应用方面

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非常重要。在理论上，正如我们在第 4.2 节中讨论的那样，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 - 鲁菲尼定理 (Ruffini, 1799; Abel, 1826) 指出，对于 5 次或更高次的多项式，这个问题不存在代数解。因此，在实际应用中，我们使用迭代方法来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。



在许多应用（如本章介绍的PCA）中，我们只需要少数几个特征向量。计算完整的特征分解然后丢弃所有特征值不在前几位的特征向量将是浪费的。事实证明，如果我们只对前几个特征向量（具有最大的特征值）感兴趣，那么直接优化这些特征向量的迭代过程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有效。幂迭代选择一个不在 S 的零空间中的随机向量 x_0 ，并遵循迭代

(10.52)

$$x_{k+1} = \frac{Sx_k}{\|Sx_k\|}, \quad k = 0, 1, \dots$$

这意味着在每个迭代中，向量 x_k 都与 S 相乘，然后进行归一化，即我们始终有 $\|x_k\| = 1$ 。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等，1999）就使用了这样的算法来根据网页的超链接对它们进行排名。

< 上一章节

下一章节 >

10.3 投影视角

10.5 高维PCA



10.5 高维PCA

为了进行PCA，我们需要计算数据的协方差矩阵。在 D 维空间中，数据协方差矩阵是一个 $D \times D$ 的矩阵。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 的三次方成正比。因此，正如我们之前讨论的那样，PCA在非常高维的情况下是不可行的。例如，如果我们的 x_n 是包含10,000个像素的图像（例如， 100×100 像素的图像），那么我们需要计算一个 $10,000 \times 10,000$ 的协方差矩阵的特征分解。以下，我们针对数据点数量远小于维度的情况（即 $N \ll D$ ）提供了一种解决方案。

假设我们有一个已居中的数据集 x_1, \dots, x_N ，其中 $x_n \in \mathbb{R}^D$ 。那么，数据的协方差矩阵定义为

$$S = \frac{1}{N}XX^\top \in \mathbb{R}^{D \times D},$$

(10.53)

其中 $X = [x_1, \dots, x_N]$ 是一个 $D \times N$ 的矩阵，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 的秩为 N ，因此它有 $D - N + 1$ 个特征值为0。直观地说，这意味着存在一些冗余。接下来，我们将利用这一点，将 $D \times D$ 的协方差矩阵转换为一个 $N \times N$ 的协方差矩阵，其所有特征值都是正的。

在PCA中，我们最终得到了特征向量方程

(10.54)

$$Sb_m = \lambda_m b_m, \quad m = 1, \dots, M,$$

其中 b_m 是主子空间的一个基向量。让我们重写这个方程：根据(10.53)中定义的 S ，我们得到

$$Sb_m = \frac{1}{N}XX^\top b_m = \lambda_m b_m.$$

(10.55)



我们现在从左侧乘以 $\mathbf{X}^\top \in \mathbb{R}^{N \times D}$, 得到 (10.56)

$$\frac{1}{N} \underbrace{\mathbf{X}^\top \mathbf{X} \mathbf{X}^\top}_{\substack{N \times N \\ =: e_m}} \mathbf{b}_m = \lambda_m \mathbf{X}^\top \mathbf{b}_m \iff \frac{1}{N} \mathbf{X}^\top \mathbf{X} \mathbf{c}_m = \lambda_m \mathbf{c}_m,$$

我们得到了一个新的特征向量/特征值方程: λ_m 仍然是特征值, 这证实了我们在第 4.5.3 节中的结果, 即 $\mathbf{X} \mathbf{X}^\top$ 的非零特征值等于 $\mathbf{X}^\top \mathbf{X}$ 的非零特征值。我们得到与 λ_m 相关联的矩阵 $\frac{1}{N} \mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{N \times N}$ 的特征向量为 $\mathbf{c}_m := \mathbf{X}^\top \mathbf{b}_m$ 。假设我们没有重复的数据点, 则该矩阵的秩为 N 且是可逆的。这也意味着 $\frac{1}{N} \mathbf{X}^\top \mathbf{X}$ 与数据协方差矩阵 \mathbf{S} 具有相同的 (非零) 特征值。但现在这是一个 $N \times N$ 的矩阵, 因此我们可以比原始的 $D \times D$ 数据协方差矩阵更有效地计算特征值和特征向量。既然我们已经得到了 $\frac{1}{N} \mathbf{X}^\top \mathbf{X}$ 的特征向量, 我们接下来将恢复原始的特征向量, 这在PCA中仍然需要。目前, 我们知道 $\frac{1}{N} \mathbf{X}^\top \mathbf{X}$ 的特征向量。如果我们用 \mathbf{X} 左乘我们的特征值/特征向量方程, 我们得到

(10.57)

$$\underbrace{\frac{1}{N} \mathbf{X} \mathbf{X}^\top}_{\mathbf{S}} \mathbf{X} \mathbf{c}_m = \lambda_m \mathbf{X} \mathbf{c}_m$$

并且我们再次恢复了数据协方差矩阵。这也意味着我们现在恢复了 $\mathbf{X} \mathbf{c}_m$ 作为 \mathbf{S} 的一个特征向量。

备注: 如果我们想应用我们在第 10.6 节中讨论的PCA算法, 我们需要将 \mathbf{S} 的特征向量 $\mathbf{X} \mathbf{c}_m$ 归一化, 使它们的范数为 1。

< 上一章节

下一章节 >

10.4 特征向量计算和低秩近似

10.6 实践中PCA的关键步骤



10.6 实践中PCA的关键步骤

接下来，我们将通过一个连续的例子来逐步介绍PCA的各个步骤，这些步骤总结在图10.11中。我们有一个二维数据集（图10.11(a)），我们想要使用PCA将其投影到一个一维子空间上。

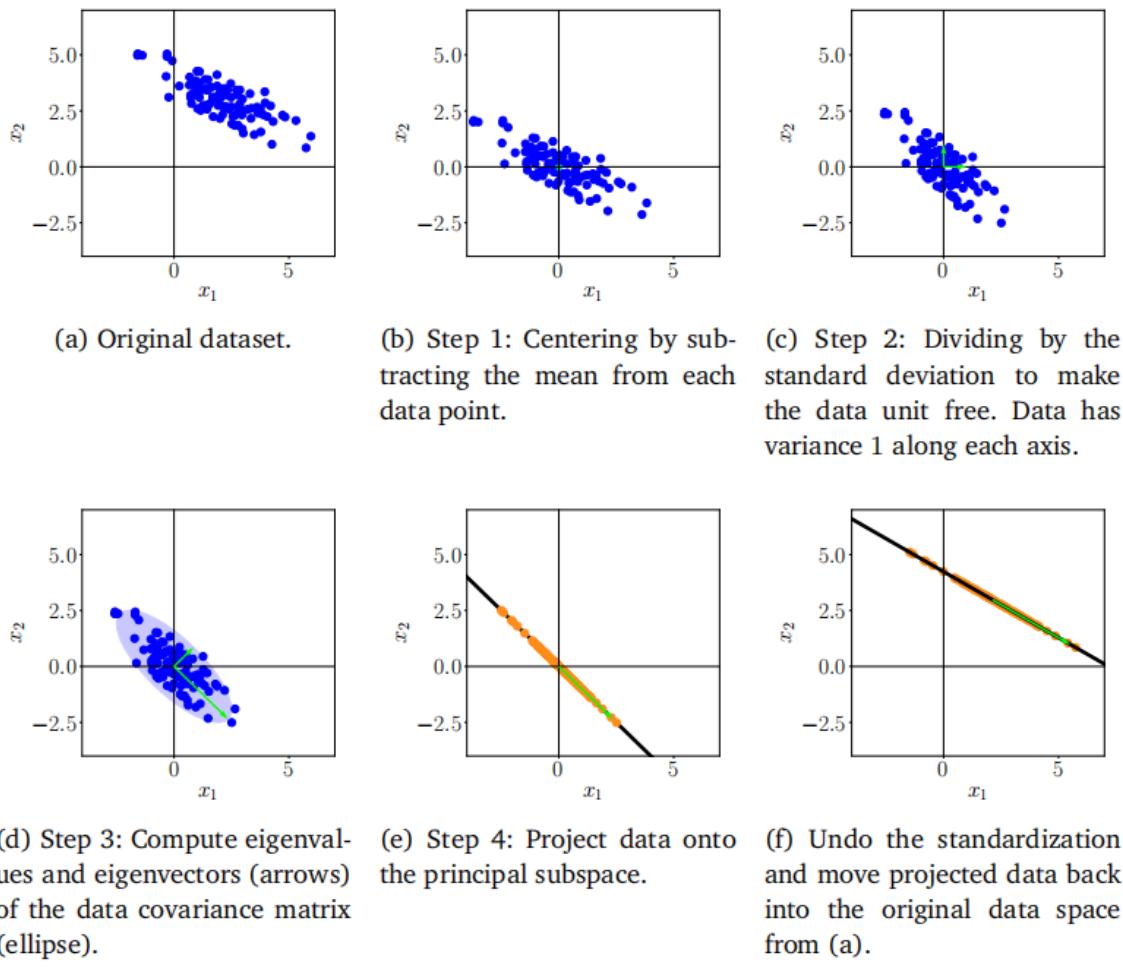


图10.11PCA.(a)原始数据集的步骤；(b)定中；(c)除以标准差；(d)特征分解；(e)投影；(f)映射回原始数据空间。

1. **均值归零**: 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这个均值来中心化数据。这确保了数据集的平均值为0（图10.11(b)）。均值归零虽然不是严格必要的，但它降低了数值问题的风险。
2. **标准化**: 对于每个维度 $d = 1, \dots, D$ ，我们将数据点除以数据集的标准差 σ_d 。现在数据是无单位的，并且每个轴上的方差为1，这在图10.11(c)中用两个箭头表



示。这一步完成了数据的标准化。

3. 协方差矩阵的特征分解：计算数据的协方差矩阵及其特征值和对应的特征向量。

由于协方差矩阵是对称的，根据谱定理（定理4.15），我们可以找到一组正交归一的特征向量基（ONB）。在图10.11(d)中，特征向量按其对应的特征值的大小进行缩放。较长的向量跨越了主子空间，我们将其表示为 \mathbf{U} 。数据的协方差矩阵由椭圆表示。

4. 投影：我们可以将任何数据点 $\mathbf{x}_* \in \mathbb{R}^D$ 投影到主子空间上：为了正确执行此操作，我们需要使用训练数据在第 d 维的均值 μ_d 和标准差 σ_d 来标准化 \mathbf{x}_* ，以便

(10.58)

$$\mathbf{x}_*^{(d)} \leftarrow \frac{\mathbf{x}_*^{(d)} - \mu_d}{\sigma_d}, \quad d = 1, \dots, D,$$

其中 $\mathbf{x}_*^{(d)}$ 是 \mathbf{x}_* 的第 d 个分量。我们得到的投影为 (10.59)

$$\tilde{\mathbf{x}}_* = \mathbf{B}\mathbf{B}^\top \mathbf{x}_*$$

其坐标为

$$\mathbf{z}_* = \mathbf{B}^\top \mathbf{x}_*$$

(10.60)

这是相对于主子空间基底的坐标。这里， \mathbf{B} 是一个矩阵，其列包含与数据协方差矩阵最大特征值相关联的特征向量。PCA返回的是坐标 (10.60)，而不是投影 \mathbf{x}_* 。

在标准化数据集后，(10.59) 仅给出了在标准化数据集上下文中的投影。为了获得原始数据空间（即标准化之前）中的投影，我们需要撤销标准化 (10.58)，并在添加均值之前乘以标准差，以便我们得到

$$\bar{\mathbf{x}}_*^{(d)} \leftarrow \mathbf{x}_*^{(d)} \sigma_d + \mu_d, \quad d = 1, \dots, D.$$

(10.61)

图10.11(f)展示了在原始数据空间中的投影。

示例 10.4 (MNIST 数字：重建)



在以下示例中，我们将PCA应用于MNIST数字数据集，该数据集包含0到9的手写数字示例共60,000个。每个数字都是一张大小为 28×28 的图像，即包含784个像素，因此我们可以将该数据集中的每张图像解释为向量 $x \in \mathbb{R}^{784}$ 。这些数字的一些示例如图10.3所示。

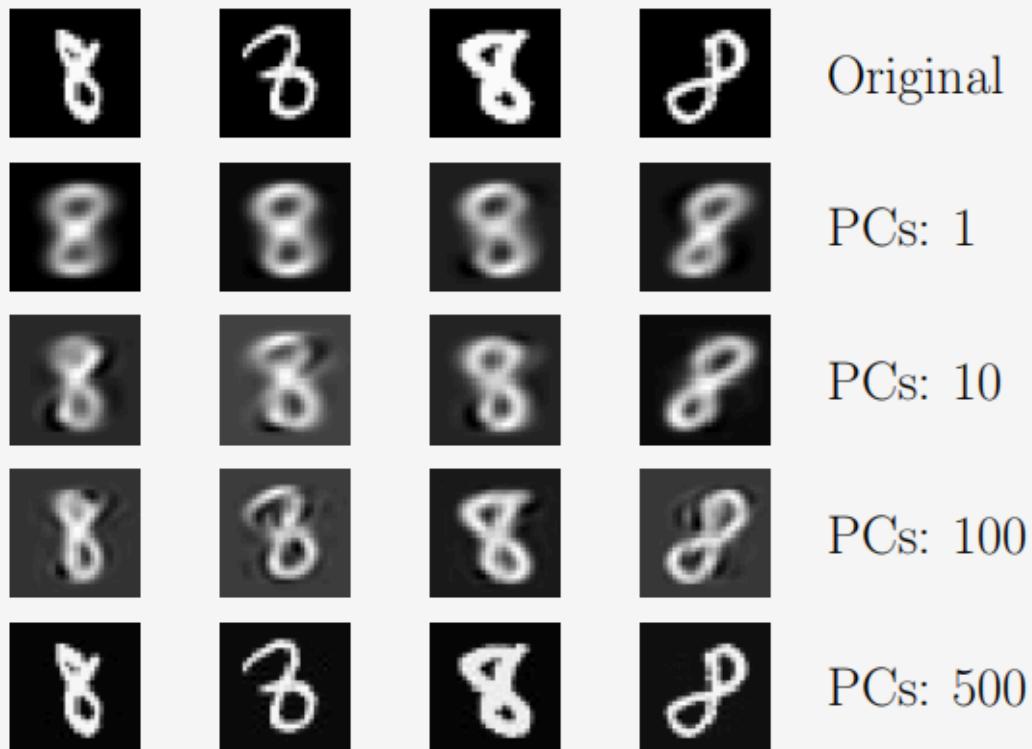


图10.12增加主成分数量对重建的影响

为了说明目的，我们将PCA应用于MNIST数字的一个子集，并专注于数字“8”。我们使用了5,389张数字“8”的训练图像，并根据本章中的详细说明确定了主子空间。然后，我们使用学习到的投影矩阵来重建一组测试图像，如图10.12所示。图10.12的第一行显示了一组来自测试集的四个原始数字。接下来的几行分别展示了当使用维度为1、10、100和500的主子空间时，这些数字的精确重建结果。我们可以看到，即使使用一维主子空间，我们也能得到原始数字的半程像样的重建，但图像模糊且通用。随着主成分（PCs）数量的增加，重建图像变得更加清晰，并保留了更多细节。使用500个主成分时，我们几乎可以完美重建图像。如果我们选择784个主成分，我们将能够无压缩损失地恢复出精确的数字。

图10.13显示了平均平方重建误差，其公式为

$$(10.62)$$



$$\frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \bar{\mathbf{x}}_n\|^2 = \sum_{i=M+1}^D \lambda_i ,$$

该误差是主成分数量 M 的函数。我们可以看到，主成分的重要性迅速下降，添加更多主成分只能获得微不足道的增益。这与我们在图10.5中的观察结果完全一致，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个主成分，我们基本上可以完全重建包含数字“8”的训练数据（数据集中一些边界周围的像素没有变化，因为它们始终是黑色的）。

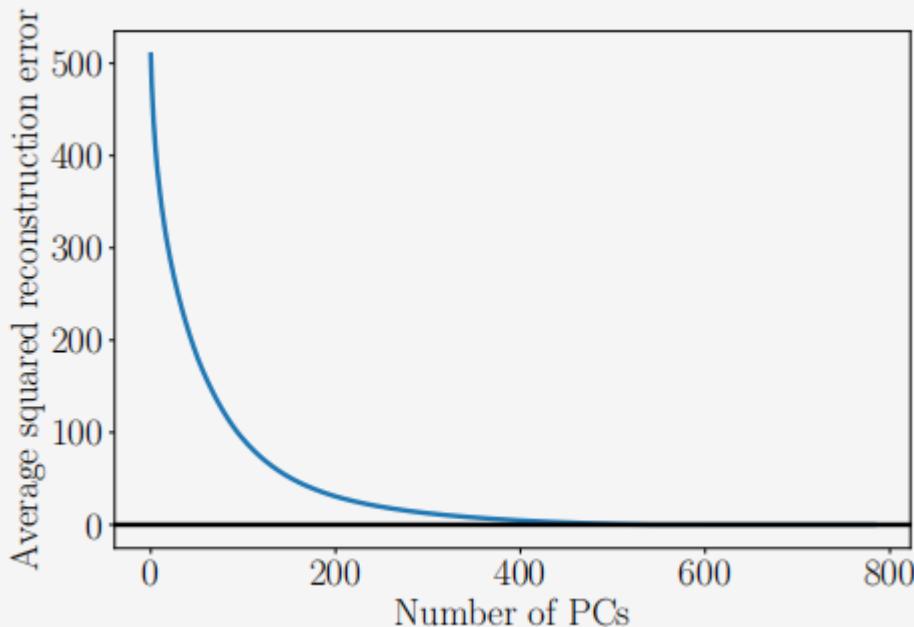


图10.13 平均平方重构误差作为主成分数量的函数。平均平方重建误差是主子空间的正交补中的特征值的和。

< 上一章节

下一章节 >

10.5 高维PCA

10.7 潜在变量视角



10.7 潜在变量视角

在前面的章节中，我们从最大方差和投影视角出发，推导了PCA，而没有引入任何概率模型的概念。一方面，这种方法可能很有吸引力，因为它使我们能够避开与概率论相关的所有数学难题；但另一方面，概率模型会为我们提供更多灵活性和有用的见解。更具体地说，概率模型会：

