

数据科学与工程 数学基础

(第1版)
黄定江 编著

草稿请勿外传

华东师范大学
上海

内 容 简 介

本书介绍了数据科学、人工智能和机器学习领域所需的核心数学基础知识，涉及矩阵计算、概率和信息论基础、优化基础。内容按照从模式分析到数据分析再到数学基础的思路来组织，围绕数据分析系统的核心构成：表示、模型和学习形成数据线和数学线两条线。数据线按照数据分析的处理流程、通过大量翔实的案例作为导引，引出所需数学；数学线紧扣数据线，按照知识内容发生的内在自然逻辑顺序展开。两者相辅相成，构成从具体到抽象、从抽象到具体的闭环。本书在数据科学的定位类似于《离散数学》在计算机科学的定位，配有相当数量的习题，可作为数据科学与大数据技术、人工智能、计算机科学和软件工程等相关专业的本科生或研究生的数学基础课程教材或参考书，也可作为学术和工业界科技人员了解和应用数据科学与大数据技术数学基础的参考手册。

传外勿请稿草

前言

本书主要介绍数据科学、机器学习和人工智能所依赖的数学基础，包括：线性代数、概率与信息论和优化理论。我们知道数据的表示需要向量，机器学习中函数模型的权重可以用矩阵来表示；数据中的不确定性或随机性描述通常由概率来刻画，大数定律为统计机器学习模型的成功提供了理论基础；而优化为最终训练出一套可靠的模型参数提供了强大的数值计算支撑。

尽管线性代数、概率与信息论和优化理论的很多内容研究已经持续了一个世纪以上，但是直到近二十多年来人们才发现它们已然成为数据科学建模求解的核心数学基础，比如，奇异值分解的广泛应用、最大似然和最大后验的成功运用、凸优化方法可靠和迅速的求解等等，使得这些理论和方法足以嵌入到基于计算机程序运行的数据分析和人工智能算法设计之中。

但是就像很多其它学科利用线性代数、概率统计和优化作为基础工具一样，现实世界的数据问题如何转换为一个线性代数计算或概率估计或优化求解问题是不容易的，特别是数据科学领域的问题与其它领域的不同之处在于它对这三部分知识的需求是如此的交错复杂和浑然一体，比如，矩阵既可以用于表示数据，但它也是函数模型变换的一部分；协方差矩阵巧妙的融合了概率和线性代数，把方差和矩阵捏合在一起，从而能用作主成分分析的建模对象；数据科学大部分优化问题是非凸的，判断它是不是凸的或者将某个问题表述为凸优化的形式是比较困难的，这可以部分地借助对称正半定矩阵的概念等来实现。

本书目的

因此，本书的主要目的是帮助读者快速理清和掌握数据科学、机器学习和人工智能领域所需的相关数学知识，即表示数据所需的向量和矩阵的概念与运算，以及数值线性代数的四大核心议题；构建数据概率模型所需的概率基础和相关的统计和信息论准则；判断、描述以及求解凸优化问题的方法和背景知识等等。全书包括四个部分，共 12 章内容。

第一部分：绪论。也即第 1 章，主要介绍数据科学与工程数学基础在数据科学与大数据技术专业中的定位、应用背景、服务学科领域和主要数学内容的构成以及相关的数学基础简史，使读者对本书有初步的了解。这一章，我们会对从图像感知到自然语言处理再到数据分析与机器学习做一个简要的概览，让读者能够从“应用驱动”的角度来了解数据科学所涉及的和所需的数据基础，为全书的数据案例和数学内容展开做好铺垫。

第二部分：数据的低维表示——矩阵分析。涵盖了从第 2 章到第 6 章的主要内容。

第 2 章主要按数据的向量和矩阵表示、数据的向量和矩阵空间、数据空间的关系以及数据空间上代数结构建立的过程来具体介绍数据科学与工程所涉及的向量和矩阵的计算所需的基本知识，包括向量和矩阵基本概念和运算、向量空间、线性映射和线性变换、矩阵的基本特征和矩阵的特征分解等。

第 3 章介绍了线性代数的几何：度量和投影，包括向量的范数和内积、矩阵的范数和内积、矩阵的四个基本子空间、投影以及特殊的正交矩阵等。这些概念有助于我们从几何的角度来理解线性代数的基本概念以及在数据科学中的应用。如范数和内积将被用作定义数据的各种相似性度量，以及防止数据模型过拟合的正则化手段；投影既是一个几何量，也是一个变换，在数据科学的降维任务中具有本质的作用。