- 附带一个似然函数，我们可以明确地处理含噪声的观测值（这是我们之前甚至都没有讨论过的）
- 允许我们通过边缘似然度进行贝叶斯模型比较，如第8.6节所述
- 将PCA视为生成模型，使我们能够模拟新数据
- 允许我们直接联系到相关算法
- 通过应用贝叶斯定理处理随机缺失的数据维度
- 给出新数据点的新颖性概念
- 为我们提供扩展模型的原则性方法，例如扩展到PCA模型的混合形式
- 将我们在前面章节中推导的PCA作为特殊情况
- 通过边缘化模型参数实现完全贝叶斯处理

通过引入连续值的潜在变量 $z \in \mathbb{R}^M$ ，可以将PCA表述为概率潜在变量模型。

Tipping和Bishop (1999) 提出了这种潜在变量模型，即概率PCA (PPCA)。PPCA解决了上述大部分问题，而我们通过最大化投影空间中的方差或最小化重建误差所获得的PCA解，是在无噪声设置下的最大似然估计的特殊情况。

10.7.1 生成过程和概率模型

在概率PCA (PPCA) 中，我们明确写出了线性降维的概率模型。为此，我们假设存在一个连续的潜在变量 $z \in \mathbb{R}^M$ ，它遵循标准正态先验 $p(z) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ ，并且潜在变量与观测到的数据 x 之间存在线性关系，其中

(10.63)

$$\mathbf{x} = \mathbf{B}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \in \mathbb{R}^D,$$

其中， $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ 是高斯观测噪声， $\mathbf{B} \in \mathbb{R}^{D \times M}$ 和 $\boldsymbol{\mu} \in \mathbb{R}^D$ 描述了从潜在变量到观测变量的线性/仿射映射。因此，PPCA通过以下方式将潜在变量和观测变量联系起来：

$$p(\mathbf{x} | \mathbf{z}, \mathbf{B}, \boldsymbol{\mu}, \sigma^2) = \mathcal{N}(\mathbf{x} | \mathbf{B}\mathbf{z} + \boldsymbol{\mu}, \sigma^2 \mathbf{I}).$$

(10.64)

总体而言，PPCA诱导了以下生成过程：

(10.65)

$$\begin{aligned} z_n &\sim \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I}) \\ x_n | z_n &\sim \mathcal{N}(\mathbf{x} | \mathbf{B}\mathbf{z}_n + \boldsymbol{\mu}, \sigma^2 \mathbf{I}) \end{aligned}$$

(10.66)

为了在给定模型参数的情况下生成一个典型的数据点，我们遵循一个祖先采样方案：首先，我们从 $p(\mathbf{z})$ 中采样一个潜在变量 z_n 。然后，我们在(10.64)中使用 \mathbf{z}_n 来根据采样得到的 z_n 条件采样一个数据点，即 $x_n \sim p(\mathbf{x} | z_n, \mathbf{B}, \boldsymbol{\mu}, \sigma^2)$ 。

这个生成过程允许我们写下概率模型（即所有随机变量的联合分布；参见第8.4节）为

(10.67)

$$p(\mathbf{x}, \mathbf{z} | \mathbf{B}, \boldsymbol{\mu}, \sigma^2) = p(\mathbf{x} | \mathbf{z}, \mathbf{B}, \boldsymbol{\mu}, \sigma^2)p(\mathbf{z}),$$

这立即导致了使用第8.5节结果的图10.14中的图形模型。

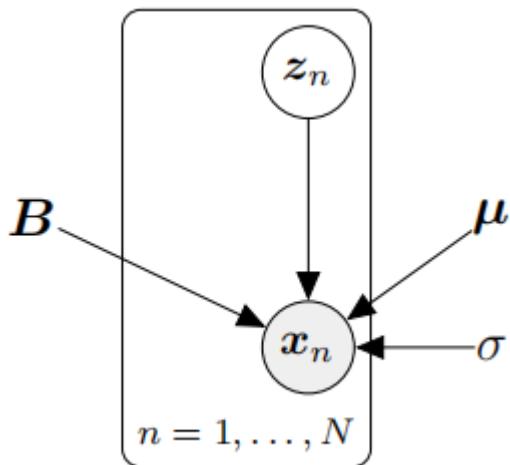


图10.14概率PCA的图形模型。观测值 x_n 明确地依赖于相应的潜在变量 $z_n \sim N$

备注。注意连接潜在变量 z 和观测数据 x 的箭头的方向：箭头指向从 z 到 x ，这意味着PPCA模型假定了高维观测 x 的低维潜在原因 z 。最后，根据一些观察结果，我们显然对发现 z 很感兴趣。为了达到这个目的，我们将应用贝叶斯推理隐式地“反转”箭头，并从观察到潜在变量。

示例 10.5（使用潜在变量生成新数据）

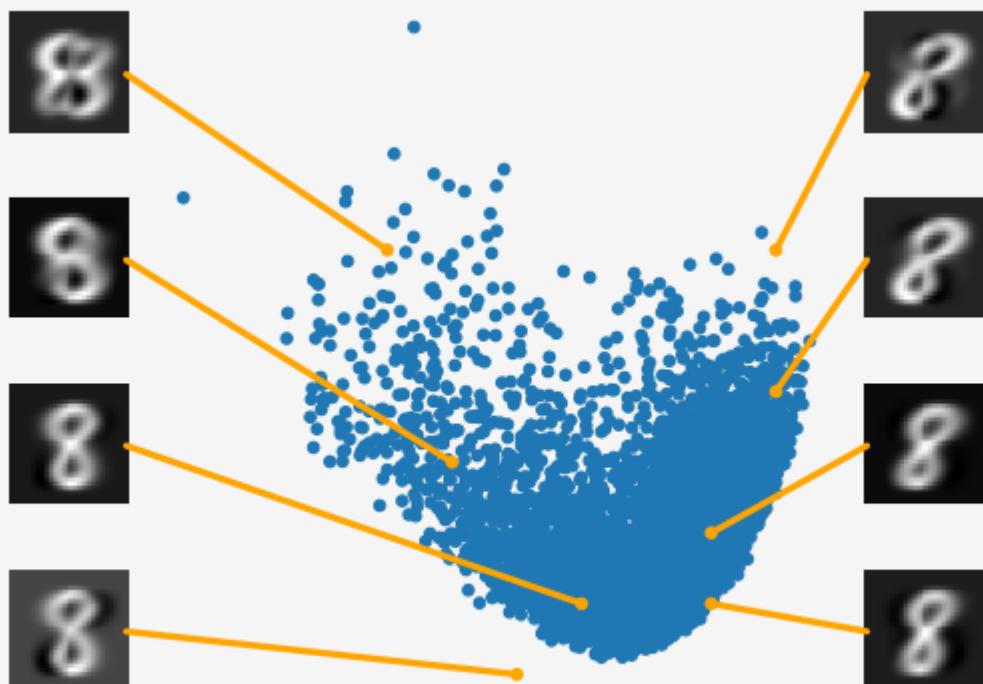


图10.15生成新的MNIST数字。潜在变量 z 可以用来生成新的数据 $\tilde{x} = Bz$ 。我们离训练数据越近，生成的数据就越真实。

图10.15展示了当使用二维主成分子空间时，PCA找到的MNIST数字“8”的潜在坐标（蓝色点）。我们可以在这个潜在空间中查询任何向量 z_* ，并生成一个图像 $\tilde{x}_* = Bz_*$ ，该图像类似于数字“8”。我们展示了八个这样的生成图像及其对应的潜在空间表示。根据我们在潜在空间中查询的位置不同，生成的图像看起来会有所不同（形状、旋转、大小等）。如果我们查询的位置远离训练数据，我们会看到越来越多的伪影，例如左上角和右上角的数字。请注意，这些生成图像的内在维度只有两个。

10.7.2 似然函数和联合分布

利用第6章的结果，我们通过积分出潜在变量 z （参见第8.4.3节）来得到这个概率模型的似然函数，即

$$\begin{aligned} p(\mathbf{x} \mid \mathbf{B}, \boldsymbol{\mu}, \sigma^2) &= \int p(\mathbf{x} \mid z, \mathbf{B}, \boldsymbol{\mu}, \sigma^2) p(z) dz \\ &= \int \mathcal{N}(x \mid Bz + \boldsymbol{\mu}, \sigma^2 \mathbf{I}) \mathcal{N}(z \mid \mathbf{0}, \mathbf{I}) dz. \end{aligned}$$

(10.68a)

(10.68b)

从第6.5节我们知道，这个积分的解是一个高斯分布，其均值为

$$\mathbb{E}_x[\mathbf{x}] = \mathbb{E}_z[Bz + \boldsymbol{\mu}] + \mathbb{E}_{\epsilon}[\boldsymbol{\epsilon}] = \boldsymbol{\mu}$$

(10.69)

协方差矩阵为

(10.70a)

$$\begin{aligned} \mathbb{V}[\mathbf{x}] &= \mathbb{V}_z[Bz + \boldsymbol{\mu}] + \mathbb{V}_{\epsilon}[\boldsymbol{\epsilon}] = \mathbb{V}_z[Bz] + \sigma^2 \mathbf{I} \\ &= B \mathbb{V}_z[z] B^T + \sigma^2 \mathbf{I} = \mathbf{B} \mathbf{B}^T + \sigma^2 \mathbf{I}. \end{aligned}$$

(10.70b)

(10.68b)中的似然函数可用于模型参数的最大似然估计或MAP估计。

备注: 我们不能使用(10.64)中的条件分布进行最大似然估计, 因为它仍然依赖于潜在变量。我们用于最大似然 (或MAP) 估计的似然函数只应是数据 \mathbf{x} 和模型参数的函数, 而不应依赖于潜在变量。

◇

从第6.5节我们知道, 高斯随机变量 z 及其线性/仿射变换 $\mathbf{x} = \mathbf{B}z$ 是联合高斯分布的。我们已经知道边缘分布 $p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I})$ 和 $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \mathbf{B}\mathbf{B}^\top + \sigma^2 \mathbf{I})$ 。缺失的互协方差为

$$\text{Cov}[\mathbf{x}, z] = \text{Cov}_z[Bz + \boldsymbol{\mu}] = \mathbf{B} \text{Cov}_z[z, z] = \mathbf{B}.$$

(10.71)

因此, PPCA的概率模型, 即潜在变量和观测随机变量的联合分布明确给出为

$$p(\mathbf{x}, \mathbf{z} | \mathbf{B}, \boldsymbol{\mu}, \sigma^2) = \mathcal{N} \left(\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} \mid \begin{bmatrix} \boldsymbol{\mu} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{B}\mathbf{B}^\top + \sigma^2 \mathbf{I} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{I} \end{bmatrix} \right),$$

(10.72)

其中均值向量的长度为 $D + M$, 协方差矩阵的大小为 $(D + M) \times (D + M)$ 。

10.7.3 后验分布

(10.72)中的联合高斯分布 $p(\mathbf{x}, \mathbf{z} | \mathbf{B}, \boldsymbol{\mu}, \sigma^2)$ 允许我们立即通过应用第6.5.1节中高斯条件分布的规则来确定后验分布 $p(\mathbf{z} | \mathbf{x})$ 。给定观测值 \mathbf{x} 时, 潜在变量的后验分布为

$$\begin{aligned} p(\mathbf{z} | \mathbf{x}) &= \mathcal{N}(\mathbf{z} | \mathbf{m}, \mathbf{C}), \\ \mathbf{m} &= \mathbf{B}^\top (\mathbf{B}\mathbf{B}^\top + \sigma^2 \mathbf{I})^{-1} (\mathbf{x} - \boldsymbol{\mu}), \\ \mathbf{C} &= \mathbf{I} - \mathbf{B}^\top (\mathbf{B}\mathbf{B}^\top + \sigma^2 \mathbf{I})^{-1} \mathbf{B}. \end{aligned}$$

(10.73) (10.74) (10.75)

注意, 后验协方差并不依赖于观测数据 \mathbf{x} 。对于数据空间中的新观测值 \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} 允许我们评估嵌入的置信度。协方差矩阵 \mathbf{C} 的行列式较小 (测量体积) 意味着潜在嵌入 \mathbf{z}_* 相当确定。如果我们得到的后验分布 $p(\mathbf{z}_* | \mathbf{x}_*)$ 方差很大, 那么我们可能遇到了一个异常值。然而, 我们

可以探索这个后验分布，以了解在这个后验下哪些其他数据点 \mathbf{x} 是合理的。为此，我们利用**PPCA**背后的生成过程，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的后验分布：

1. 从潜在变量的后验分布(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 $\tilde{x}_* \sim p(\mathbf{x} | z_*, \mathbf{B}, \boldsymbol{\mu}, \sigma^2)$ 。

如果我们多次重复这个过程，就可以探索潜在变量 z_* 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数据，这些数据在后验分布下是合理的。

< 上一章节

下一章节 >

10.6 实践中PCA的关键步骤

10.8 拓展阅读



10.8 拓展阅读

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也可以从不同角度进行解释。让我们回顾一下我们所做的: 我们采用了高维数据 $\mathbf{x} \in \mathbb{R}^D$, 并使用矩阵 \mathbf{B}^\top 来找到一个低维表示 $\mathbf{z} \in \mathbb{R}^M$ 。矩阵 \mathbf{B} 的列是数据协方差矩阵 \mathbf{S} 与最大特征值相关联的特征向量。一旦我们有了低维表示 \mathbf{z} , 我们就可以通过 $\mathbf{x} \approx \tilde{\mathbf{x}} = \mathbf{B}\mathbf{z} = \mathbf{B}\mathbf{B}^\top\mathbf{x} \in \mathbb{R}^D$ 得到其高维版本 (在原始数据空间中), 其中 $\mathbf{B}\mathbf{B}^\top$ 是一个投影矩阵。

我们还可以将PCA视为如图10.16所示的线性自编码器。自编码器将数据 $\mathbf{x}_n \in \mathbb{R}^D$ 编码为代码 $\mathbf{z}_n \in \mathbb{R}^M$, 并将其解码为与 \mathbf{x}_n 相似的 $\tilde{\mathbf{x}}_n$ 。从数据到代码的映射称为编码器, 而从代码返回原始数据空间的映射称为解码器。如果我们考虑线性映射, 其中代码由 $\mathbf{z}_n = \mathbf{B}^\top\mathbf{x}_n \in \mathbb{R}^M$ 给出, 并且我们关注于最小化数据 \mathbf{x}_n 与其重构 $\tilde{\mathbf{x}}_n = \mathbf{B}\mathbf{z}_n, n = 1, \dots, N$ 之间的平均平方误差, 我们得到

(10.76)

$$\frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2 = \frac{1}{N} \sum_{n=1}^N \left\| \mathbf{x}_n - \mathbf{B}\mathbf{B}^\top\mathbf{x}_n \right\|^2.$$

这意味着我们最终得到了与第10.3节中讨论的(10.29)相同的目标函数, 因此当我们最小化平方自编码损失时, 我们得到了PCA的解。如果我们用非线性映射替换PCA的线性映射, 我们得到一个非线性自编码器。这种情况的一个突出例子是深度自编码器, 其中线性函数被深度神经网络所替代。在这种情况下, 编码器也被称为识别网络或推理网络, 而解码器也被称为生成器。

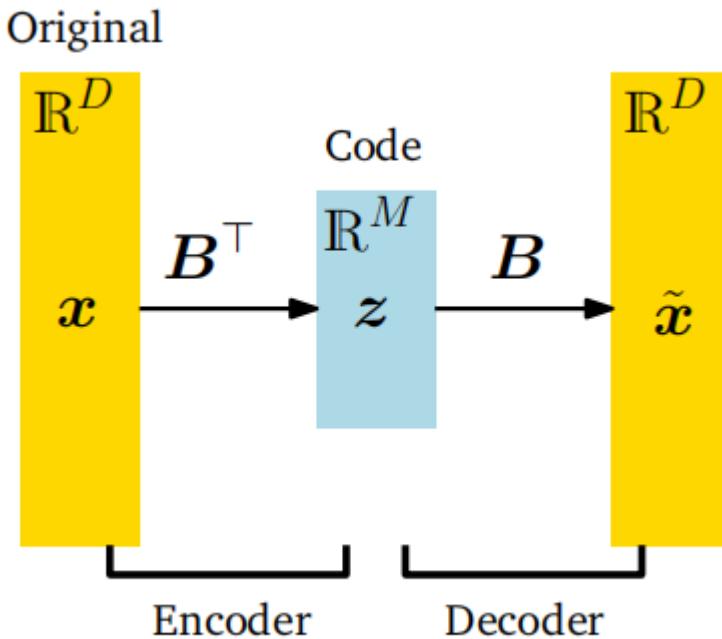


图10.16 PCA可以看作是一个线性的自动编码器。它将高维数据 x 编码为低维表示（代码） $z \in \mathbb{R}^M$ ，并使用解码器对 z 进行解码。解码的向量 \tilde{x} 是原始数据 x 在 M 维主子空间上的正交投影。

PCA的另一种解释与信息论有关。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，我们不会得到完全相同的数据点，而是其稍微失真或带有噪声的版本。这意味着我们的压缩是“有损”的。直观地说，我们希望最大化原始数据与低维代码之间的相关性。更正式地说，这与互信息有关。然后，我们可以通过最大化互信息（信息论中的一个核心概念，MacKay, 2003）来得到我们在第10.3节中讨论的PCA的相同解。

在关于PPCA的讨论中，我们假设了模型的参数，即 B 、 μ 和似然参数 σ^2 是已知的。Tipping和Bishop (1999) 描述了如何在PPCA设置中推导出这些参数的最大似然估计（请注意，我们在本章中使用了不同的符号）。当将 D 维数据投影到 M 维子空间时，最大似然参数为

(10.77)



$$\begin{aligned}\mu_{\text{ML}} &= \frac{1}{N} \sum_{n=1}^N x_n, \\ B_{\text{ML}} &= T(\Lambda - \sigma^2 I)^{\frac{1}{2}} R, \\ \sigma_{\text{ML}}^2 &= \frac{1}{D-M} \sum_{j=M+1}^D \lambda_j,\end{aligned}$$

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，矩阵 $\Lambda - \sigma^2 I$ 中， $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_M) \in \mathbb{R}^{M \times M}$ 是一个对角矩阵，其对角线上的元素是与主成分相对应的特征值（见(10.78)），而 $R \in \mathbb{R}^{M \times M}$ 保证是任意正交矩阵。最大似然解 B_{ML} 在任意正交变换下是唯一的，例如，我们可以将 B_{ML} 与任意旋转矩阵 R 右乘，所以 (10.78) 本质上是数据协方差矩阵的奇异值分解（见第4.5节）。Tipping 和 Bishop (1999) 给出了证明的概要。

(10.77) 中给出的 μ 的最大似然估计是数据的样本均值。(10.79) 中给出的观测噪声方差 σ^2 的最大似然估计是在主分子空间的正交补空间中的平均方差，即我们不能用前 M 个主成分捕获的平均剩余方差被视为观测噪声。

在无噪声极限下，即 $\sigma \rightarrow 0$ 时，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（见第4.4节），即存在 S 的特征向量矩阵 T ，使得

$$S = T \Lambda T^{-1}.$$

在 PPCA 模型中，数据协方差矩阵是高斯似然 $p(\mathbf{x}_\perp | \mathbf{B}, \boldsymbol{\mu}, \sigma^2)$ 的协方差矩阵，即 $\mathbf{B}\mathbf{B}^\top + \sigma^2 \mathbf{I}$ ，见(10.70b)。对于 $\sigma \rightarrow 0$ ，我们得到 $\hat{\mathbf{B}}\mathbf{B}^\top$ ，因此这个数据协方差必须等于 PCA 的数据协方差（以及其在(10.80) 中给出的分解），从而

$$\text{Cov}[\mathcal{X}] = T \Lambda T^{-1} = \mathbf{B}\mathbf{B}^\top \iff \mathbf{B} = T \Lambda^{\frac{1}{2}} \mathbf{R},$$

(10.81)

即，我们在(10.78) 中获得了 $\sigma = 0$ 时的最大似然估计。从(10.78) 和(10.80) 可以看出，(P)PCA 对数据协方差矩阵进行了分解。

在数据流设置中，数据是顺序到达的，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度（即代码的长度，或我们将数据投影到的低维子空间的维度），Gavish 和 Donoho (2014) 提出了一种启发式方法：如果我们能够估计数据的噪

声方差 σ^2 , 则应丢弃所有小于 $\frac{4\sigma\sqrt{D}}{\sqrt{3}}$ 的奇异值。或者, 我们可以使用(嵌套)交叉验证(第8.6.1节)或贝叶斯模型选择标准(第8.6.2节讨论)来确定数据内在维度的一个良好估计(Minka, 2001b)。

类似于我们在第9章关于线性回归的讨论, 我们可以在模型的参数上放置一个先验分布, 并将其积分出来。这样做可以(a)避免参数的点估计以及这些点估计带来的问题(见第8.6节), (b)允许自动选择潜在空间的适当维度 M 。在Bishop(1999)提出的贝叶斯PCA中, 模型参数上放置了一个先验 $p(\boldsymbol{\mu}, \mathbf{B}, \sigma^2)$ 。生成过程允许我们积分出模型参数而不是对其进行条件化, 这解决了过拟合问题。由于这种积分在解析上是不可行的, Bishop(1999)建议使用近似推理方法, 如MCMC或变分推理。关于这些近似推理技术的更多细节, 请参考Gilks et al.(1996)和Blei et al.(2017)的工作。

在PPCA中, 我们考虑了线性模型 $p(x_n | z_n) = \mathcal{N}(x_n | \mathbf{B}z_n + \boldsymbol{\mu}, \sigma^2 \mathbf{I})$, 其中先验 $p(z_n) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, 所有观测维度都受到相同数量的噪声影响。如果我们允许每个观测维度 d 具有不同的方差 σ_d^2 , 则得到因子分析(FA)(Spearman, 1904; Bartholomew et al., 2011)。这意味着FA比PPCA在似然上提供了更多的灵活性, 但仍然迫使数据由模型参数 $\mathbf{B}, \boldsymbol{\mu}$ 来解释。然而, FA不再允许封闭形式的最大似然解, 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。在PPCA中, 所有驻点都是全局最优解, 但在FA中则不然。与PPCA相比, 如果我们缩放数据, FA不会改变, 但如果我们将数据旋转, FA会返回不同的解。

与PCA紧密相关的另一种算法是独立成分分析(ICA, Hyvarinen et al., 2001)。再次从潜在变量的角度出发 $p(x_n | z_n) = \mathcal{N}(x_n | \mathbf{B}z_n + \boldsymbol{\mu}, \sigma^2 \mathbf{I})$, 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。想象一下你身处一个繁忙的火车站, 周围有很多人说话。你的耳朵充当麦克风的作用, 它们会线性地混合火车站中的不同语音信号。盲源分离的目标是识别出混合信号中的组成部分。正如之前在讨论PPCA的最大似然估计时所提到的, 原始的PCA解决方案对任何旋转都是不变的。因此, PCA可以识别出信号所在的最佳低维子空间, 但无法识别信号本身(Murphy, 2012)。ICA通过修改潜在源上的先验分布 $p(\mathbf{z})$ 来解决这个问题, 要求非高斯先验 $p(\mathbf{z})$ 。关于ICA的更多细节, 请参考Hyvarinen et al. (2001) 和Murphy (2012) 的著作。

PCA、因子分析和ICA是使用线性模型进行降维的三个例子。Cunningham和Ghahramani (2015) 对线性降维进行了更广泛的综述。

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 远大于数据点数量 N 时如何进行PCA。通过利用PCA可以通过计算

(许多) 内积来执行的见解, 这个想法可以通过考虑无限维特征而被推向极端。核技巧是核PCA的基础, 它允许我们隐式地计算无限维特征之间的内积 (Schölkopf et al., 1998; Schölkopf和Smola, 2002)。

有一些从PCA衍生出的非线性降维技术 (Burges, 2010提供了一个很好的概述)。我们在本节前面讨论的PCA的自编码器视角可以将其呈现为深度自编码器的一个特例。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, 这些神经网络本身是非线性映射。如果我们将这些神经网络中的激活函数设置为恒等函数, 则该模型变得与PCA等价。非线性降维的另一种方法是Lawrence (2005) 提出的高斯过程潜在变量模型 (GP-LVM)。GP-LVM从我们用于推导PPCA的潜在变量视角出发, 将潜在变量 z 与观测值 x 之间的线性关系替换为高斯过程 (GP)。与我们在PPCA中估计映射参数不同, GP-LVM对模型参数进行边缘化, 并对潜在变量 z 进行点估计。与贝叶斯PCA类似, Titsias和Lawrence (2010) 提出的贝叶斯GP-LVM在潜在变量 z 上保持一个分布, 并使用近似推理来将其积分出来。

< 上一章节

下一章节 >

10.7 潜在变量视角

习题

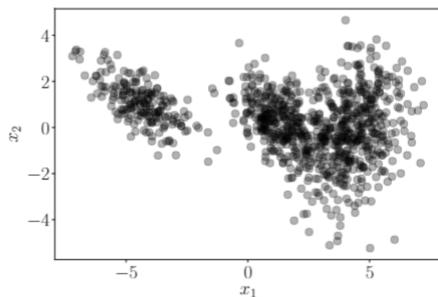


404 - Not found



第十一章 密度估计和 Gauss 混合模型

在前面的章节中，我们已经介绍了机器学习中的两个基本问题：回归（第9章）和降维（第10章）。在本章中，我们将探讨机器学习的第三大支柱：密度估计。在这个过程中，我们将引入一些重要的概念，例如期望最大化（EM）算法，以及从潜在变量的角度看待使用混合模型进行密度估计。当我们将机器学习应用于数据时，我们通常希望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；图11.1给出了一个示例。然而，如果数据集非常大，或者我们对表示数据的特征感兴趣，那么这种方法可能就不太有用了。在密度估计中，我们使用参数族中的一个密度函数（例如高斯分布或贝塔分布）来紧凑地表示数据。例如，我们可能会寻找数据集的均值和方差，以便使用高斯分布来紧凑地表示数据。均值和方差可以使用我们在8.3节中讨论过的工具来找到：最大似然估计或最大后验估计。然后，我们可以使用这个高斯分布的均值和方差来表示数据背后的分布，也就是说，如果我们从这个分布中采样，我们会认为这个数据集是这个分布的一个典型实现。



在实践中，高斯分布（或者到目前为止我们遇到的所有其他分布）的建模能力是有限的。例如，用高斯分布来近似图11.1中数据的密度将是一个很差的近似。接下来，我们将研究一类更具表达能力的分布，我们可以用它们来进行密度估计：混合模型。

混合模型 混合模型可以通过K个简单（基础）分布的凸组合来描述一个分布 $p(x)$ ：



$$p(\mathbf{x}) \leq \sum_{k=1}^K \pi_k p_k(\mathbf{x}) \quad (11.1)$$

$$0 \leq \pi_k \leq 1, \quad \sum_{k=1}^K \pi_k = 1, \quad (11.2)$$

其中，分量 p_k 是基本分布族的成员，例如高斯分布、伯努利分布或伽马分布，而 π_k 是混合权重。混合模型比相应的基础分布更具表达能力，因为它们允许对多峰数据进行表示，也就是说，它们可以描述具有多个“簇”的数据集，如图11.1中的示例。

混合权重 我们将重点关注高斯混合模型（GMM），其中基本分布是高斯分布。对于给定的数据集，我们的目标是最大化模型参数的似然，以训练GMM。为此，我们将使用第5章、第6章和7.2节中的结果。然而，与我们之前讨论过的其他应用（线性回归或主成分分析）不同，我们不会找到一个封闭形式的最大似然解。相反，我们将得到一组相互依赖的联立方程，我们只能迭代地求解它们。

< 上一章节

第十章 降维和主成分分析

下一章节 >

第十二章 分类和支持向量机



11.1 Gauss 混合模型

高斯混合模型是一种密度模型，我们将有限数量的 K 个高斯分布 $N(x|\mu_k, \Sigma_k)$ 组合起来，使得：

$$p(\mathbf{x} | \theta) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (11.3)$$

$$0 \leq \pi_k \leq 1, \quad \sum_{k=1}^K \pi_k = 1 \quad (11.4)$$

其中，我们将 $\theta := \mu_k, \Sigma_k : k = 1, \dots, K$ 定义为模型所有参数的集合。这种高斯分布的凸组合为我们建模复杂密度提供了比简单高斯分布（当 $K = 1$ 时，我们可以从（11.3）中得到简单高斯分布）显著更多的灵活性。图11.2给出了一个示例，展示了加权分量和混合密度，其表达式为：

$$p(x | \theta) = 0.5\mathcal{N}\left(x | -2, \frac{1}{2}\right) + 0.2\mathcal{N}(x | 1, 2) + 0.3\mathcal{N}(x | 4, 1) \quad (11.5)$$

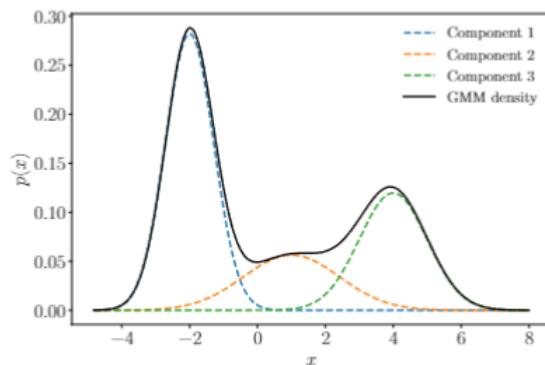
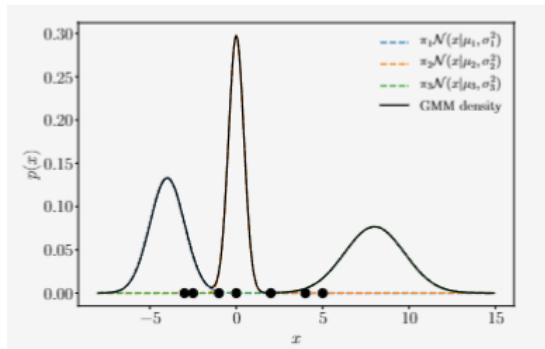


图11.2 高斯混合模型。高斯混合分布（黑色）由高斯分布的凸组合组成，比任何单个分量都更具表达能力。虚线表示加权的高斯分量。



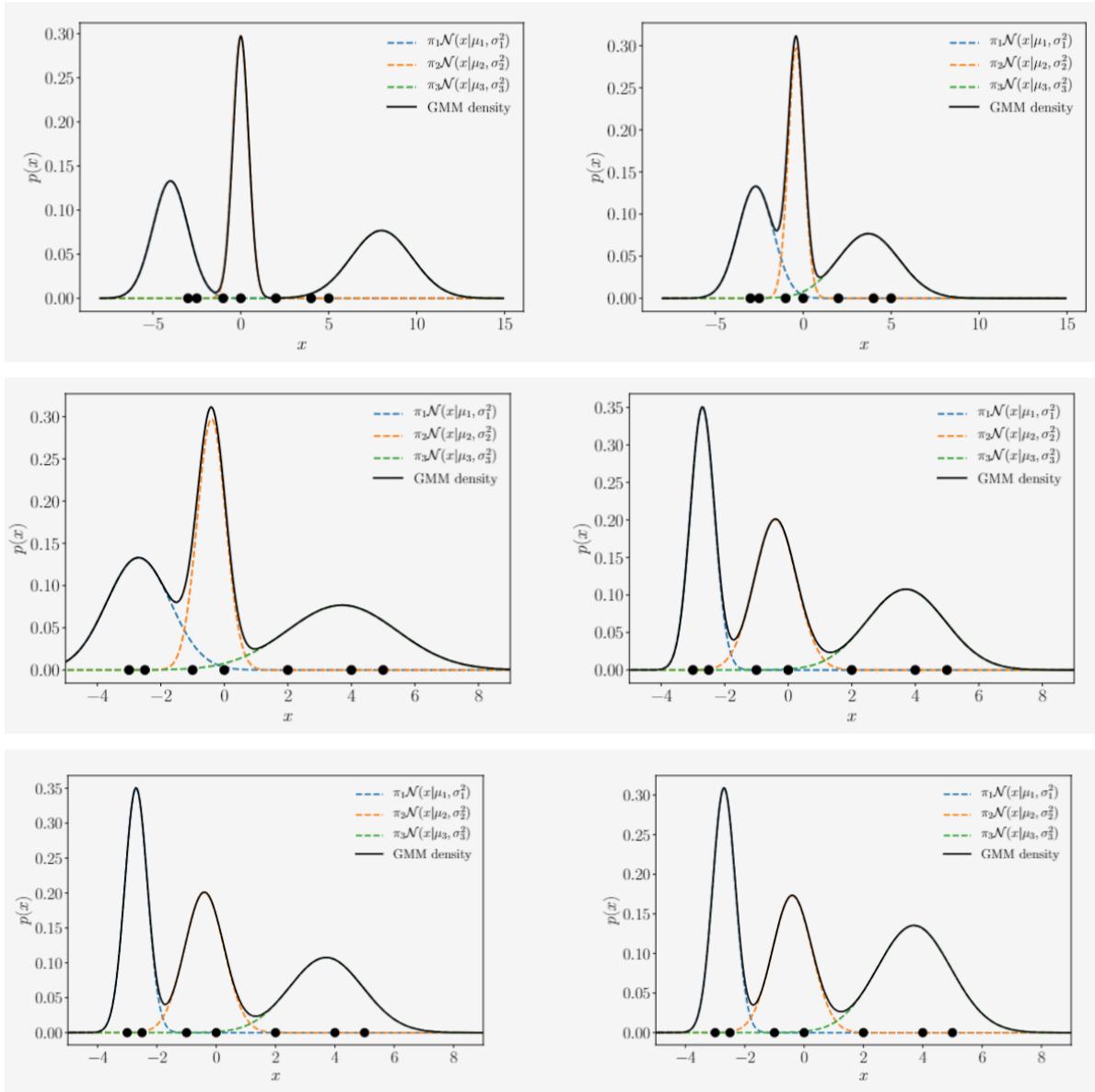
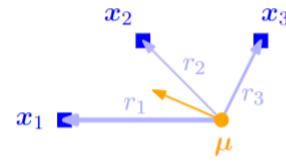


下一章节 >

11.2 通过最大似然估计学习参数



11.2 通过最大似然估计学习参数



$$p_1(x) = \mathcal{N}(x | -4, 1) \quad (11.6)$$

$$p_2(x) = \mathcal{N}(x | 0, 0.2) \quad (11.7)$$

$$p_3(x) = \mathcal{N}(x | 8, 3) \quad (11.8)$$

$$p(\mathcal{X} \mid \boldsymbol{\theta}) = \prod_{n=1}^N p(\mathbf{x}_n \mid \boldsymbol{\theta}), \quad p(\mathbf{x}_n \mid \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (11.9)$$

$$\log p(\mathcal{X} \mid \boldsymbol{\theta}) = \sum_{n=1}^N \log p(\mathbf{x}_n \mid \boldsymbol{\theta}) = \underbrace{\sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{=: \mathcal{L}} \quad (11.10)$$

$$\log \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{D}{2} \log(2\pi) - \frac{1}{2} \log \det(\boldsymbol{\Sigma}) - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}).$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = \mathbf{0}^\top \iff \sum_{n=1}^N \frac{\partial \log p(\mathbf{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \mathbf{0}^\top, \quad (11.12)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = \mathbf{0} \iff \sum_{n=1}^N \frac{\partial \log p(\mathbf{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} = \mathbf{0}, \quad (11.13)$$

$$\frac{\partial \mathcal{L}}{\partial \pi_k} = 0 \iff \sum_{n=1}^N \frac{\partial \log p(\mathbf{x}_n \mid \boldsymbol{\theta})}{\partial \pi_k} = 0. \quad (11.14)$$

$$\frac{\partial \log p(\mathbf{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{p(\mathbf{x}_n \mid \boldsymbol{\theta})} \frac{\partial p(\mathbf{x}_n \mid \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \quad (11.15)$$

$$\frac{1}{p(\mathbf{x}_n \mid \boldsymbol{\theta})} = \frac{1}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \quad (11.16)$$

11.2.1

$$r_{n,k} := \frac{\pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \quad (11.17)$$

$$p(\mathbf{x}_n \mid \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \pi \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (11.18)$$

$$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 0.057 & 0.943 & 0.0 \\ 0.001 & 0.999 & 0.0 \\ 0.0 & 0.066 & 0.934 \\ 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \in \mathbb{R}^{N \times K}. \quad (11.19)$$