第 4 章介绍了五种常用的矩阵分解方法，包括 LU（三角）分解、QR（正交）三角分解、谱（特征）分解、Cholesky 分解和奇异值分解等。线性代数包含很多有趣的矩阵，如：对角阵、三角矩阵、正交矩阵、对称矩阵、置换矩阵、投影矩阵和关联矩阵等等。在这些矩阵当中对称正（半）定矩阵是核心，因为数据科学与机器学习中大部分矩阵都是非方阵，而非方阵总是可以通过与其自身的转置相乘得到对称正（半）定矩阵。对称正（半）定矩阵有正（非负）的特征值，并且有正交的特征向量，它也可以表示成一些秩 1 矩阵的线性组合，因此可以方便的用于做低秩近似计算。在机器学习中，我们主要处理的是这些大规模的对称正定矩阵或复杂的非方阵矩阵，需要借助矩阵分解的技术，特别是奇异值分解，把它表示为对角阵、三角阵和正交矩阵的乘积等等，然后利用这些特殊且简单的矩阵实现复杂矩阵的特征值等矩阵基本特征的快速计算，并用于数据压缩、数据降维以及矩阵低秩近似问题的求解等等，这对帮助理解原本复杂的高维数据矩阵的结构和性质具有重要的作用。

第 5 章介绍了数值线性代数三大核心主题内容，包括线性方程组的求解、最小二乘问题和特征值的求解。数据科学中的很多问题最终都归结为线性方程组的求解，因此这一章主要介绍线性方程组的类型和解的结构，引入基于矩阵分解的线性方程组和最小二乘问题的求解方法，并讨论解的敏感性，这些内容将与后续优化问题求解、数据科学中的线性回归问题相联系。此外，还介绍了大规模矩阵求解特征值的一些计算方法，包括幂迭代法，这已被广泛应用于数据科学中的搜索技术 pagerank 的矩阵特征值计算。

第 6 章主要介绍向量和矩阵微分。包括向量和矩阵函数，以及数据科学和统计机器学习中常见的各种函数（包括模型函数、损失函数和目标函数等）、深度神经网络中函数的构造（包括模型函数和激活函数等），梯度和高阶导数的定义和性质、向量值函数和矩阵函数的梯度和求解方法以及用迹微分法求梯度的方法，并引入深度网络中的反向传播和自动微分求解方法。这一章介绍的函数模型是数据科学中两大类型的模型之一。这些内容将在优化方法介绍和数据科学中的各种优化问题求解中反复使用。

第三部分：数据的随机表示——概率和信息论。涵盖了从第 7 章至第 9 章的内容。

第 7 章回顾概率论的基本概念，建立用随机变量和分布来描述数据中的不确定性的思想。包括概率论的基本概念、随机变量及其分布、随机变量的数字特征、概率不等式、大数定律和中

心极限定理、随机过程初步等。其中，概率不等式在机器学习的理论分析，通常也称为计算学习理论，如 PAC 可学习性以及算法的泛化界和收敛性分析等方面具有重要的应用。此外，大数据定律将被推广用于统计学习理论中经验风险最小化准则的建立。

第 8 章介绍香农熵、信息熵、KL 散度和微分熵等信息论基本概念和性质，并引入基于熵概念的信息度量准则和数据科学建模原理。信息论与机器学习有着紧密的联系，学习某种意义上就是一个熵减的过程，学习的过程也就是使信息的不确定度下降的过程，因此这些内容可以用于创造和改进学习算法（主要是分类问题），甚至衍生出了一个新方向——信息理论学习。特别可用于数据科学中基于概率和熵的相似性度量，这与第 3 章中非概率的相似性度量形成对应。

第 9 章介绍概率模型。包括数据建模的概率思想、模型的参数估计和非参数估计、概率模型的图语言描述和统计决策理论。其中数据建模的概率思想将引出数据科学和机器学习中模型的概率表示和类型等；模型的参数估计和非参数估计重点介绍极大似然、极大后验、直方图估计、核密度估计和非参回顾估计等；概率模型的图语言描述将给出条件独立性、有向非循环图、无向图、团和势等，这为以后学习朴素贝叶斯、隐马尔科夫等概率图模型内容奠定基础；统计决策理论主要涉及模型参数估计的好坏判断，这与机器学习中建立模型的策略密切相关。这一章介绍的概率模型是数据科学中另一大类型的模型之一，与第 6 章中的非概率模型，也即函数模型形成对应。

第四部分：数据的数值优化——凸优化。也即第 10 章至第 12 章的内容。

第 10 章介绍优化的基础理论。包括优化问题的分类，凸集和凸函数的定义和判别方法以及保凸运算，引入凸优化问题的定义和标准形式，并介绍数据科学和机器学习中常见的典型优化问题。事实上，机器学习中通过经验风险最小化准则建立的很多问题都可以建模为凸优化问题。

第 11 章介绍拉格朗日对偶函数和拉格朗日对偶问题，把标准形式（可能是非凸）的优化问题转化为对偶问题进行求解；介绍凸优化的最优性条件；介绍数据科学中各种常见的优化问题的对偶性问题。