11.2.2 更新均值向量



$$\boldsymbol{\mu}_k^{\text{new}} = \frac{\sum_{n=1}^N r_{n,k} \mathbf{x}_n}{\sum_{n=1}^N r_{n,k}}, \quad (11.20)$$

$$\frac{\partial p(\mathbf{x}_n | \theta)}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^K \pi_j \frac{\partial \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\partial \boldsymbol{\mu}_k} = \pi_k \frac{\partial \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\partial \boldsymbol{\mu}_k} \quad (11.21a)$$

$$= \pi_k (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (11.21b)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = \sum_{n=1}^N \frac{\partial \log p(\mathbf{x}_n | \theta)}{\partial \boldsymbol{\mu}_k} = \sum_{n=1}^N \frac{1}{p(\mathbf{x}_n | \theta)} \frac{\partial p(\mathbf{x}_n | \theta)}{\partial \boldsymbol{\mu}_k} \quad (11.22a)$$

$$= \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{=r_{n,k}} \quad (11.22b)$$

$$= \sum_{n=1}^N r_{n,k} (\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}. \quad (11.22c)$$

$$\begin{aligned} \sum_{n=1}^N r_{n,k} \mathbf{x}_n &= \sum_{n=1}^N r_{n,k} \boldsymbol{\mu}_k^{\text{new}} \\ &\Updownarrow \\ \boldsymbol{\mu}_k^{\text{new}} &= \frac{\sum_{n=1}^N r_{n,k} \mathbf{x}_n}{\boxed{\sum_{n=1}^N r_{n,k}}} = \frac{1}{\boxed{N_k}} \sum_{n=1}^N r_{n,k} \mathbf{x}_n, \end{aligned} \quad (11.23)$$

$$N_k := \sum_{n=1}^N r_{n,k} \quad (11.24)$$

$$r_k := [r_{1,k}, \dots, r_{n,k}]^\top / N_k, \quad (11.25)$$

$$\boldsymbol{\mu}_k \leftarrow \mathbb{E}_{r_k}[\mathcal{X}]. \quad (11.26)$$

$$\boldsymbol{\mu}_1 : -4 \rightarrow 2.7 \quad (11.27)$$

$$\boldsymbol{\mu}_2 : 0 \rightarrow -0.4 \quad (11.28)$$

$$\boldsymbol{\mu}_3 : 8 \rightarrow 3.7 \quad (11.29)$$

11.2.3 更新协方差矩阵

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N r_{n,k} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^{\top}, \quad (11.30)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = \sum_{n=1}^N \frac{\partial \log p(\mathbf{x}_n | \theta)}{\partial \boldsymbol{\Sigma}_k} = \sum_{n=1}^N \frac{1}{p(\mathbf{x}_n | \theta)} \frac{\partial p(\mathbf{x}_n | \theta)}{\partial \boldsymbol{\Sigma}_k}. \quad (11.31)$$

$$\begin{aligned} & \frac{\partial p(\mathbf{x}_n | \theta)}{\partial \boldsymbol{\Sigma}_k} \\ &= \frac{\partial}{\partial \boldsymbol{\Sigma}_k} \left(\pi_k (2\pi)^{-\frac{D}{2}} \det(\boldsymbol{\Sigma}_k)^{-\frac{1}{2}} \exp \left(-\frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^{\top} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \right) \right) \\ &= \pi_k (2\pi)^{-\frac{D}{2}} \left[\frac{\partial}{\partial \boldsymbol{\Sigma}_k} \det(\boldsymbol{\Sigma}_k)^{-\frac{1}{2}} \exp \left(-\frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^{\top} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \right) \right] \\ &+ \det(\boldsymbol{\Sigma}_k)^{-\frac{1}{2}} \frac{\partial}{\partial \boldsymbol{\Sigma}_k} \exp \left(-\frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^{\top} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \right). \end{aligned} \quad (11.32)$$

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_k} \det(\boldsymbol{\Sigma}_k)^{-1/2} \xrightarrow{(5.101)} -\frac{1}{2} \det(\boldsymbol{\Sigma}_k)^{-1/2} \boldsymbol{\Sigma}_k^{-1}, \quad (11.33)$$

$$\frac{\partial}{\partial \boldsymbol{\sigma}_k} (\mathbf{x}_n - \boldsymbol{\mu}_k)^{\top} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \xrightarrow{(5.103)} -\boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^{\top} \boldsymbol{\Sigma}_k^{-1}$$

$$\frac{\partial p(\mathbf{x}_n | \theta)}{\partial \boldsymbol{\Sigma}_k} = \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \cdot \left[-\frac{1}{2} (\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^{\top} \boldsymbol{\Sigma}_k^{-1}) \right].$$

$$\frac{\partial \mathcal{L}}{\partial \Sigma_k} = \sum_{n=1}^N \frac{\partial \log p(\mathbf{x}_n | \theta)}{\partial \Sigma_k} = \sum_{n=1}^N \frac{1}{p(\mathbf{x}_n | \theta)} \frac{\partial p(\mathbf{x}_n | \theta)}{\partial \Sigma_k} \quad (11.36a)$$

$$= \sum_{n=1}^N \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \Sigma_j)}}_{=r_{nk}} \cdot \left[-\frac{1}{2} (\Sigma_k^{-1} - \Sigma_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}) \right] \quad (11.36b)$$

$$= -\frac{1}{2} \sum_{n=1}^N r_{nk} (\Sigma_k^{-1} - \Sigma_k^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}) \quad (11.36c)$$

$$= -\frac{1}{2} \underbrace{\Sigma_k^{-1} \sum_{n=1}^N r_{nk}}_{=N_k} + \frac{1}{2} \Sigma_k^{-1} \left(\sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \right) \quad (11.36d)$$

$$N_k \Sigma_k^{-1} = \Sigma_k^{-1} \left[\sum_{n=1}^N r_{n,k} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \right] \Sigma_k^{-1} \quad (11.37a)$$

$$\iff N_k \mathbf{I} = \left[\sum_{n=1}^N r_{n,k} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top \right] \Sigma_k^{-1}. \quad (11.37b)$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N r_{n,k} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top, \quad (11.38)$$

$$\boldsymbol{\mu}_1 : 1 \rightarrow 0.14 \quad (11.39)$$

$$\boldsymbol{\mu}_2 : 0.2 \rightarrow 0.44 \quad (11.40)$$

$$\boldsymbol{\mu}_3 : 3 \rightarrow 1.53 \quad (11.41)$$

11.2.4 更新混合权重

$$\pi_k^{\text{new}} = \frac{N_k}{N}, \quad k = 1, \dots, K, \quad (11.42)$$

$$\mathfrak{L} = \mathcal{L} + \lambda \left(\sum_{k=1}^K \pi_k - 1 \right) \quad (11.43a)$$

$$= \sum_{n=1}^N \log \left[\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k) \right] + \lambda \left(\sum_{k=1}^K \pi_k - 1 \right) \quad (11.43b)$$

$$\frac{\partial \mathcal{L}}{\partial \pi_k} = \sum_{n=1}^N \frac{\mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda \quad (11.44a)$$

$$= \frac{1}{\pi_k} \sum_{n=1}^N \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda = \frac{N_k}{\pi_k} + \lambda \quad (11.44b)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{k=1}^K \pi_k - 1. \quad (11.45)$$

$$\pi_k = -\frac{N_k}{\lambda}, \quad (11.46)$$

$$1 = \sum_{k=1}^K \pi_k. \quad (11.47)$$

$$\sum_{k=1}^K \pi_k = 1 \iff -\sum_{k=1}^K \frac{N_k}{\lambda} = 1 \iff -\frac{N}{\lambda} = 1 \iff \lambda = -N \quad (11.48)$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}, \quad (11.49)$$

$$\pi_1 : \frac{1}{2} \rightarrow 0.29 \quad (11.50)$$

$$\pi_2 : \frac{1}{3} \rightarrow 0.29 \quad (11.51)$$

$$\pi_3 : \frac{1}{3} \rightarrow 0.42 \quad (11.52)$$

< 上一章节

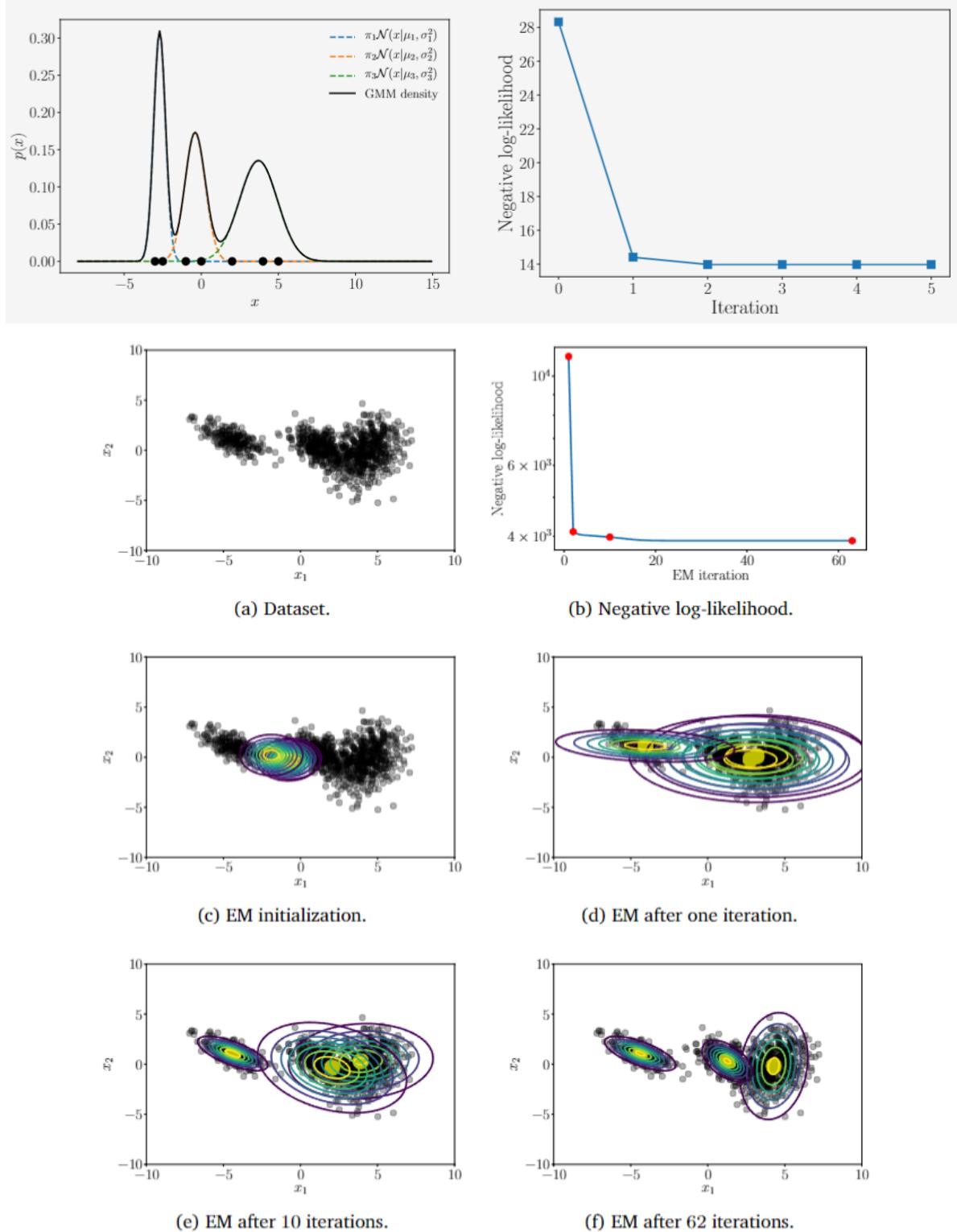
下一章节 >

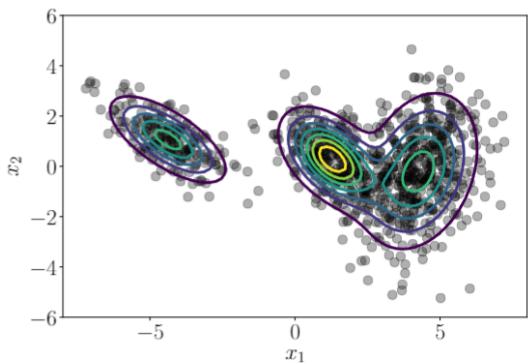
11.1 混合Gauss模型

11.3 EM 算法

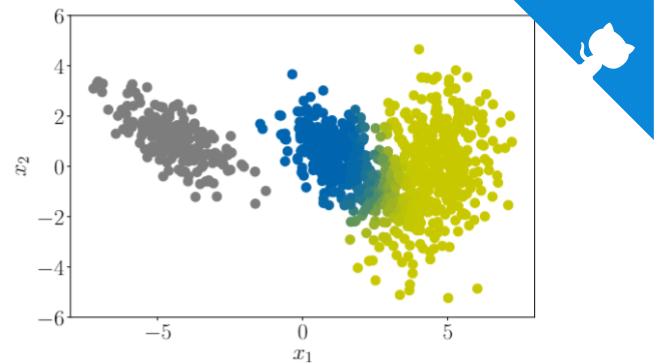


11.3 EM 算法





(a) GMM fit after 62 iterations.



(b) Dataset colored according to the responsibilities of the mixture components.

$$r_{n,k} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \quad (11.53)$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N r_{n,k} \mathbf{x}_n, \quad (11.54)$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N r_{n,k} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\top, \quad (11.55)$$

$$\pi_k = \frac{N_k}{N}. \quad (11.56)$$

$$p(x) = 0.29\mathcal{N}(x | -0.275, 0.06) + 0.28\mathcal{N}(x | -0.50, 0.25) + 0.43\mathcal{N}(x | 3.74, 1.6^2)$$

< 上一章节

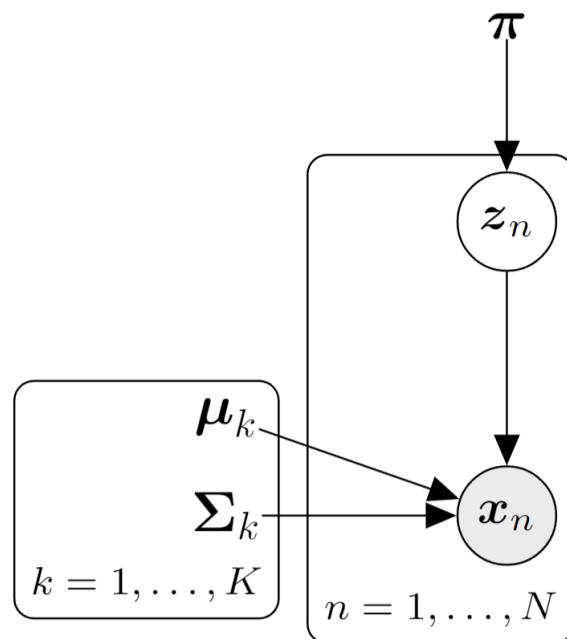
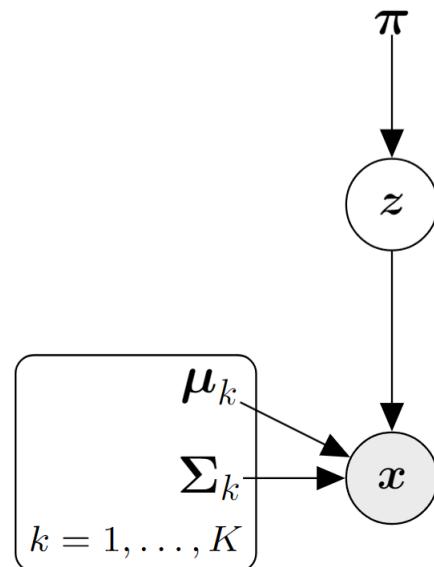
下一章节 >

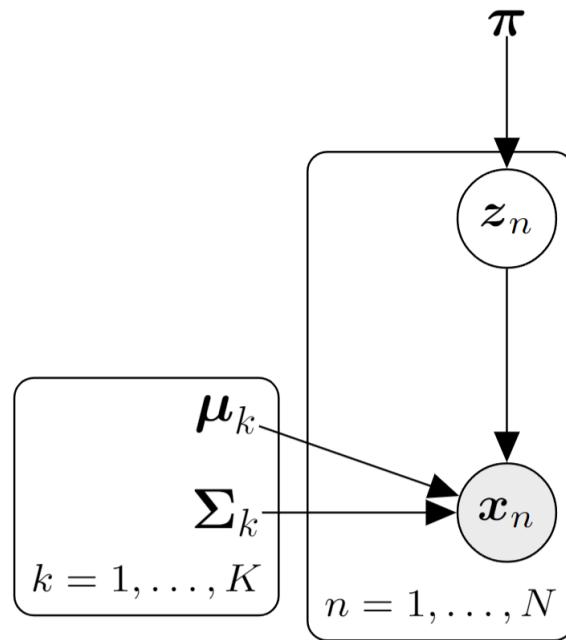
11.2 通过最大似然估计学习参数

11.4 隐变量的视角



11.4 隐变量的视角





11.4.1 生成过程和概率模型

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \quad (11.58)$$

$$p(z) = \boldsymbol{\pi} = [\pi_1, \dots, \pi_K]^\top, \quad \sum_{k=1}^K \pi_k = 1, \quad (11.59)$$

$$\pi_k = p(z_k = 1) \quad (11.60)$$

$$p(\mathbf{x}, z_k = 1) = p(\mathbf{x}|z_k = 1)p(z_k = 1) = \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (11.61)$$

$$p(\mathbf{x}, \mathbf{z}) = \begin{bmatrix} p(\mathbf{x}, z_1 = 1) \\ \vdots \\ p(\mathbf{x}, z_K = 1) \end{bmatrix} = \begin{bmatrix} \pi_1 \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \\ \vdots \\ \pi_K \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K) \end{bmatrix},$$

11.4.2 似然函数

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_z p(\mathbf{x}|\boldsymbol{\theta}, z)p(z|\boldsymbol{\theta}), \quad \boldsymbol{\theta} := \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k : k, 1, \dots, K\} \quad (11.63)$$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (11.64)$$

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_z p(\mathbf{x}|\boldsymbol{\theta}, z)p(z|\boldsymbol{\theta}) \quad (11.65a)$$

$$= \sum_{k=1}^K p(\mathbf{x}|\boldsymbol{\theta}, z_k = 1)p(z_k = 1|\boldsymbol{\theta}) \quad (11.65b)$$

$$p(\mathbf{x}|\boldsymbol{\theta}) \xrightarrow{(11.65b)} \sum_{k=1}^K p(\mathbf{x}|\boldsymbol{\theta}, z_k = 1)p(z_k = 1|\boldsymbol{\theta}) \quad (11.66a)$$

$$= \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (11.66b)$$

$$p(\mathcal{X}|\boldsymbol{\theta}) = \prod_{n=1}^N p(\mathbf{x}_n|\boldsymbol{\theta}) \xrightarrow{(11.66b)} \prod_{n=1}^N \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (11.67)$$

11.4.3 后验分布

$$p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{p(\mathbf{x})} \quad (11.68)$$

$$p(z_k = 1|\mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x}|z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x}|z_j = 1)} = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \quad (11.69)$$

11.4.4 延拓至整个数据集

$$\mathbf{z}_n = [z_{n,1}, \dots, z_{n,K}]^\top \in \mathbb{R}^K. \quad (11.70)$$

$$p(\mathbf{x}_1, \dots, \mathbf{x}_N | \mathbf{z}_1, \dots, \mathbf{z}_N) = \prod_{n=1}^N p(\mathbf{x}_n | \mathbf{z}_n). \quad (11.71)$$

$$p(z_{n,k} = 1|\mathbf{x}_n) = \frac{p(z_{n,k} = 1)p(\mathbf{x}_n|z_{n,k} = 1)}{\sum_{j=1}^K p(z_{n,j} = 1)p(\mathbf{x}_n|z_{n,j} = 1)} \quad (11.72a)$$

$$= \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = r_{n,k}. \quad (11.72b)$$