第 12 章介绍无约束优化问题的性质和求解方法，包括直线搜索、梯度下降、最速下降、随机梯度下降方法等零阶和一阶方法；约束优化问题的求解方法，包括可行函数法和罚函数法；凸优化问题求解的高阶算法，包括牛顿法、内点法和拟牛顿法等二阶方法以及深度学习中一些常见的优化技术。这些方法将用于数据科学与机器学习中各种优化问题的求解。

读者范围

本书主要面向“数据科学与大数据技术”、“人工智能”、“计算机科学”等专业的本科生或低年级研究生。对于在工作中需要用到数值线性代数、概率估计和数学优化，或者更一般地说，用到计算数学的科研人员、科学家以及工程师，本书也较为合适。这些人群包括直接从事数据分析、机器学习和人工智能算法的科技工作者，亦包括一些工作在其他科学和工程领域但是需要借助数据科学数学基础的科技工作者，这些领域包括计算科学、经济学、金融、统计学、数据挖掘等。在阅读本书之前，读者只需要掌握现代微积分的基础知识即可。如果读者对一些基本的线性代数和基本的概率论有一定的了解，应能较好地理解本书的所有论证和讨论。当然，我

们希望即使没有学过线性代数和概率论的读者也能够理解本书所有的基本思想和要点。

使用本书作为教材

我们希望本书能够在不同的课程中作为基本教材或者是参考教材来发挥它的作用，这些课程包括数据科学与工程的数学基础、人工智能的数学基础、机器学习的数学基础和计算机科学的数学基础（偏应用）等。从 2018 年开始，我们即在华东师范大学数据学院的本科生和低年级研究生的同名课程中使用本书的初稿。我们的经验表明，用 3 个学分，也即 48 学时到 54 学时，可以粗略讲授本书的大部分内容。如果用一个 4 学分的课程时间，也即 64 学时到 72 学时，讲课进度就可以比较从容，也可以增加更多的例子，并且可以更加详尽地讨论有关理论。若能用 5 个学分的课程时间，就可以对奇异值分解、最小二乘问题、特征值的计算、线性规划和二次规划（对于以应用为目的的学生极为重要）这些基本内容进行较广泛的细致讨论，或者加强这些内容对应算法方面的介绍或对学生布置更多的习题训练。本书可以作为线性代数、概率统计、线性优化和非线性优化等基础的参考读物。此外，对于像数学系更关注理论的课程，本书可以作为辅助教材，它提供了一些简单的实际例子。

致谢

本书是在华东师范大学周傲英副校长和数据科学与工程学院的大力支持下历时两年多完成的，虽然两年的时间不算短，并且主要内容也在华东师范大学数据学院本科生和低年级研究生的同名课程中使用过并取得了不错的讲授和学习效果，但是作为为“数据科学与大数据技术”这样一个崭新的硬专业提供一本适用的教材，这点时间显然是不够的，很多内容还没有得到很好的打磨以适应不同层次水平的学生或相关的科研人员。但我们还是希望能够快速出版以满足日益增长的专业需求和读者们对这一领域持续探索的热情。我们只能期待在使用的过程中不断获得反馈以便快速迭代，从而获得更广泛的使用普遍性。这正如数据科学、人工智能和计算机科学这一领域从业者的行事准则：上线、迭代更新、再迭代，…，直至打磨稳定。我们也计划采用这种方式，所以恳请读者们如果碰到任何书本有关的问题，能及时反馈给我们 djhuang@dase.ecnu.edu.cn，以便我们能够改进，我们将不吝感激。

本书的写作过程中得到了来自华东师范大学、北京大学、中国人民大学、中山大学、北京理工大学、东北大学、西北工业大学以及河南大学等 15 所高校组成的数据专业协作组以及高等教育出版社和华东师范大学出版社的专家们的反馈和建议，同时也获得了华东师范大学很多同事，我课题组的研究生们以及我课程上的学生们的反馈和建议。篇幅所限，我们无法一一表达我们的感谢，只能在此对大家一并表达诚挚的谢意。

最后要特别感谢我的课题组的研究生们，我的博士生郝珊锋、申弋斌、刘友超和硕士生唐贊喆、赖叶静、张洋、余若男、汤路民、杨康、周雪茗、杨礼孟、王明和李特等同学，他们花费了很多时间来协助我一起修改、编辑书中的公式、表格和图片等，才使得本书能够快速面世。郝珊锋和唐贊喆也协助我一起制作了与本书配套的同名课程的 MOOC 视频（本课程在融优学堂和超星泛雅 <http://mooc1.chaoxing.com/course/208843967.html> 上线），感谢他们的努力付出。