11.4.5 重访 EM 算法

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}^{(t)}}[\log p(\mathbf{x},\mathbf{z}|\boldsymbol{\theta})] \quad (11.73a)$$

$$= \int p(\mathbf{z}|\mathbf{x},\boldsymbol{\theta}^{(t)}) \cdot \log p(\mathbf{x},\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} \quad (11.73b)$$

< 上一章节

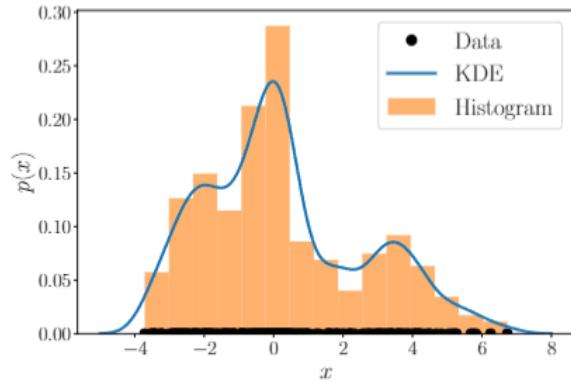
下一章节 >

11.3 EM 算法

11.5 拓展阅读



11.5 拓展阅读



$$p(\mathbf{x}) = \frac{1}{Nh} \sum_{n=1}^N k\left(\frac{\mathbf{x} - \mathbf{x}_n}{h}\right), \quad (11.74)$$

< 上一章节

11.4 隐变量的视角



第12章 支持向量机分类

在许多情况下，我们希望机器学习算法能够预测多个（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，这就有两种结果。另一个例子是望远镜识别夜空中的对象是星系、恒星还是行星。通常结果的数量很少，而且更重要的是，这些结果之间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。这种机器学习任务被称为二分类。这与第9章不同，第9章我们考虑的是具有连续值输出的预测问题。

对于二分类，标签/输出可能取得的值集合是二进制的，在本章中我们用 $+1, -1$ 来表示它们。换句话说，我们考虑的预测器形式为

$$f : \mathbb{R}^D \rightarrow \{+1, -1\}.$$

(12.1)

回顾第8章，我们将每个示例（数据点） \mathbf{x}_n 表示为一个 D 个实数的特征向量。标签通常分别称为正类和负类。需要注意的是，不要从 $+1$ 类的正性中推断出直观的属性。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 $True, False$ 、 $0, 1$ 或 $red, blue$ 。二分类问题已经被广泛研究，我们将在第12.6节介绍其他方法。

我们介绍了一种称为支持向量机（SVM）的方法，它解决了二分类任务。与回归一样，我们有一个监督学习任务，其中我们有一组示例 $\mathbf{x}_n \in \mathbb{R}^D$ 以及它们对应的（二进制）标签 $y_n \in +1, -1$ 。给定一个由示例-标签对 $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ 组成的训练数据集，我们希望估计模型参数，以便给出最小的分类错误。与第9章类似，我们考虑一个线性模型，并将非线性隐藏在示例的变换 ϕ 中（9.13）。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，并具有可靠的理论保证（Steinwart and Christmann, 2008）。我们选择使用SVM来说明二分类有两个主要原因。首先，SVM允许我们通过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了机器学习问题，并使用最大似然估计和贝叶斯推断来攻击它，但在这里我们将考虑一种替代方法，即我们对机器学习任务进行几何推理。它大量依赖于我们在第3章中讨论的内积和投影等概念。其次，我们发现SVM具有启发性是因为与第9章不

同，SVM的优化问题不允许解析解，因此我们需要求助于第7章中介绍的各种优化工具。

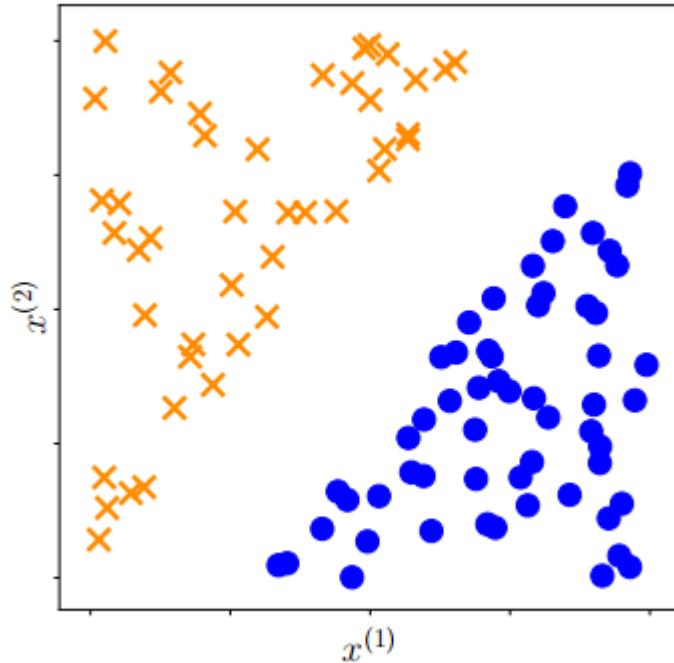


图12.1例子2D数据，说明了数据的直觉，我们可以找到一个线性分类器，分离橙色交叉和蓝色圆盘。

SVM对机器学习的看法与第9章的最大似然观点略有不同。最大似然观点基于数据的概率分布模型提出一个模型，并从中推导出优化问题。相比之下，SVM观点则是从基于几何直觉的设计一个特定函数开始，该函数在训练过程中需要被优化。我们在第10章中已经看到了类似的东西，其中我们从几何原理推导出PCA。在SVM的情况下，我们首先从设计一个损失函数开始，该损失函数在训练数据上需要被最小化，遵循经验风险最小化原则（第8.2节）。

让我们推导出与在样本-标签对上训练支持向量机（SVM）相对应的优化问题。直观上，我们想象二分类数据，这些数据可以通过一个超平面进行分离，如图12.1所示。在这里，每个样本 \mathbf{x}_n （一个二维向量）是一个二维位置（由 $\mathbf{x}_n^{(1)}$ 和 $\mathbf{x}_n^{(2)}$ 组成），而对应的二分类标签 y_n 是两种不同符号之一（橙色叉或蓝色圆）。‘超平面’是机器学习中常用的术语，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（如果对应的线性空间维度为 D ）。样本由两类组成（有两个可能的标签），这些样本的特征（表示样本的向量的分量）以这样的方式排列，使得我们可以通过画一条直线来分离/分类它们。

接下来，我们将寻找两个类别之间线性分隔器的想法形式化。我们引入间隔的概念，然后将线性分隔器扩展到允许样本落在“错误”的一侧，从而产生分类错误。我们提出了两种等价的方式来形式化SVM：几何视角（第12.2.4节）和损失函数视角（第12.2.5节）。我们使用Lagrange 乘数法推导出SVM的对偶形式（第7.2节）。对偶SVM使我们能够观察到第三种形式化SVM的方式：即基于每个类别样本的凸包（第12.3.2节）。最后，我们简要介绍了核函数以及如何数值求解非线性核SVM优化问题。

< 上一章节

第十一章 密度估计和混合Gauss模型



12.1 分隔超平面

给定两个以向量形式表示的样本 \mathbf{x}_i 和 \mathbf{x}_j ，计算它们之间相似度的一种方法是使用内积 $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ 。回顾第3.2节，内积与两个向量之间的角度紧密相关。两个向量之间的内积值取决于每个向量的长度（范数）。此外，内积使我们能够严格定义诸如正交性和投影等几何概念。

许多分类算法背后的主要思想是将数据表示为 \mathbb{R}^D 中的点，然后对这个空间进行划分，理想情况下是使得具有相同标签的样本（且没有其他样本）位于同一划分中。在二分类的情况下，空间将被分成两部分，分别对应于正类和负类。我们考虑一种特别方便的划分方式，即使用超平面（线性地）将空间分成两半。设样本 $\mathbf{x} \in \mathbb{R}^D$ 是数据空间中的一个元素。考虑一个函数

$$\begin{aligned} f : \mathbb{R}^D &\rightarrow \mathbb{R} \\ \mathbf{x} \mapsto f(\mathbf{x}) &:= \langle \mathbf{w}, \mathbf{x} \rangle + b, \end{aligned}$$

(12.2a) (12.2b)

其中参数为 $\mathbf{w} \in \mathbb{R}^D$ 和 $b \in \mathbb{R}$ 。回顾第2.8节，超平面是仿射子空间。因此，我们将二分类问题中分隔两个类别的超平面定义为

$$\{ \mathbf{x} \in \mathbb{R}^D : f(\mathbf{x}) = 0 \}.$$

(12.3)

超平面的一个图示如图12.2所示，其中向量 \mathbf{w} 是超平面的法向量， b 是截距。我们可以选择超平面上的任意两个样本 \mathbf{x}_a 和 \mathbf{x}_b ，并证明它们之间的向量与 \mathbf{w} 正交，来推导出 \mathbf{w} 是超平面(12.3)的法向量。以方程的形式表示，

$$\begin{aligned} f(\mathbf{x}_a) - f(\mathbf{x}_b) &= \langle \mathbf{w}, \mathbf{x}_a \rangle + b - (\langle \mathbf{w}, \mathbf{x}_b \rangle + b) \\ &= \langle \mathbf{w}, \mathbf{x}_a - \mathbf{x}_b \rangle, \end{aligned}$$

其中第二行是通过内积的线性性质（第3.2节）得到的。由于我们已经选择 \mathbf{x}_a 和 \mathbf{x}_b 在超平面上，这意味着 $f(\mathbf{x}_a) = 0$ 和 $f(\mathbf{x}_b) = 0$ ，因此 $\langle \mathbf{w}, \mathbf{x}_a - \mathbf{x}_b \rangle = 0$ 。回忆两个向量当且仅当它们的内积为零时正交。因此，我们得到 \mathbf{w} 与超平面上的任何向量都正交。

注：回顾第2章，我们知道可以以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。相比之下，我们将样本向量 x 视为数据点（由其坐标指示），即我们认为 x 是相对于标准基向量的向量坐标。

当给出一个测试样本时，我们根据它位于超平面的哪一侧来将其分类为正或负。请注意，(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面和负面。因此，为了对测试样本 x_{test} 进行分类，我们计算函数 $f(x_{\text{test}})$ 的值，并在 $f(x_{\text{test}}) \geq 0$ 时将其分类为+1，否则分类为-1。从几何角度来看，正样本位于超平面的“上方”，而负样本位于超平面的“下方”。

在训练分类器时，我们希望确保带有正标签的样本位于超平面的正面，即

(12.5)

$$\langle w, x_n \rangle + b \geq 0 \quad \text{当 } y_n = +1$$

并且带有负标签的样本位于超平面的负面，即

(12.6)

$$\langle w, x_n \rangle + b < 0 \quad \text{当 } y_n = -1.$$

参考图12.2，可以获得正负样本的几何直观理解。这两个条件通常可以合并为一个方程

(12.7)

$$y_n(\langle w, x_n \rangle + b) \geq 0.$$

当我们分别在(12.5)和(12.6)的两边乘以 $y_n = 1$ 和 $y_n = -1$ 时，方程(12.7)与(12.5)和(12.6)是等价的。

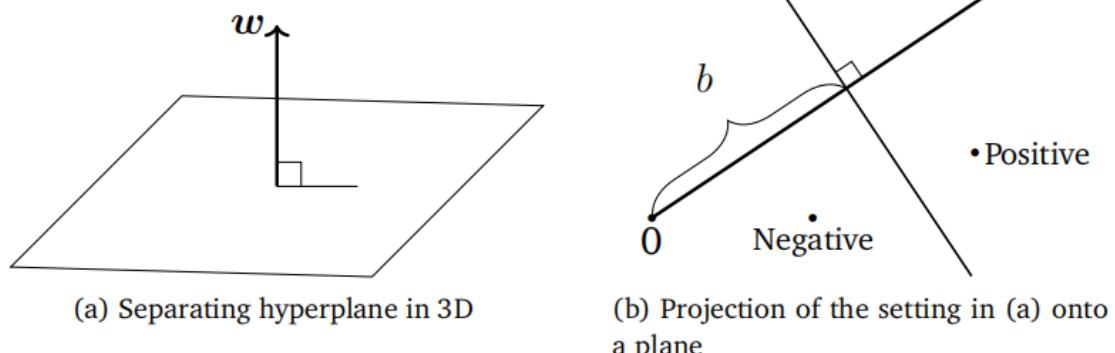


图12.2 分隔超平面(12.3)的方程 (a) 3D中方程的标准表示方式 (b) 为了便于绘制, 我们从侧面查看超平面。

下一章节 >

12.2 初级支持向量机



12.2 初级支持向量机

基于点到超平面的距离概念，我们现在可以讨论支持向量机了。对于一个线性可分的数据集 $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ ，我们有无数个候选超平面（参考图12.3），因此也有无数个分类器，它们可以在没有任何（训练）错误的情况下解决我们的分类问题。为了找到一个唯一解，一个想法是选择分隔超平面，该超平面最大化正例和反例之间的间隔。换句话说，我们希望正例和反例被一个较大的间隔分开（第12.2.1节）。接下来，我们计算一个样本与超平面之间的距离，以推导出这个间隔。回想一下，给定点（样本 \mathbf{x}_n ）到超平面上最近点的距离是通过正交投影获得的（第3.8节）。

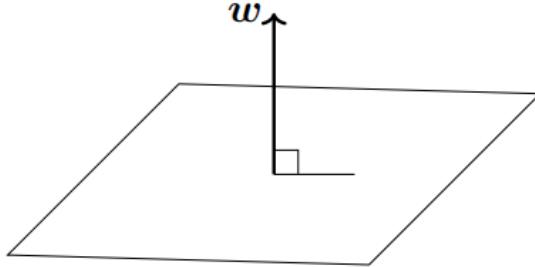
12.2.1 间隔的概念

间隔的概念直观上很简单：在假设数据集是线性可分的情况下，它是分隔超平面到数据集中最近样本的距离。然而，在尝试将这个距离形式化时，可能会遇到一个技术上的难题。这个技术难题在于我们需要定义一个测量距离的尺度。一个潜在的尺度是考虑数据的尺度，即 \mathbf{x}_n 的原始值。但这存在问题，因为我们可以改变 \mathbf{x}_n 的测量单位，从而改变 \mathbf{x}_n 中的值，进而改变到超平面的距离。正如我们稍后将看到的，我们将基于超平面方程(12.3)本身来定义这个尺度。

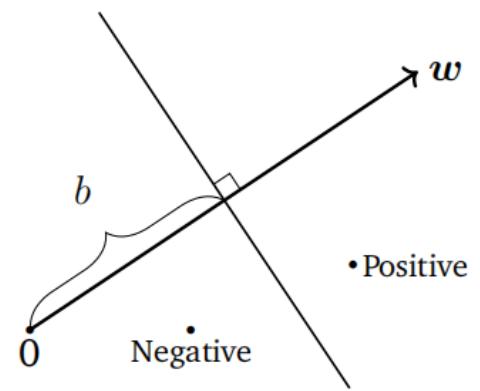
考虑一个超平面 $\langle \mathbf{w}, \mathbf{x} \rangle + b$ 和一个样本 \mathbf{x}_α ，如图12.4所示。不失一般性，我们可以考虑样本 \mathbf{x}_α 位于超平面的正面，即 $\langle \mathbf{w}, \mathbf{x}_\alpha \rangle + b > 0$ 。我们想要计算 \mathbf{x}_α 到超平面的距离 $r > 0$ 。我们通过考虑 \mathbf{x}_α 到超平面的正交投影（第3.8节）来实现这一点，我们将其表示为 \mathbf{x}'_α 。由于 \mathbf{w} 与超平面正交，我们知道距离 r 只是这个向量 \mathbf{w} 的一个缩放。如果知道 \mathbf{w} 的长度，那么我们可以使用这个缩放因子 r 来计算 \mathbf{x}_α 和 \mathbf{x}'_α 之间的绝对距离。为了方便起见，我们选择使用单位长度的向量（其范数为1），并通过将 \mathbf{w} 除以其范数 $\frac{\mathbf{w}}{\|\mathbf{w}\|}$ 来获得。使用向量加法（第2.4节），我们得到

(12.8)

$$\mathbf{x}_\alpha = \mathbf{x}'_\alpha + r \frac{\mathbf{w}}{\|\mathbf{w}\|}.$$



(a) Separating hyperplane in 3D



(b) Projection of the setting in (a) onto a plane

图12.2分离超平面方程 (12.3)。 (a)用三维方法表示该方程的标准方法。(b)为了便于绘制, 我们将查看超平面边缘。

另一种思考 r 的方式是, 它是 x_α 在由 $w/\|w\|$ 跨越的子空间中的坐标。现在我们已经将 x_α 到超平面的距离表示为 r , 如果我们选择 x_α 为最接近超平面的点, 那么这个距离 r 就是间隔。

回想一下, 我们希望正样本距离超平面超过 r , 负样本距离超平面 (在负方向上) 超过距离 r 。类似于将(12.5)和(12.6)组合成(12.7), 我们将这个目标表述为

(12.9)

$$y_n(\langle \mathbf{w}, \mathbf{x}_n \rangle + b) \geq r .$$

换句话说, 我们将样本至少距离超平面 r (在正方向和负方向上) 的要求合并为一个不等式。由于我们只关心方向, 我们在模型中增加了一个假设, 即参数向量 w 的单位长度为1, 即 $\|\mathbf{w}\| = 1$, 其中我们使用 Euclid 范数 $\|\mathbf{w}\| = \sqrt{\mathbf{w}^\top \mathbf{w}}$ (第3.1节)。这个假设也使得对距离 r (12.8) 的解释更加直观, 因为它是长度为1的向量的缩放因子。

备注。熟悉其他间隔表示法的读者会注意到, 如果支持向量机 (SVM) 是由 Schölkopf 和 Smola (2002) et al. 提供的, 那么我们对 $\|\mathbf{w}\| = 1$ 的定义与标准表示法不同。在第12.2.3节中, 我们将展示这两种方法的等价性。