由于水平有限，书中难免有不妥和错误之处，欢迎读者批评指正。

草稿勿外传

数学符号

下面简要介绍本书所使用的数学符号。如果你不熟悉数学符号所表示的数学概念，可以参考对应的章节。

向量和矩阵

a	标量 (整数或实数)
\mathbf{a}	向量
A	矩阵
\mathbf{A}	张量
I_n	n 行 n 列的单位矩阵
\mathbf{I}	维度蕴含于上下文的单位矩阵
$e^{(i)}$	索引 i 处值为 1 其它值为 0 的标准基向量
$\text{diag}(\mathbf{a})$	对角方阵，其中对角元素由 \mathbf{a} 给定
a	标量随机变量
\mathbf{a}	向量随机变量
\mathbf{A}	矩阵随机变量
a_i	向量 \mathbf{a} 的第 i 个元素，其中索引从 1 开始
a_{-i}	除了第 i 个元素, \mathbf{a} 的所有元素
$a_{i,j}$	矩阵 \mathbf{A} 的 i 行 j 列元素
$A_{i,:}$	矩阵 \mathbf{A} 的第 i 行
$A_{:,i}$	矩阵 \mathbf{A} 的第 i 列
$A_{i,j,k}$	3 维张量 \mathbf{A} 的 (i,j,k) 元素
$A_{::i}$	3 维张量的 2 维切片
a_i	随机向量 \mathbf{a} 的第 i 个元素
\mathbb{A}	集合
\mathbb{R}	实数集
\mathbb{C}	复数域集
$\{0, 1\}$	包含 0 和 1 的集合
$\{0, 1, \dots, n\}$	包含 0 和 n 之间所有整数的集合
$(a, b]$	不包含 a 但包含 b 的实数区间
$\mathbb{A} \setminus \mathbb{B}$	差集，即其元素包含于 \mathbb{A} 但不包含于 \mathbb{B}

草稿勿外传

\mathbb{R}^n	n 维实向量空间
\mathbb{C}^n	n 维复向量空间
$\dim(\mathbb{V})$	空间 \mathbb{V} 的维数
\mathbf{A}^{-1}	矩阵的逆
\mathbf{A}^\top	矩阵 \mathbf{A} 的转置
\mathbf{A}^\dagger	\mathbf{A} 的 Moore-Penrose 伪逆
$\mathbf{A} \odot \mathbf{B}$	\mathbf{A} 和 \mathbf{B} 的逐元素乘积 (Hadamard 乘积)
$ \mathbf{A} $	\mathbf{A} 的行列式
$\text{rank}(\mathbf{A})$	矩阵的秩
A_{ij}	元素 a_{ij} 的代数余子式
\mathbf{A}^*	\mathbf{A} 的伴随矩阵
$\text{Tr}(\mathbf{A})$	矩阵的迹
λ	矩阵的特征值
范数	
$\ \cdot\ $	范数
$\ \mathbf{x}\ _2$	向量的 l_2 范数
$\ \mathbf{x}\ _1$	向量的 l_1 范数
$\ \mathbf{x}\ _\infty$	向量的 l_∞ 范数
$\ \mathbf{X}\ _2$	矩阵 \mathbf{X} 的谱范数
$\text{sim}_{\cos}(\mathbf{x}, \mathbf{y})$	余弦相似度
$\text{vec}(\mathbf{A})$	矩阵的向量化
$\ \mathbf{A}\ _F$	矩阵的 F 范数
$\ \mathbf{A}\ _*$	矩阵的核范数
$\text{Col}(\mathbf{A})$	\mathbf{A} 的列空间
$\text{Row}(\mathbf{A})$	行空间
$\text{Null}(\mathbf{A})$	零空间
$\text{Null}(\mathbf{A}^\top)$	左零空间
微分	
$\frac{dy}{dx}$	y 关于 x 的导数
$\frac{\partial y}{\partial x}$	y 关于 x 的偏导
$\nabla_{\mathbf{x}} y$	y 关于 \mathbf{x} 的梯度
$\nabla_{\mathbf{X}} y$	y 关于 \mathbf{X} 的矩阵导数
$\nabla_{\mathbf{X}} y$	y 关于 \mathbf{X} 求导后的张量
$\frac{\partial f}{\partial x}$	$f : R^n \rightarrow R^m$ 的 Jacobian 矩阵 $J \in R^{m \times n}$

$\nabla_{\mathbf{x}}^2 f(\mathbf{x}) \mathbf{H}(f)(\mathbf{x})$	f 在点 \mathbf{x} 处的 Hessian 矩阵
	数据集
\mathbb{X}	输入空间
\mathbb{Y}	输出空间
$\mathbf{x} \in \mathbb{X}$	输入, 实例
$y \in \mathbb{Y}$	输出, 标记
$T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$	训练数据集
N	样本容量
(\mathbf{x}_i, y_i)	第 i 个训练数据点
$\mathbf{x} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)})^T$	输入向量, n 维实数向量
$\mathbf{x}_i^{(j)}$	输入向量 \mathbf{x}_i 的第 j 分量
	概率基础
Ω	样本空间
E	随机试验
A	事件
$P(A)$	事件 A 发生的概率
$P(B A)$	事件 A 发生的情况下, 事件 B 发生的概率
$F_X(x)$	累积分布函数 CDF
$f_X(x)$	概率密度函数
x^+	从右边趋向于 x
$E(X)$	随机变量 X 的期望
$D(X)$	随机变量 X 的方差
$Cov(X, Y)$	随机变量 X, Y 的协方差
$N(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	均值为 $\boldsymbol{\mu}$ 协方差为 $\boldsymbol{\Sigma}$, \mathbf{x} 的高斯分布
$\Phi_X(t)$	X 的矩母函数, t 为实数
$\mathbb{E}_{x \sim P}[f(x)]$	$f(x)$ 关于 $P(x)$ 的期望
$X_n \xrightarrow{P} X$	X_n 依概率收敛于 X
$X_n \rightsquigarrow X$	X_n 依分布收敛于 X
$X_n \xrightarrow{qm} X$	X_n 均方意义上收敛于 X
$\Phi(z)$	标准正态分布的累积分布函数
$R_{exp}(f)$	期望风险
$R_{emp}(f)$	经验风险
	信息论基础
$I(x_i)$	事件 x_i 的自信息

草稿勿外传

草稿勿外传

$I(x_i; y_i)$	事件 x_i 和事件 y_i 的互信息
$H(X)$	随机变量 X 的信息熵
$p(x_i)$	事件 x_i 的概率分布
$H(\mathbf{p})$	熵函数
$D_{KL}(P\ Q)$	P 和 Q 的 KL 散度
$H(P, Q)$	P 和 Q 交叉熵
$h(X)$	连续随机变量 X 的微分熵
概率模型和参数估计	
θ	待估参数
Θ	待估参数可能的取值
$L(\theta)$	样本在参数 θ 下的似然函数
$f(x; \theta)$	由 θ 参数化, 关于 x 的函数
$p(\mathcal{D} \theta)$	由 θ 参数化, 获得给定数据 \mathcal{D} 的概率
$l(\theta)$	对数似然函数
$\mathbf{1}_{condition}$	如果条件为真则为 1, 否则为 0
$K(u)$	参数为 u 的核函数
$\Gamma(x)$	伽马函数
$Beta(a, b)$	Beta 函数
$Pa(x_i)$	随机变量 x_i 的父节点
$\Phi_C(x_C)$	团的势函数, 变量 x_C 属于集合 C
$NLL(\theta)$	模型参数为 θ 的极小负 log 似然损失
ω	权重向量
优化	
$\mathbf{aff}C$	集合 C 的仿射包
$\mathbf{relint}C$	集合 C 的相对内部
$\mathbf{conv}C$	集合 C 的凸包
$\mathbf{int}C$	集合 C 的内部
$\mathbf{cl}C$	集合 C 的闭包
$\mathbf{bd}C$	集合 C 的边界: $\mathbf{bd}C = \mathbf{cl}C \setminus \mathbf{int}C$
I_C	集合 C 的示性函数
S_C	集合 C 的支撑函数
$x \preceq y$	向量 x 和 y 之间的分量不等式
S^n	对称的 $n \times n$ 矩阵
S_+^n, S_{++}^n	对称半正定、正定 $n \times n$ 矩阵

$\mathbb{R}_+, \mathbb{R}_{++}$	非负、正实数
$\text{epi } f$	函数 f 的上镜图
$\text{prob} S$	事件 S 的概率
$\text{dom } f$	函数 f 的定义域
$\lambda_{\max}(X), \lambda_{\min}(X)$	对称矩阵 X 的最大、最小特征值
$dist(A, B)$	集合（或点） A 和 B 之间的距离
∇f	函数的导数
f^*	f 的共轭函数
$H(\mathbf{x})$	$f(\mathbf{x})$ 在点 \mathbf{x} 处的 Hessian 矩阵

传
外
勿
请
稿
草

目录

第一章 绪论	1
1.1 本教材产生的背景和定位	1
1.2 从图像感知到自然语言处理	4
1.2.1 猫、分类和神经网络	5
1.2.2 文本、词向量和朴素贝叶斯	11
1.3 从数据分析到数学基础	18
1.3.1 数据分析和机器学习概览	19
1.3.2 数据	22
1.3.3 模型	26
1.3.4 学习	29
1.3.5 机器学习的应用	35
1.4 数据分析和机器学习所需数学内容框架	37
1.5 数据科学与工程数学的历史	39
1.5.1 早期阶段：线性代数的诞生	39
1.5.2 概率论的起源	40
1.5.3 优化作为理论工具	40
1.5.4 数值线性代数的出现	40
1.5.5 线性和二次规划的出现	41
1.5.6 凸规划的出现	41
1.5.7 现阶段	42
1.6 本教材的使用建议	42
第二章 向量和矩阵基础	47
2.1 向量与矩阵的概念与运算	48
2.1.1 向量与矩阵的基本概念：数据表示的观点	48