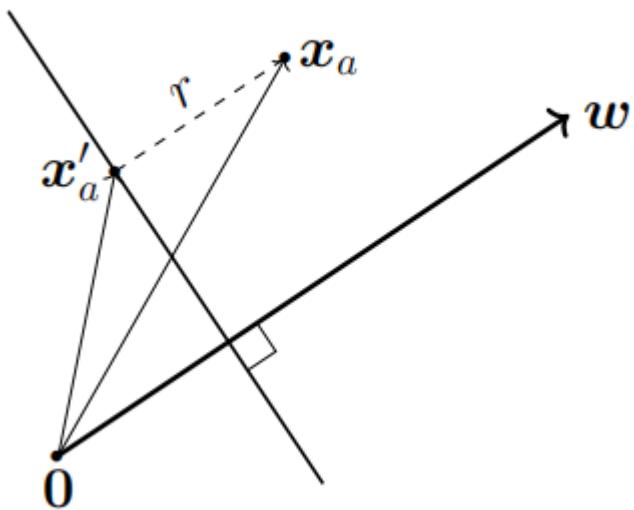


图12.4表示到超平面距离的向量加法: $\mathbf{x}_a = \mathbf{x}_0 + r \mathbf{w}$

将这三个要求合并为一个带约束的优化问题, 我们得到目标函数

(12.10)

$$\begin{aligned} & \max_{w,b,r} \underbrace{r}_{\text{margin}} \\ & \text{subject to } \underbrace{y_n(\langle w, x_n \rangle + b) \geq r}_{\text{数据拟合}}, \underbrace{\|w\| = 1}_{\text{归一化}}, \quad r > 0, \end{aligned}$$

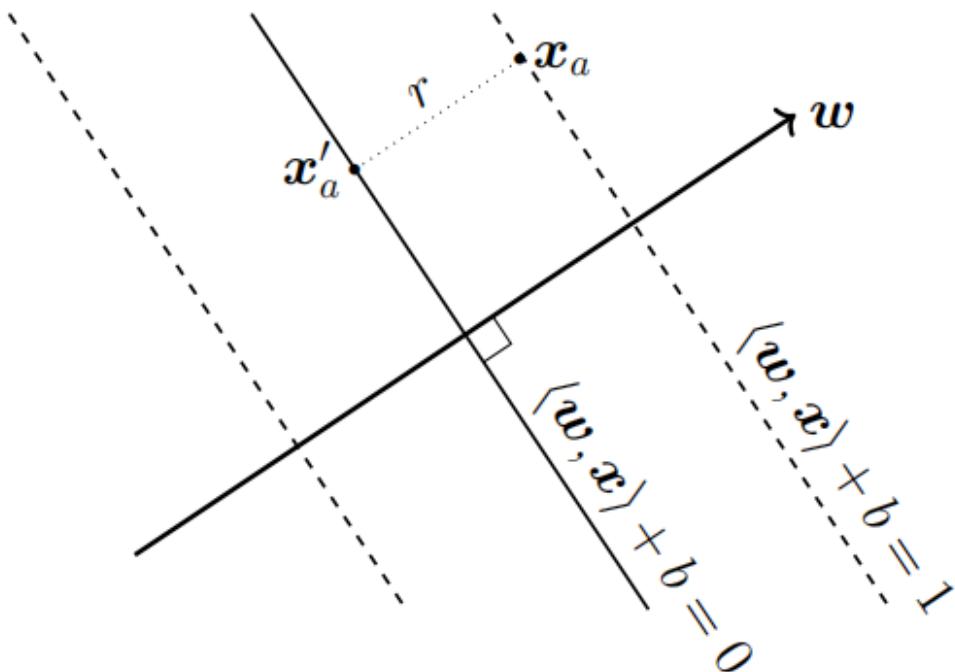


图12.5边际的推导: $r = 1$ k周。

这表示我们想要最大化间隔 r , 同时确保数据位于超平面的正确一侧。

备注。间隔的概念在机器学习中非常普遍。Vladimir Vapnik和Alexey Chervonenkis使用这个概念来表明, 当间隔较大时, 函数类的“复杂性”较低, 因此学习是可能的(Vapnik, 2000)。事实证明, 这个概念对于从理论上分析泛化误差的各种不同方法非常有用(Steinwart和Christmann, 2008; Shalev-Shwartz和Ben-David, 2014)。

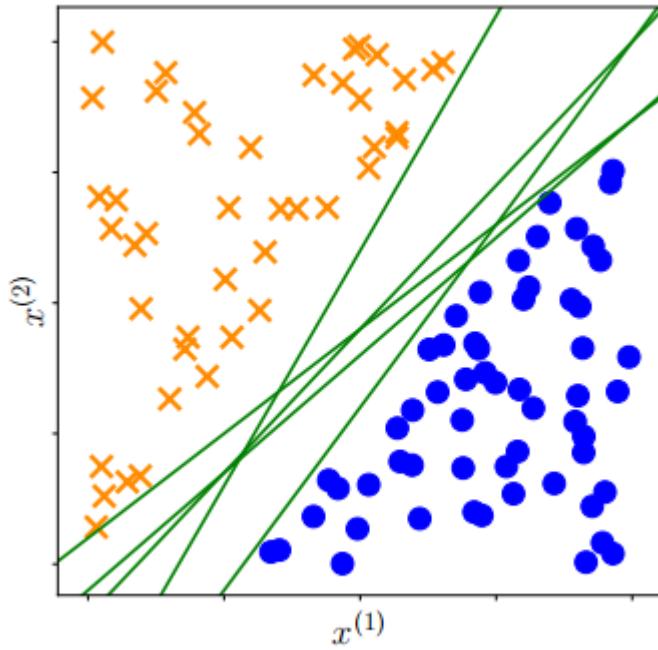


图12.3可能的分离的超平面。有许多线性分类器（绿色的线）将橙色的交叉和蓝色的圆盘分开。

12.2.2 间隔的传统推导

在上一节中，我们通过观察到我们只关心 w 的方向而不是其长度，从而得出了(12.10)，并假设了 $\|w\| = 1$ 。在本节中，我们将通过不同的假设来推导间隔最大化问题。我们不是选择参数向量进行归一化，而是选择数据的比例尺。我们选择这个比例尺，使得预测器 $\langle w, \mathbf{x} \rangle + b$ 在最接近的样本上的值为1。我们还将数据集中最接近超平面的样本表示为 x_a 。

图12.5与图12.4相同，但现在我们重新调整了坐标轴的比例，使得样本 x_a 正好位于间隔上，即 $\langle w, x_a \rangle + b = 1$ 。由于 x'_a 是 x_a 在超平面上的正交投影，根据定义，它必须位于超平面上，即

(12.11)

$$\langle w, x'_a \rangle + b = 0 .$$

将(12.8)代入(12.11)，我们得到

(12.12)



$$\left\langle \mathbf{w}, \mathbf{x}_a - r \frac{\mathbf{w}}{\|\mathbf{w}\|} \right\rangle + b = 0.$$

利用内积的双线性性质（见第3.2节），我们得到

$$\langle \mathbf{w}, \mathbf{x}_a \rangle + b - r \frac{\langle \mathbf{w}, \mathbf{w} \rangle}{\|\mathbf{w}\|} = 0.$$

(12.13)

根据我们设定的比例尺，第一项为1，即 $\langle \mathbf{w}, \mathbf{x}_a \rangle + b = 1$ 。从第3.1节的(3.16)中，我们知道 $\langle \mathbf{w}, \mathbf{w} \rangle = \|\mathbf{w}\|^2$ 。因此，第二项简化为 $r\|\mathbf{w}\|$ 。使用这些简化，我们得到

(12.14)

$$r = \frac{1}{\|\mathbf{w}\|}.$$

这意味着我们根据超平面的法向量 \mathbf{w} 推导出了距离 r 。乍一看，这个方程似乎有些反直觉，因为我们似乎是用向量 \mathbf{w} 的长度来表示了到超平面的距离，但我们还不知道这个向量。一种思考方式是将距离 r 视为一个临时变量，我们仅在此推导中使用它。因此，在本节的其余部分，我们将到超平面的距离表示为 $\frac{1}{\|\mathbf{w}\|}$ 。在第12.2.3节中，我们将看到选择间隔等于1与我们在第12.2.1节中的假设 $\|\mathbf{w}\| = 1$ 是等价的。

类似于获得(12.9)的论证，我们希望正样本和负样本都至少距离超平面1个单位，这产生了条件

(12.15)

$$y_n(\langle \mathbf{w}, \mathbf{x}_n \rangle + b) \geq 1.$$

将间隔最大化与样本需要根据其标签位于超平面的正确一侧这一事实相结合，我们得到

(12.16)

$$\begin{aligned} & \max_{\mathbf{w}, b} \quad \frac{1}{\|\mathbf{w}\|} \\ & \text{subject to } y_n(\langle \mathbf{w}, \mathbf{x}_n \rangle + b) \geq 1 \quad \text{for all } n = 1, \dots, N. \end{aligned}$$

(12.17)

而不是像 (12.16) 那样最大化范数的倒数，我们通常最小化范数的平方。我们还经常包含一个常数 $\frac{1}{2}$ ，它不会影响最优的 w, b ，但在我们计算梯度时会得到一个更整洁的形式。然后，我们的目标变为

(12.18)

$$\min_{w,b} \quad \frac{1}{2} \|w\|^2$$

受约束于 $y_n(\langle w, x_n \rangle + b) \geq 1$ 对所有 $n = 1, \dots, N$ 成立。 (12.19)

方程 (12.18) 被称为硬间隔SVM。“硬”这个表达的原因是因为该公式不允许间隔条件有任何违反。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“硬”条件可以放宽以容纳违反情况。

12.2.3 为什么我们可以将间隔设置为1

在12.2.1节中，我们论证了希望最大化某个值 r ，它代表最接近超平面的样本点的距离。在12.2.2节中，我们对数据进行了缩放，使得最接近超平面的样本点到超平面的距离为1。在本节中，我们将这两个推导联系起来，并证明它们是等价的。

定理12.1. 最大化间隔 r ，其中我们考虑如(12.10)所示的规范化权重，

$$\max_{w,b,r} \underbrace{r}_{\text{间隔}}$$

约束条件 $\underbrace{y_n(\langle w, x_n \rangle + b) \geq r}_{\text{数据拟合}}, \quad \underbrace{\|w\| = 1}_{\text{规范化}}, \quad r > 0$

(12.20)

这等价于对数据进行缩放，使得间隔为1：

(12.21)

$$\min_{w,b} \quad \underbrace{\frac{1}{2} \|w\|^2}_{\text{间隔}}$$

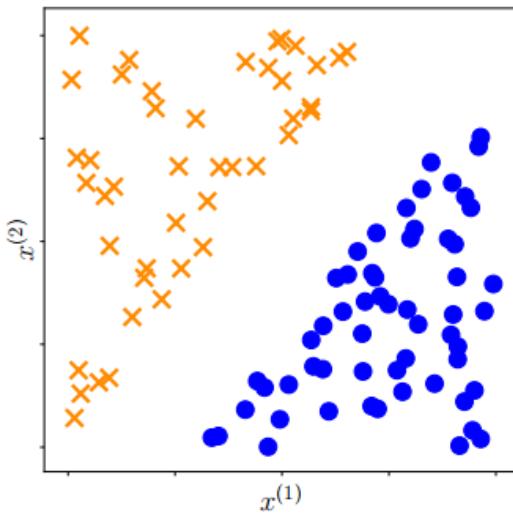
约束条件 $\underbrace{y_n(\langle w, x_n \rangle + b) \geq 1}_{\text{数据拟合}}.$

证明：考虑(12.20)。由于平方是对于非负参数的严格单调变换，如果我们在目标函数中考虑 r^2 ，则最大值保持不变。由于 $\|w\| = 1$ ，我们可以使用新的权重向量 w' （不显式地进行规范化）来重新参数化方程，即使用 $\frac{w'}{\|w'\|}$ 。我们得到

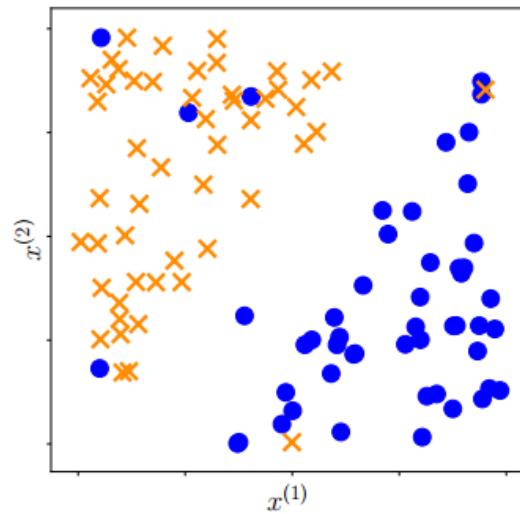
(12.22)

$$\max_{w', b, r} r^2$$

约束条件 $y_n \left(\underbrace{\left\langle \frac{w'}{\|w'\|}, \mathbf{x}_n \right\rangle + b}_{w''} \right) \geq r, \quad r > 0$



(a) Linearly separable data, with a large margin



(b) Non-linearly separable data

图12.6 (a)线性可分数据和(b)非线性可分数据。

方程(12.22)明确指出距离 r 是正数。因此，我们可以将第一个约束条件除以 r ，得到

(12.23)

$$\max_{w', b, r} r^2$$

约束条件 $y_n \left(\underbrace{\left\langle \frac{w'}{\|w'\| r}, \mathbf{x}_n \right\rangle + \underbrace{\frac{b}{r}}_{w''}}_{w''} \right) \geq 1, \quad r > 0$



将参数重命名为 w'' 和 b'' 。由于 $w'' = \frac{w'}{\|w'\|_r}$, 重新排列得到

$$\|w''\| = \left\| \frac{w'}{\|w'\|_r} \right\| = \frac{1}{r} \cdot \left\| \frac{w'}{\|w'\|} \right\| = \frac{1}{r}.$$

(12.24)

将这个结果代入(12.23), 我们得到

$$\begin{aligned} & \max_{w'', b''} \frac{1}{\|w''\|^2} \\ \text{约束条件 } & y_n (\langle w'', x_n \rangle + b'') \geq 1 \end{aligned}$$

(12.25)

最后一步是观察到, 最大化 $\frac{1}{\|w''\|^2}$ 与最小化 $\frac{1}{2}\|w''\|^2$ 得到相同的解, 这完成了定理12.1的证明。

12.2.4 软间隔支持向量机: 几何视角

当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间隔区域内, 甚至落在超平面的错误一侧, 如图12.6所示。允许一定分类错误的模型被称为软间隔支持向量机 (soft margin SVM)。在本节中, 我们将使用几何论证来推导出相应的优化问题。在12.2.5节中, 我们将使用损失函数的思想推导出等价的优化问题。利用Lagrange乘子 (第7.2节), 我们将在12.3节中推导出SVM的对偶优化问题。这个对偶优化问题使我们能够观察到SVM的第三种解释: 作为平分正样本和负样本凸包之间连线的超平面 (12.3.2节)。

关键的几何思想是引入一个松弛变量 ξ_n , 对应于每个样本-标签对 (x_n, y_n) , 允许特定样本位于间隔内甚至超平面的错误一侧 (参考图12.7)。我们从间隔中减去 ξ_n 的值, 并约束 ξ_n 为非负。为了鼓励样本的正确分类, 我们将 ξ_n 添加到目标函数中:

$$\begin{aligned} & \min_{w, b, \xi} \frac{1}{2} \|w\|^2 + C \sum_{n=1}^N \xi_n \\ \text{subject to } & y_n (\langle w, x_n \rangle + b) \geq 1 - \xi_n \\ & \xi_n \geq 0 \end{aligned}$$

(12.26a) (12.26b) (12.26c)

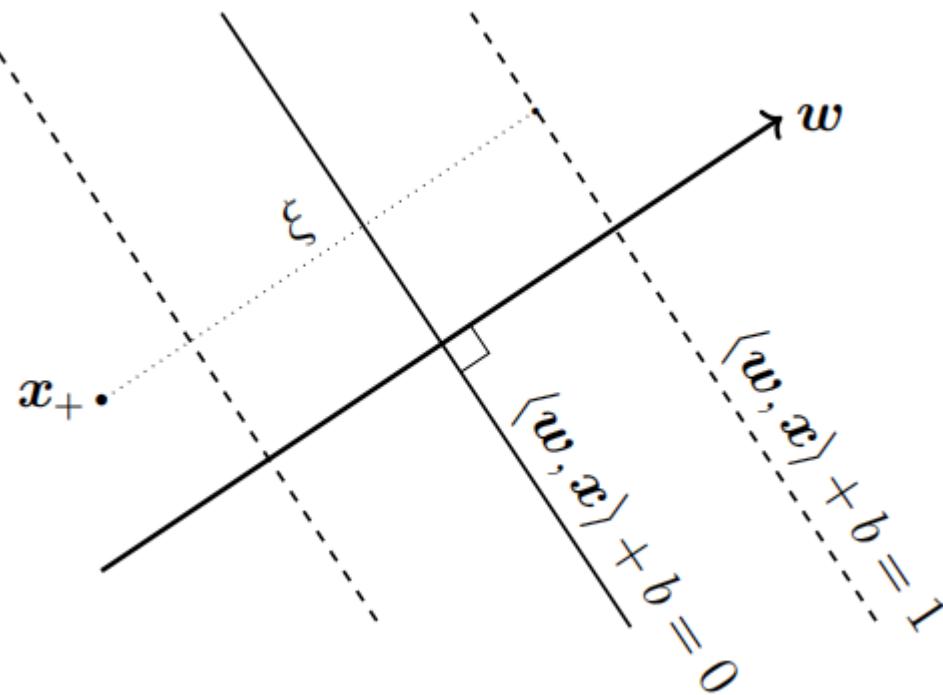


图12.7软边缘SVM允许示例在超平面的边缘内或在错误的一侧。当 x_+ 在错误的一侧时，松弛变量 ξ 测量一个正的例子 x_+ 到正的边缘超平面 $h(w, x) + b = 1$ 的距离。

对于 $n = 1, \dots, N$ 。与硬间隔SVM的优化问题（12.18）相比，这被称为软间隔SVM。参数 $C > 0$ 用于权衡间隔大小和总松弛量。这个参数被称为正则化参数，因为正如我们将在下一节看到的那样，目标函数（12.26a）中的间隔项是一个正则化项。间隔项 $\|w\|^2$ 被称为正则化器，在许多数值优化书籍中，正则化参数会乘以这个项（第8.2.3节）。这与我们在本节中的表述不同。在这里， C 的较大值意味着较低的正则化，因为我们给松弛变量更大的权重，因此更优先考虑不在间隔正确一侧的样本。

备注：在软间隔SVM的表述（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 来看到这一点。未正则化的项 b 使理论分析复杂化（Steinwart和Christmann, 2008, 第1章），并降低了计算效率（Fan等, 2008）。