外传
读稿请

2.1.2 向量的运算	53
2.1.3 矩阵的运算	53
2.1.4 线性方程组	58
2.2 向量空间	61
2.2.1 向量空间的基本概念：数据处理空间的出发点	61
2.2.2 向量子空间	63
2.2.3 子空间的交、和、直和	64
2.2.4 线性无关性	66
2.2.5 生成集、基底与坐标	68
2.2.6 秩	72
2.2.7 仿射空间	74
2.3 线性映射与线性变换	75
2.3.1 映射	76
2.3.2 线性映射：线性模型的观点	78
2.3.3 线性映射的矩阵表示	81
2.3.4 线性变换	87
2.3.5 仿射映射	91
2.4 矩阵的基本特征	93
2.4.1 行列式	94
2.4.2 迹运算	99
2.4.3 对称矩阵与二次型	100
2.4.4 特征值与特征向量	105
2.5 阅读材料	109
第三章 度量与投影	115
3.1 内积与范数：数据度量的观点	116
3.1.1 向量范数、长度与距离	117
3.1.2 l_p 范数	118
3.1.3 范数的几何意义与性质	120
3.1.4 内积与夹角	122
3.1.5 数据科学中常用的相似性度量 I	128
3.1.6 矩阵的内积与范数	133
3.1.7 范数在机器学习中的应用	139
3.2 正交与投影	142
3.2.1 矩阵的四个基本子空间	143

3.2.2 四个基本子空间的正交性	147
3.2.3 正交投影	149
3.3 正交基与 Gram-Schmidt 正交化	155
3.3.1 标准正交基	155
3.3.2 Gram-Schmidt 正交化	155
3.4 具有特殊结构和性质的矩阵	157
3.4.1 特殊的正交变换矩阵——旋转	157
3.4.2 反射矩阵	162
3.4.3 信号处理中常见的正交矩阵	165
3.5 阅读材料	172

第四章 矩阵分解 179

4.1 数学中常见的具有特殊结构的矩阵	180
4.2 数据科学中常见的矩阵	185
4.2.1 图的基本概念回顾	187
4.2.2 有向图相关的矩阵	188
4.2.3 无向图相关的矩阵	189
4.2.4 加权图相关的矩阵	190
4.2.5 稀疏矩阵	193
4.3 LU 分解	200
4.3.1 LU 分解	200
4.3.2 选主元的 LU 分解	204
4.4 QR 分解	207
4.4.1 QR 分解	207
4.4.2 基于 Gram-Schmidt 正交化的 QR 分解	208
4.4.3 基于 Householder 变换的 QR 分解	213
4.4.4 基于 Givens 旋转的 QR 分解	217
4.5 谱分解与 Cholesky 分解	223
4.5.1 对称矩阵的谱分解	223
4.5.2 正半定矩阵与 Cholesky 分解	229
4.5.3 改进的 Cholesky 分解	230
4.6 奇异值分解	231
4.6.1 引例	232
4.6.2 奇异值分解	232
4.6.3 奇异值分解的几何解释	233

传
外
人
情
稿
草

情
稿
草
稿
外
傳

4.6.4	紧奇异值分解和截断奇异值分解	235
4.6.5	奇异值分解基本定理	236
4.6.6	奇异值分解的计算	238
4.6.7	奇异值分解和特征分解	240
4.6.8	基于奇异值分解的矩阵性质	241
4.6.9	奇异值分解与低秩表示	246
4.7	阅读材料	252
第五章 矩阵计算问题		257
5.1	线性方程组的直接解法	258
5.1.1	线性方程组问题	258
5.1.2	容易求解的线性方程组	263
5.1.3	线性方程组的直接解法	268
5.1.4	敏感分析与其他方法	274
5.2	最小二乘问题	278
5.2.1	最小二乘问题与线性回归	278
5.2.2	最小二乘问题的求解方法	283
5.2.3	最小二乘问题的变体	287
5.2.4	最小二乘问题的解的敏感性	289
5.3	特征值计算	290
5.3.1	矩阵特征值分布范围的估计	290
5.3.2	幂法	293
5.3.3	加速幂法的方法	296
5.3.4	反幂法	297
5.3.5	特征值计算的应用: Pagerank 网页排名	299
5.4	阅读材料	303
第六章 向量与矩阵微分		307
6.1	向量函数和矩阵函数	308
6.1.1	函数	308
6.1.2	算子	312
6.1.3	泛函	313
6.1.4	机器学习中的风险泛函	314
6.2	统计机器学习中的非概率型函数模型	317
6.2.1	线性模型中的函数	317