12.2.5 软间隔支持向量机：损失函数视角

让我们考虑一种不同的方法来推导支持向量机（SVM），遵循经验风险最小化原则（第8.2节）。对于SVM，我们选择超平面作为假设类，即

(12.27)



$$f(x) = \langle \mathbf{w}, \mathbf{x} \rangle + b.$$

我们将在本节中看到，间隔对应于正则化项。剩下的问题是，损失函数是什么？与第9章考虑回归问题（预测器的输出是实数）不同，本章我们考虑二分类问题（预测器的输出是两个标签之一 $\{+1, -1\}$ ）。因此，每个样本-标签对的误差/损失函数需要适用于二分类。例如，用于回归的平方损失（9.10b）不适用于二分类。

备注：二进制标签之间的理想损失函数是计算预测与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，则损失为一。这表示为 $1_{(f(x_n) \neq y_n)}$ ，并称为零一损失。不幸的是，零一损失导致了一个组合优化问题，用于寻找最佳参数 w, b 。组合优化问题（与第7章中讨论的连续优化问题相比）通常更难解决。

◇

SVM对应的损失函数是什么？考虑预测器 $f(x_n)$ 的输出与标签 y_n 之间的误差。损失描述了训练数据上的误差。推导（12.26a）的等效方法是使用合页损失（hinge loss）

$$\ell(t) = \max\{0, 1 - t\} \quad \text{其中 } t = y f(\mathbf{x}) = y(\langle \mathbf{w}, \mathbf{x} \rangle + b).$$

(12.28)

如果 $f(\mathbf{x})$ 位于超平面的正确一侧（基于相应的标签 y ），并且距离大于1，这意味着 $t \geq 1$ ，并且合页损失返回零。如果 $f(\mathbf{x})$ 位于正确一侧但太接近超平面（ $0 < t < 1$ ），则样本 \mathbf{x} 位于间隔内，并且合页损失返回一个正值。当样本位于超平面的错误一侧（ $t < 0$ ）时，合页损失返回一个更大的值，该值线性增加。换句话说，一旦我们比间隔更接近超平面，即使预测是正确的，我们也会受到惩罚，并且该惩罚线性增加。合页损失的另一种表示方法是将其视为两个线性部分

$$\ell(t) = \begin{cases} 0 & \text{if } t \geq 1 \\ 1 - t & \text{if } t < 1 \end{cases},$$

(12.29)

如图12.8所示。硬间隔对应的损失定义为



$$\ell(t) = \begin{cases} 0 & \text{if } t \geq 1 \\ \infty & \text{if } t < 1 \end{cases}.$$

(12.30)

这种损失可以理解为绝不允许任何样本位于间隔内部。

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，同时使用 ℓ_2 正则化（见第8.2.3节）对目标进行正则化。使用合页损失（12.28），我们得到了无约束优化问题

(12.31)

$$\min_{w,b} \underbrace{\frac{1}{2} \|w\|^2}_{\text{正则化项}} + C \underbrace{\sum_{n=1}^N \max\{0, 1 - y_n(\langle w, x_n \rangle + b)\}}_{\text{误差项}}.$$

(12.31)中的第一项称为正则化项或正则器（见第8.2.3节），第二项称为损失项或误差项。回想第12.2.4节， $\frac{1}{2} \|w\|^2$ 这一项直接来源于间隔。换句话说，最大化间隔可以解释为正则化。

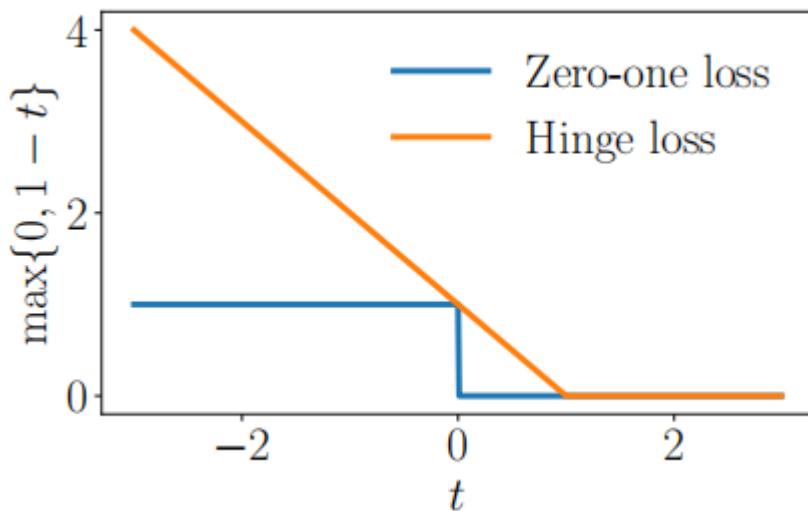


图12.8铰链损耗是零1损耗的凸上界。

原则上，(12.31)中的无约束优化问题可以直接用第7.1节中描述的（子）梯度下降法求解。为了看到(12.31)和(12.26a)是等价的，请注意合页损失（12.28）本质上由两部分线性函数组成，如(12.29)所示。考虑单个样本-标签对的合页损失（12.28）。我

们可以等价地将 t 上的合页损失最小化替换为带有两个约束的松弛变量 ξ 的最小化。

(12.32)

$$\min_t \max\{0, 1 - t\}$$

等价于

$$\begin{aligned} & \min_{\xi, t} \quad \xi \\ & \text{subject to} \quad \xi \geq 0, \quad \xi \geq 1 - t. \end{aligned}$$

(12.33)

将此表达式代入(12.31)并重新排列其中一个约束，我们正好得到软间隔SVM
(12.26a)。

备注：让我们将本节中选择的损失函数与第9章中线性回归的损失函数进行对比。回想第9.2.1节，为了找到最大似然估计量，我们通常最小化负对数似然。此外，由于带有高斯噪声的线性回归的似然项是高斯分布，因此每个样本的负对数似然是一个平方误差函数。平方误差函数是在寻找最大似然解时最小化的“损失函数”。

< 上一章节

下一章节 >

12.1 分隔超平面

12.3 对偶支持向量机



12.3 对偶支持向量机

前面几节中对支持向量机（SVM）的描述，涉及变量 w 和 b ，这被称为原始SVM。回想一下，我们考虑的输入 $x \in \mathbb{R}^D$ 具有 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（即 w 的维度）随特征数量的增加而线性增长。

接下来，我们考虑一个等效的优化问题（即所谓的对偶视图），它与特征数量无关。相反，参数的数量随训练集中样本数量的增加而增加。我们在第10章中看到过类似的想法，即以一种不随特征数量变化的方式来表达学习问题。这对于特征数量多于训练数据集中样本数量的问题非常有用。对偶SVM还具有另一个优点，即它很容易应用核函数，我们将在本章末尾看到这一点。在数学文献中，“对偶”一词经常出现，在这个特定情况下，它指的是凸对偶性。以下小节基本上是对我们在第7.2节中讨论的凸对偶性的应用。

12.3.1 通过Lagrangre 乘数法实现的凸对偶性

回顾原始软间隔SVM（12.26a）。我们将与原始SVM相对应的变量 w , b 和 ξ 称为原始变量。我们使用 $\alpha_n \geq 0$ 作为与约束（12.26b）相对应的Lagrangre 乘数，该约束要求样本被正确分类；使用 $\gamma_n \geq 0$ 作为与松弛变量的非负性约束相对应的Lagrangre 乘数；参见（12.26c）。然后，Lagrangre 函数由下式给出：

通过对Lagrangre 函数（12.34）分别关于三个原始变量 w , b 和 ξ 求导，我们得到：

(12.35)

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w} &= w^\top - \sum_{n=1}^N \alpha_n y_n x_n^\top, \\ \frac{\partial \mathcal{L}}{\partial b} &= - \sum_{n=1}^N \alpha_n y_n, \\ \frac{\partial \mathcal{L}}{\partial \xi_n} &= C - \alpha_n - \gamma_n.\end{aligned}$$

我们现在通过将每个偏导数设置为零来找到Lagrangre 函数的最大值。通过将 (12.35) 设置为零, 我们发现:

(12.38)

$$\boldsymbol{w} = \sum_{n=1}^N \alpha_n y_n \boldsymbol{x}_n ,$$

这是表示定理 (Kimeldorf和Wahba, 1970) 的一个特例。方程 (12.38) 表明, 原始问题中的最优权重向量是样本 \boldsymbol{x}_n 的线性组合。回想第2.6.1节的内容, 这意味着优化问题的解位于训练数据的张成空间中。此外, 通过将 (12.36) 设置为零得到的约束意味着最优权重向量是样本的仿射组合。表示定理对于正则化经验风险最小化的非常一般设置都是成立的 (Hofmann et al., 2008; Argyriou和Dinuzzo, 2014) 。该定理有更一般的形式 (Schölkopf et al., 2001) , 并且可以在Yu et al. (2013) 中找到其存在性的必要和充分条件。

备注。表示定理 (12.38) 还解释了“支持向量机”这个名字的由来。对于对应的参数 $\alpha_n = 0$ 的样本 \boldsymbol{x}_n , 它们对解 \boldsymbol{w} 没有任何贡献。而其他 $\alpha_n > 0$ 的样本被称为支持向量, 因为它们“支撑”超平面。

通过将 \boldsymbol{w} 的表达式代入Lagrangre 函数 (12.34) , 我们得到对偶函数

$$\begin{aligned} \mathfrak{D}(\xi, \alpha, \gamma) = & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle - \sum_{i=1}^N y_i \alpha_i \left\langle \sum_{j=1}^N y_j \alpha_j \boldsymbol{x}_j, \boldsymbol{x}_i \right\rangle \\ & + C \sum_{i=1}^N \xi_i - b \sum_{i=1}^N y_i \alpha_i + \sum_{i=1}^N \alpha_i - \sum_{i=1}^N \alpha_i \xi_i - \sum_{i=1}^N \gamma_i \xi_i . \end{aligned}$$

注意, 此时已经不再包含原始变量 \boldsymbol{w} 的任何项。通过将 (12.36) 设置为零, 我们得到 $\sum_{n=1}^N y_n \alpha_n = 0$ 。因此, 包含 b 的项也消失了。回想一下, 内积是对称且双线性的 (参见第3.2节)。因此, (12.39) 中的前两项是针对相同对象的。这些项 (用蓝色标记) 可以简化, 我们得到Lagrangre 函数

$$\mathfrak{D}(\xi, \alpha, \gamma) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \sum_{i=1}^N \alpha_i + \sum_{i=1}^N (C - \alpha_i - \gamma_i) \xi_i .$$

该方程中的最后一项是所有包含松弛变量 ξ_i 的项的集合。通过将 (12.37) 设置为零, 我们可以看到 (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Lagrangre 乘数 γ_i 是非负的, 我们得出 $\alpha_i \leq C$ 。现在, 我们得到了SVM的对偶优

化问题，它完全用Lagrange 乘数 α_i 表示。从Lagrange 对偶性（定义7.1）中我们知道，我们需要最大化对偶问题。这等价于最小化负对偶问题，从而我们得到对偶SVM

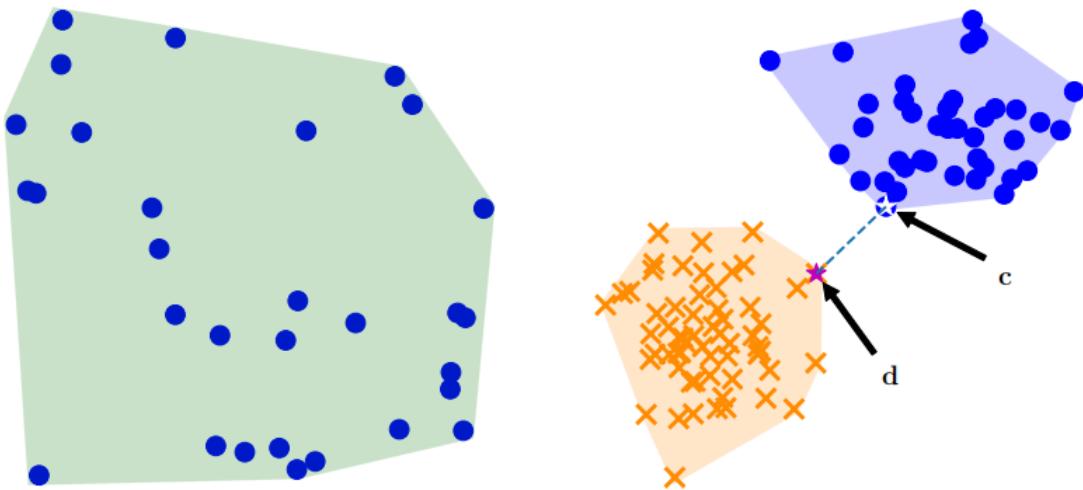
$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \sum_{i=1}^N \alpha_i \\ \text{subject to} \quad & \sum_{i=1}^N y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C \quad \text{for all } i = 1, \dots, N. \end{aligned}$$

(12.41)

(12.41) 中的等式约束是通过将 (12.36) 设置为零得到的。不等式约束 $\alpha_i \geq 0$ 是对不等式约束的Lagrange 乘数施加的条件（第7.2节）。不等式约束 $\alpha_i \leq C$ 在前面的段落中已讨论。

SVM中的不等式约束集被称为“盒约束”，因为它们将Lagrange 乘数的向量 $\alpha = [\alpha_1, \dots, \alpha_N]^\top \in \mathbb{R}^N$ 限制在每个轴上由0和 C 定义的盒子内。这些轴对齐的盒子在数值求解器中实现时特别高效（Dostál, 2009, 第5章）。

一旦我们获得了对偶参数 α ，我们就可以使用表示定理 (12.38) 来恢复原始参数 w 。让我们将最优原始参数称为 w^* 。但是，如何获得参数 b^* 仍然是一个问题。考虑一个正好位于边界上的样本 x_n ，即 $\langle w^*, x_n \rangle + b = y_n$ 。回想一下， y_n 要么是+1，要么是-1。因此，唯一未知的是 b ，它可以通过 <http://fouryears.eu/2012/06/07/the-svm-bias-term-conspiracy/>. 访问。



(a) Convex hull.

(b) Convex hulls around positive (blue) and negative (orange) examples. The distance between the two convex sets is the length of the difference vector $c - d$.

图12.9凸包。(a)凸包的点，其中一些位于边界内；凸出正例子和负例子的(b)凸壳。

12.3.2 双SVM：凸包视图

获得双SVM的另一种方法是考虑另一种几何论证。考虑具有相同标签的示例集 \mathbf{x}_n 。我们希望构建一个包含所有示例的凸集，并且这个凸集尽可能小。这被称为凸包，如图12.9所示。

首先，让我们对点的凸组合有一些直观的理解。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 以及一个权重 $\alpha_3 \geq 0$ ，使得 $\sum_{n=1}^3 \alpha_n = 1$ 时发生的情况。这三个点 x_1, x_2, x_3 的凸组合跨越了一个二维区域。这个区域的凸包是由每对点对应的边所形成的三角形。随着我们添加更多的点，并且点的数量大于维度数时，一些点将位于凸包内部，如图12.9(a)所示。

一般来说，构建凸包可以通过为每个示例 x_n 引入非负权重 $\alpha_n \geq 0$ 来完成。然后，凸包可以描述为集合

$$\text{conv}(\mathbf{X}) = \left\{ \sum_{n=1}^N \alpha_n \mathbf{x}_n \right\} \quad \text{with} \quad \sum_{n=1}^N \alpha_n = 1 \quad \text{and} \quad \alpha_n \geq 0,$$

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开的，则它们的凸包不会重叠。给定训练数据 $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ ，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 \mathbf{c} ，它位于正例集合的凸包内，并且最接近负类分布。类似地，我们在负例集合的凸包中选择一个点 \mathbf{d} ，它最接近正类分布；如图 12.9(b) 所示。我们定义 \mathbf{d} 和 \mathbf{c} 之间的差向量为

(12.44)

$$\mathbf{w} := \mathbf{c} - \mathbf{d}.$$

选择 \mathbf{c} 和 \mathbf{d} 点如前面所述，并要求它们彼此最接近，这相当于最小化 \mathbf{w} 的长度/范数，因此我们得到了相应的优化问题

$$\arg \min_{\mathbf{w}} \|\mathbf{w}\| = \arg \min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2.$$

(12.45)

由于 \mathbf{c} 必须在正凸包内，因此它可以表示为正例的凸组合，即对于非负系数 α_n^+

(12.46)

$$\mathbf{c} = \sum_{n:y_n=+1} \alpha_n^+ \mathbf{x}_n.$$

在(12.46)中，我们使用符号 $n : y_n = +1$ 来表示 $y_n = +1$ 的索引集 n 。类似地，对于具有负标签的示例，我们得到

(12.47)

$$\mathbf{d} = \sum_{n:y_n=-1} \alpha_n^- \mathbf{x}_n.$$

通过将(12.44)、(12.46)和(12.47)代入(12.45)，我们得到目标函数

$$\min_{\alpha} \frac{1}{2} \left\| \sum_{n:y_n=+1} \alpha_n^+ \mathbf{x}_n - \sum_{n:y_n=-1} \alpha_n^- \mathbf{x}_n \right\|^2.$$

(12.48)

设 $\boldsymbol{\alpha}$ 为所有系数的集合，即 $\boldsymbol{\alpha}^+$ 和 $\boldsymbol{\alpha}^-$ 的连接。回顾一下，我们要求每个凸包的系数之和为 1，即



$$\sum_{n:y_n=+1} \alpha_n^+ = 1 \quad \text{和} \quad \sum_{n:y_n=-1} \alpha_n^- = 1 .$$
(12.49)

这意味着存在约束

$$\sum_{n=1}^N y_n \alpha_n = 0 .$$

(12.50)

这个结果可以通过分别乘以每个类别的系数来观察:

(12.51a)

$$\begin{aligned} \sum_{n=1}^N y_n \alpha_n &= \sum_{n:y_n=+1} (+1) \alpha_n^+ + \sum_{n:y_n=-1} (-1) \alpha_n^- \\ &= \sum_{n:y_n=+1} \alpha_n^+ - \sum_{n:y_n=-1} \alpha_n^- = 1 - 1 = 0 . \end{aligned}$$

(12.51b)

目标函数 (12.48) 和约束 (12.50)，以及假设 $\alpha \geq 0$ ，共同构成了一个带约束的（凸）优化问题。可以证明，这个优化问题与对偶硬间隔SVM (Bennett 和 Bredensteiner, 2000a) 的优化问题是相同的。

备注：为了获得软间隔对偶，我们考虑缩减的凸包。缩减的凸包与凸包类似，但系数的大小有一个上限。 α 中元素的最大可能值限制了凸包可以取的大小。换句话说，对 α 的限制将凸包缩小到了一个更小的体积 (Bennett 和 Bredensteiner, 2000b)。

< 上一章节

下一章节 >

12.2 初级支持向量机

12.4 核函数



12.4 核函数

考虑对偶SVM的公式 (12.41)。注意到目标函数中的内积仅发生在样本 x_i 和 x_j 之间，而没有样本与参数之间的内积。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。这种模块化特性允许我们分别考虑分类方法(SVM)和特征表示 $\phi(x)$ 的选择，从而为我们独立探索这两个问题提供了灵活性。在本节中，我们将讨论表示 $\phi(x)$ 并简要介绍核函数的概念，但不涉及技术细节。

由于 $\phi(x)$ 可能是非线性函数，我们可以使用SVM（它假设了一个线性分类器）来构造在样本 x_n 上非线性的分类器。除了软间隔之外，这为用户处理非线性可分的数据集提供了第二条途径。事实证明，在对偶SVM中我们观察到的这种特性（即仅存在样本之间的内积）在许多算法和统计方法中都存在。我们不需要显式地定义一个非线性特征映射 $\phi(\cdot)$ 并计算样本 x_i 和 x_j 之间的内积，而是可以定义一个相似度函数 $k(x_i, x_j)$ 来表示 x_i 和 x_j 之间的关系。对于一类特定的相似度函数，称为核函数，这个相似度函数隐式地定义了一个非线性特征映射 $\phi(\cdot)$ 。核函数根据定义是函数 $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ ，其中存在一个希尔伯特空间 \mathcal{H} 和一个特征映射 $\phi : \mathcal{X} \rightarrow \mathcal{H}$ ，使得

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}.$$

(12.52)

与每个核函数 k 相关联的，都有一个唯一的再生核希尔伯特空间 (Aronszajn, 1950; Berlinet 和 Thomas-Agnan, 2004)。在这种独特的关联中， $\phi(x) = k(\cdot, x)$ 被称为规范特征映射。从内积到核函数的推广 (12.52) 被称为核技巧 (Schölkopf 和 Smola, 2002; Shawe-Taylor 和 Cristianini, 2004)，因为它隐藏了显式的非线性特征映射。

由数据集上的内积或应用 $k(\cdot, \cdot)$ 得到的矩阵 $K \in \mathbb{R}^{N \times N}$ 被称为Gram矩阵，通常简称为核矩阵。核函数必须是对称和正半定函数，以确保每个核矩阵 K 都是对称和正半定的 (第3.2.3节)：

$$\forall z \in \mathbb{R}^N : z^\top K z \geq 0.$$

(12.53)

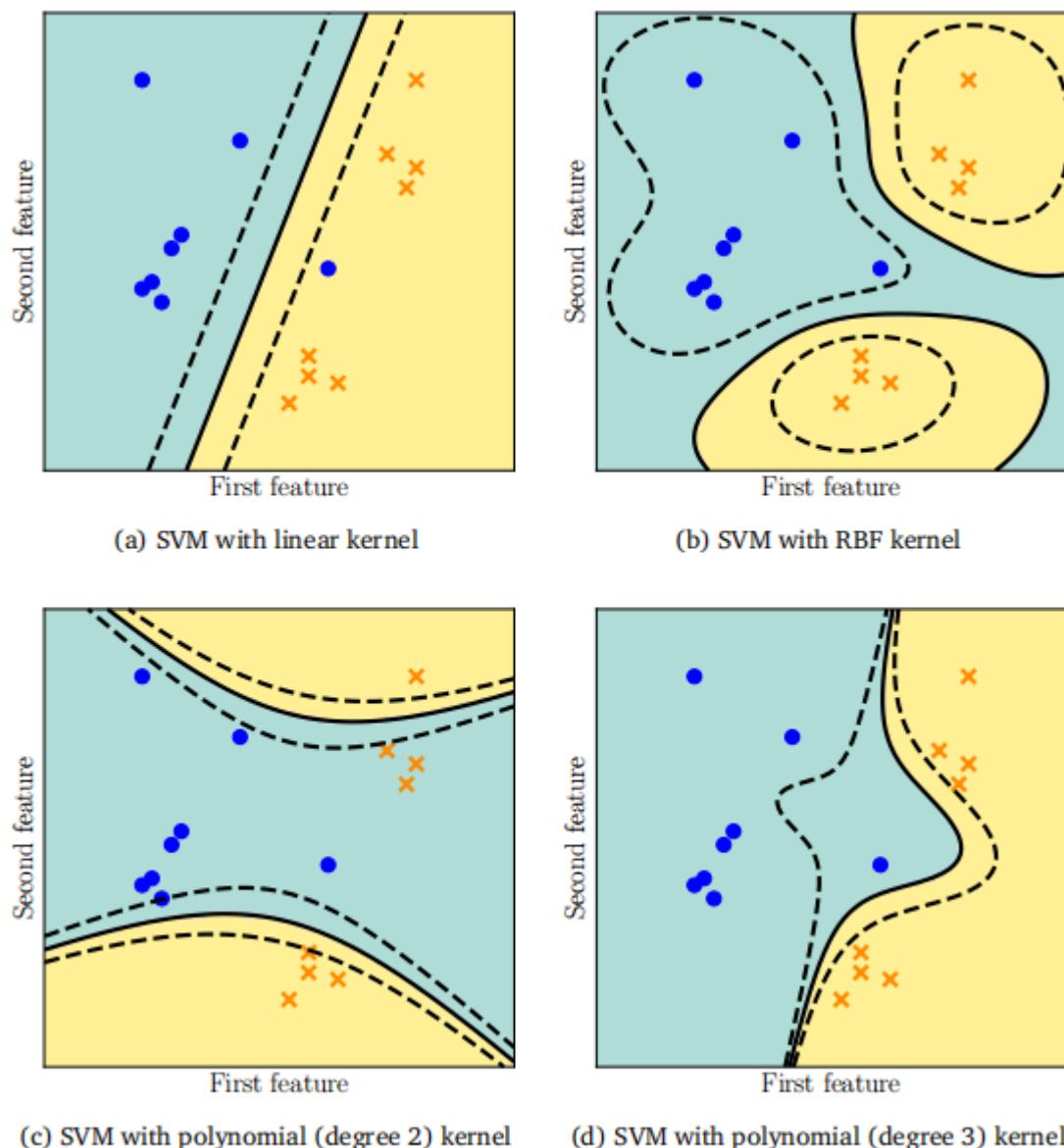


图12.10使用不同内核的SVM。注意，虽然决策边界是非线性的，但要解决的潜在问题是求解一个线性分离超平面（尽管有一个非线性核）。

对于多变量实值数据 $x_i \in \mathbb{R}^D$ ，一些流行的核函数示例包括多项式核、高斯径向基函数核和有理二次核（Schölkopf 和 Smola, 2002; Rasmussen 和 Williams, 2006）。图12.10展示了不同核函数在示例数据集上对超平面分隔效果的影响。请注意，我们仍然是在求解超平面，即函数假设类仍然是线性的。非线性曲面是由核函数引起的。

备注。对于初学者来说，不幸的是，“核”（kernel）这个词有多种含义。在本章中，“核”一词来源于再生核希尔伯特空间（RKHS）的概念（Aronszajn, 1950; Saitoh, 1988）。我们在线性代数中已经讨论过“核”的概念（第2.7.3节），在那里，“核”是零

空间的另一种说法。在机器学习中，“核”一词的第三个常见用途是核密度估计中的滑核（第11.5节）。

由于显式表示 $\phi(x)$ 在数学上与核表示 $k(x_i, x_j)$ 等价，因此从业者通常会设计核函数，使其计算效率高于显式特征映射之间的内积。例如，考虑多项式核（Schölkopf 和 Smola, 2002），当输入维度较大时，显式展开中的项数会迅速增长（即使是低次多项式）。核函数每输入一个维度只需进行一次乘法运算，这可以显著节省计算量。另一个例子是高斯径向基函数核（Schölkopf 和 Smola, 2002; Rasmussen 和 Williams, 2006），其对应的特征空间是无限维的。在这种情况下，我们无法显式地表示特征空间，但仍可以使用核来计算两个示例之间的相似性。核技巧的另一个有用方面是，原始数据不需要已经表示为多变量实值数据。请注意，内积是在函数 $\phi(\cdot)$ 的输出上定义的，但并不限制输入为实数。因此，函数 $\phi(\cdot)$ 和核函数 $k(\cdot, \cdot)$ 可以定义在任何对象上，例如集合、序列、字符串、图和分布（Ben-Hur et al., 2008; Gärtner, 2008; Shi et al., 2009; Sriperumbudur et al., 2010; Vishwanathan et al., 2010）。

< 上一章节

下一章节 >

12.3 对偶支持向量机

12.5 数值解



12.5 数值解

我们通过探讨如何根据第7章介绍的概念来表达本章中推导的问题，来结束对支持向量机（**SVMs**）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**的损失视角（8.2.2节），并将其表达为一个无约束优化问题。然后，我们将原始和对偶**SVM**的约束版本表达为标准形式的二次规划（7.3.2节）。

考虑**SVM**的损失函数视角（12.31）。这是一个凸无约束优化问题，但合页损失（12.28）不可微。因此，我们采用次梯度方法来解决它。然而，合页损失几乎在所有地方都是可微的，除了合页 $t = 1$ 处的单点。在这一点上，梯度是一个介于0和-1之间的可能值集。因此，合页损失的次梯度 g 由下式给出：

$$g(t) = \begin{cases} -1 & t < 1 \\ [-1, 0] & t = 1 \\ 0 & t > 1 \end{cases}$$

$$(12.54)$$
$$\begin{aligned} g(t) = & \begin{cases} -1 & t < 1 \\ [-1, 0] & t = 1 \\ 0 & t > 1 \end{cases} \\ & \text{使用这个次梯度，我们可以应用第7.1节中介绍的优化方法。} \end{aligned}$$

原始和对偶**SVM**都导致了凸二次规划问题（约束优化）。请注意，原始**SVM**（12.26a）中的优化变量具有输入示例维度 D 的大小。对偶**SVM**（12.41）中的优化变量具有示例数量 N 的大小。

为了将原始**SVM**表达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。我们重新排列原始**SVM**的方程（12.26a），使得优化变量都在右侧，并且约束的不等式与标准形式相匹配。这产生了以下优化问题：

$$(12.55)$$
$$\begin{aligned} & \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n \\ \text{subject to } & -y_n \mathbf{x}_n^\top \mathbf{w} - y_n b - \xi_n \leq -1 \\ & -\xi_n \leq 0 \end{aligned}$$

$n = 1, \dots, N$ 。通过将变量 w, b, \boldsymbol{x}_n 连接成一个单独的向量，并仔细收集项，我们得到软间隔SVM的以下矩阵形式：

$$\begin{aligned} & \min_{w, b, \boldsymbol{\xi}} \quad \frac{1}{2} \begin{bmatrix} w \\ b \\ \boldsymbol{\xi} \end{bmatrix}^\top \begin{bmatrix} \mathbf{I}_D & \mathbf{0}_{D, N+1} \\ \mathbf{0}_{N+1, D} & \mathbf{0}_{N+1, N+1} \end{bmatrix} \begin{bmatrix} w \\ b \\ \boldsymbol{\xi} \end{bmatrix} + [\mathbf{0}_{D+1, 1} \quad C\mathbf{1}_{N, 1}]^\top \begin{bmatrix} w \\ b \\ \boldsymbol{\xi} \end{bmatrix} \\ & \text{subject to } \begin{bmatrix} -\mathbf{Y}\mathbf{X} & -\mathbf{y} & -\mathbf{I}_N \\ \mathbf{0}_{N, D+1} & & -\mathbf{I}_N \end{bmatrix} \begin{bmatrix} w \\ b \\ \boldsymbol{\xi} \end{bmatrix} \leq \begin{bmatrix} -\mathbf{1}_{N, 1} \\ \mathbf{0}_{N, 1} \end{bmatrix}. \end{aligned}$$

在前面的优化问题中，最小化是针对参数 $[w^\top, b, \boldsymbol{\xi}^\top]^\top \in \mathbb{R}^{D+1+N}$ 进行的，我们使用的符号包括： \mathbf{I}_m 表示大小为 $m \times m$ 的单位矩阵， $\mathbf{0}_{m,n}$ 表示大小为 $m \times n$ 的零矩阵， $\mathbf{1}_{m,n}$ 表示大小为 $m \times n$ 的全1矩阵。此外， \mathbf{y} 是标签向量 $[y_1, \dots, y_N]^\top$ ， $\mathbf{Y} = \text{diag}(\mathbf{y})$ 是一个 $N \times N$ 的对角矩阵，其对角线元素来自 \mathbf{y} ，且 $\mathbf{X} \in \mathbb{R}^{N \times D}$ 是通过连接所有示例获得的矩阵。

我们同样可以对支持向量机（SVM）的对偶版本（12.41）中的项进行一系列收集。为了将对偶SVM表达为标准形式，我们首先需要表示核矩阵 \mathbf{K} ，使得其每个元素为 $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ 。如果我们有明确的特征表示 \mathbf{x}_i ，则我们定义 $K_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$ 。为了方便表示，我们引入一个矩阵，其所有元素均为零，除了对角线上存储标签的位置，即 $\mathbf{Y} = \text{diag}(\mathbf{y})$ 。对偶SVM可以表示为

$$\begin{aligned} & \min_{\boldsymbol{\alpha}} \quad \frac{1}{2} \boldsymbol{\alpha}^\top \mathbf{Y} \mathbf{K} \mathbf{Y} \boldsymbol{\alpha} - \mathbf{1}_{N, 1}^\top \boldsymbol{\alpha} \\ & \text{subject to } \begin{bmatrix} \mathbf{y}^\top \\ -\mathbf{y}^\top \\ -\mathbf{I}_N \\ \mathbf{I}_N \end{bmatrix} \boldsymbol{\alpha} \leq \begin{bmatrix} \mathbf{0}_{N+2, 1} \\ C\mathbf{1}_{N, 1} \end{bmatrix}. \end{aligned}$$

(12.57)

备注。在7.3.1和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

(12.58)

$$Ax = b \quad \text{被替换为} \quad Ax \leq b \quad \text{和} \quad Ax \geq b.$$

凸优化方法的特定软件实现可能提供了表达等式约束的能力。



由于SVM有许多不同的可能视角，因此解决由此产生的优化问题也有许多方法。这里介绍的方法，即将SVM问题表达为标准凸优化形式，在实践中并不常用。SVM求解器的两个主要实现是Chang和Lin（2011）（开源）以及Joachims（1999）。由于SVM具有清晰且定义良好的优化问题，因此可以应用许多基于数值优化技术（Nocedal和Wright, 2006）的方法（Shawe-Taylor和Sun, 2011）。

< 上一章节

下一章节 >

12.4 核函数

12.6 拓展阅读



本教程由 [Datawhale 开源社区](#) 编译，与对应的英文原版均开源免费

12.6 拓展阅读

支持向量机（SVM）是研究二分类问题的众多方法之一。其他方法包括感知机、逻辑回归、费舍尔判别分析、最近邻、朴素贝叶斯和随机森林（Bishop, 2006; Murphy, 2012）。Ben-Hur et al. (2008) 的文献中提供了关于SVM和离散序列上核的简短教程。SVM的发展与第8.2节中讨论的经验风险最小化密切相关，因此SVM具有强大的理论特性（Vapnik, 2000; Steinwart和Christmann, 2008）。关于核方法的书籍（Schölkopf和Smola, 2002）详细介绍了支持向量机的许多细节以及如何优化它们。另一本关于核方法的更广泛的书籍（Shawe-Taylor和Cristianini, 2004）也包含了许多针对不同机器学习问题的线性代数方法。

利用勒让德-芬切尔变换（Legendre-Fenchel transform，第7.3.3节）的思想，可以得到对偶SVM的另一种推导。该推导分别考虑了SVM无约束形式（12.31）的每一项，并计算了它们的凸共轭（Rifkin和Lippert, 2007）。对SVM的功能分析视角（也是正则化方法视角）感兴趣的读者可以参考Wahba (1990) 的工作。核的理论阐述（Aronszajn, 1950; Schwartz, 1964; Saitoh, 1988; Manton和Amblard, 2015）需要线性算子基础知识（Akhiezer和Glazman, 1993）。核的概念已被推广到巴拿赫空间（Banach spaces）（Zhang et al., 2009）和克列因空间（Kreĭn spaces）（Ong et al., 2004; Loosli et al., 2016）。

请注意，合页损失函数有三种等价表示，如（12.28）和（12.29）所示，以及（12.33）中的约束优化问题。在将SVM损失函数与其他损失函数进行比较时，（12.28）式经常被使用（Steinwart, 2007）。（12.29）式的两段形式便于计算次梯度，因为每段都是线性的。第12.5节中看到的第三种形式（12.33）使得能够使用凸二次规划（第7.3.2节）工具。

由于二分类是机器学习中研究得很好的一项任务，因此有时也会使用其他术语，如判别、分离和决策。此外，二分类器的输出可以是三个量之一。首先是线性函数本身的输出（通常称为分数），它可以取任何实数值。这个输出可以用于对示例进行排名，而二分类可以被认为是在排名后的示例上选择一个阈值（Shawe-Taylor和Cristianini, 2004）。第二个经常被认为是二分类器输出的是，在通过非线性函数传递后确定的输出，以将其值限制在有界范围内，例如在区间[0,1]内。一个常见的非线性函数是



Sigmoid函数（Bishop, 2006）。当非线性结果得到良好校准的概率（Gneiting和Raftery, 2007; Reid和Williamson, 2011）时，这被称为类别概率估计。二分类器的第三个输出是最终的二值决策{+1,-1}，这是最常假设为分类器输出的形式。

SVM是一种二分类器，它本身并不自然地适合概率解释。有几种方法可以将线性函数的原始输出（分数）转换为校准后的类别概率估计 ($P(Y = 1|X = x)$)，这些方法涉及一个额外的校准步骤（Platt, 2000; Zadrozny和Elkan, 2001; Lin et al., 2007）。从训练的角度来看，有许多相关的概率方法。在第12.2.5节的末尾，我们提到损失函数和似然之间存在关系（也请比较第8.2节和第8.3节）。在训练过程中，与良好校准的变换相对应的最大似然方法称为逻辑回归，它来自一类称为广义线性模型的方法。从这个角度来看，逻辑回归的详细信息可以在Agresti（2002，第5章）和McCullagh和Nelder（1989，第4章）中找到。当然，可以通过使用贝叶斯逻辑回归估计后验分布来更贝叶斯地看待分类器输出。贝叶斯视角还包括先验的规范，其中包括与似然相关的设计选择，如共轭性（第6.6.1节）。此外，还可以将潜在函数视为先验，这导致了高斯过程的分类(Rasmussen and Williams, 2006, chapter 3).

« 上一章节

12.5 数值解