6.2.2 感知机模型中的函数	318
6.2.3 支持向量机	320
6.2.4 降维和主成分分析中函数	327
6.2.5 聚类中的函数	329
6.3 深度神经网络中的函数构造	331
6.3.1 深度神经网络模型函数的构造过程	332
6.3.2 激活函数	335
6.3.3 跨步下采样与池化	339
6.4 向量值函数和矩阵微分	341
6.4.1 向量函数的梯度	342
6.4.2 矩阵函数的梯度	346
6.4.3 对矩阵微分	348
6.5 迹函数和行列式的微分	351
6.5.1 关于逆矩阵的函数的微分	355
6.5.2 关于行列式函数的梯度	355
6.6 向量值函数和矩阵值函数的梯度	358
6.6.1 向量值函数的梯度	358
6.6.2 矩阵值函数的梯度	359
6.6.3 向量值函数微分	359
6.7 链式法则	361
6.8 反向传播和自动微分	362
6.8.1 反向传播	363
6.8.2 自动微分	365
6.9 高阶微分和泰勒展开	371
6.9.1 Hessian 矩阵	371
6.9.2 线性化和多元泰勒级数	372
6.10 阅读材料	373
第七章 概率基础	377
7.1 概率论基本概念回顾：数据不确定性描述的观点	377
7.1.1 概率论基本概念	377
7.1.2 概率论公理	379
7.1.3 独立事件和条件概率	380
7.1.4 贝叶斯理论	381
7.2 随机变量及其分布	382

7.2.1	随机变量的常见分布	384
7.2.2	多维随机变量及其分布函数	385
7.3	随机变量的数字特征	388
7.3.1	期望	388
7.3.2	方差	390
7.3.3	一些重要的随机变量的期望和方差	391
7.3.4	协方差和相关系数	393
7.3.5	矩和协方差矩阵	395
7.3.6	条件期望	398
7.3.7	方差的应用: 过拟合与偏差-方差分解	400
7.4	概率不等式	403
7.5	大数定律与中心极限定理	407
7.5.1	引言	407
7.5.2	大数定律	408
7.5.3	中心极限定理	413
7.5.4	统计学习策略	415
7.5.5	泛化误差上界	416
7.6	随机过程简介	418
7.6.1	马尔科夫链	419
7.6.2	高斯过程	425
7.7	阅读材料	426
第八章	信息论基础	429
8.1	熵、相对熵和互信息	430
8.1.1	自信息	431
8.1.2	熵及其性质	432
8.1.3	联合熵和条件熵	436
8.1.4	互信息和相对熵	438
8.1.5	熵、相对熵和互信息的链式法则	442
8.1.6	信息不等式	442
8.2	连续分布的微分熵和最大熵	444
8.2.1	连续信源的微分熵	444
8.2.2	连续信源的最大熵	447
8.3	信息论在数据科学中的应用	447
8.3.1	基于信息量的度量	447

8.3.2 其他概率相关的度量	449
8.4 阅读材料	451
第九章 概率模型	454
9.1 建模的概率思想	454
9.1.1 分布和推断	455
9.1.2 生成式模型 vs. 判别式模型	455
9.1.3 参数化模型和参数估计	456
9.1.4 非参数模型和非参数估计	456
9.1.5 概率图模型	457
9.1.6 统计推断的基本概念	459
9.2 参数估计	462
9.2.1 矩估计	463
9.2.2 极大似然估计	464
9.2.3 常见分布的极大似然参数估计	465
9.2.4 极大后验估计	470
9.2.5 贝叶斯推断	473
9.3 非参数估计	476
9.3.1 直方图估计	476
9.3.2 核密度估计	477
9.3.3 非参数回归估计	477
9.3.4 CDF 和统计泛函的估计	479
9.4 概率模型的图语言描述	481
9.4.1 条件独立性	481
9.4.2 DAGs	482
9.4.3 无向图	484
9.4.4 概率与图	484
9.4.5 团与势	485
9.5 统计决策理论	486
9.5.1 引言	486
9.5.2 比较风险函数	487
9.5.3 最小最大规则	488
9.5.4 极大似然、最小最大和贝叶斯	490
9.6 阅读材料	490

草稿
请勿外传

第十章 优化基础	498
10.1 优化简介	499
10.1.1 数据科学与机器学习中最优化问题的例子	500
10.1.2 其他常见的优化问题举例	502
10.1.3 优化问题的一般形式	505
10.1.4 优化问题的分类	507
10.2 凸集	510
10.2.1 仿射集合和凸集	510
10.2.2 重要的凸集例子	513
10.2.3 保持凸集的运算	516
10.2.4 分离与支撑超平面	520
10.3 凸函数	522
10.3.1 凸函数的定义和基本性质	523
10.3.2 凸函数举例	525
10.3.3 凸函数的判定条件	525
10.3.4 保凸运算	528
10.3.5 共轭函数	535
10.4 凸优化	538
10.4.1 优化问题	538
10.4.2 凸优化问题	538
10.4.3 常见的凸优化问题	544
10.5 阅读材料	551
10.6 习题	553
10.7 参考文献	555
第十一章 最优性条件和对偶理论	561
11.1 Lagrange 对偶函数	561
11.1.1 Lagrange 函数与对偶函数	562
11.1.2 常见优化问题目标函数的对偶函数	564
11.1.3 共轭函数	566
11.2 Lagrange 对偶问题	568
11.2.1 Lagrange 对偶问题	568
11.2.2 对偶性质	570
11.2.3 常见优化问题的对偶问题	571
11.3 最优性条件	574

11.3.1 次优解认证和终止准则	574
11.3.2 互补松弛条件	575
11.3.3 KKT 最优性条件	576
11.3.4 通过解对偶问题求解原问题	577
11.4 数据科学中常见模型的对偶问题	579
11.4.1 分类模型：感知机	579
11.4.2 分类模型：支持向量机	582
11.5 阅读材料	588
11.6 习题	588
11.7 参考文献	590
第十二章 优化算法	592
12.1 无约束优化	592
12.1.1 零阶方法	596
12.1.2 一阶方法	602
12.1.3 二阶方法	616
12.2 约束优化	623
12.2.1 可行方向法	625
12.2.2 制约函数法	629
12.3 深度学习常用优化算法	636
12.3.1 随机梯度下降	637
12.3.2 动量梯度下降	639
12.3.3 自适应学习速率	642
12.4 阅读材料	645
12.5 习题	645
12.6 参考文献	648

草稿请勿外传

第一章 绪论

本章我们将简要介绍数据科学与工程数学基础在数据科学与大数据专业中的定位、应用背景、服务学科领域和主要数学内容的构成以及相关的数学基础简史，使读者对本书有初步的了解。1.1节主要从大数据结构的角度来探讨数据科学与工程数学基础在数据科学与大数据专业中的定位。1.2节从应用的角度探讨各种智能处理任务如何在数据的框架下归结为数据分析的各种基本运算任务。1.3节叙述数据分析的各种基本运算任务的理论背景也即机器学习的基本概念、问题模式、方法要素和应用任务以及与这些理论涉及的相关数学基础。1.4节给出数据科学与工程所需的数学内容框架，给出粗略的概览，界定本教材涉及的数学内容的范围。1.5节概览本教材涉及的数学基础简史。1.6节介绍本教材的使用方式，相应的教学资源和教学建议。

1.1 本教材产生的背景和定位

近年来，人工智能的强势崛起，特别是2016年AlphaGo和韩国九段棋手李世石的人机大战，让我们深刻地领略到了数据（data）和模型驱动的机器学习技术的巨大潜力。数据是载体，智能是目标，而数据分析技术、特别是机器学习是从数据通往智能的技术、方法和途径。因此，机器学习是数据分析的核心，是现代人工智能的本质。

机器学习就是关于计算机基于数据构建数学模型并运用模型对数据进行预测与分析，从数据中挖掘出有价值的信息的学科。数据本身是无意识的，它不能自动呈现出有用的信息。通俗地说，数据是指对客观事件进行记录并可以鉴别的符号，数据是信息的载体，我们研究数据是希望获得信息，没有联系的，孤立的数据是不能获得信息的，只有当这些数据可以用来描述一个客观事物和客观事物的关系，形成有逻辑的数据流，他们才能被称为信息。因此信息是来源于数据并高于数据。但是信息具有实效性，只有通过对信息进行归纳、演绎、比较等手段进行挖掘，使其有价值的部分沉淀下来，并与已存在的人类知识体系相结合，这部分有价值的信息就转变成知识。因此，我们研究数据的目标之一是发展一套数据处理技术以期从中获得信息和知识。

那么有哪些类型的数据需要研究呢？我们这里所描述的数据是可以被计算机识别存储并加工处理的描述客观事物的信息符号的总称，所有能被输入计算机中，且能被计算机处理的符号

N	数据类型	N	大数据特性	数量
1	关系数据	1	高维	
2	时间序列	2	海量	
3	图数据	3	多模	
4	文本数据	4	高速	
5	图片	5	噪声	
6	视频	6	缺失	
7	音频	7	非平衡	
		8	稀疏	

图 1.1: 数据类型和大数据的特性描述

的集合，它是计算机程序加工处理的对象。客观事物包括数值、字符、声音、图形、图像等，如图1.1，它们本身并不是数据，通常被称为衍生数据，只有通过编码变成能被计算机识别、存储和处理的符号形式后才是数据。当前由于信息技术和互联网的广泛发展，形成了由大量衍生数据为基础构成的所谓大数据。那么怎样才能从大数据中找出有价值的东西呢？这首先需要我们对大数据的结构特性有清晰的理解，然后基于这种结构来发展相应的数据处理技术以从中获得相应的信息和知识。然而，目前我们对大数据的刻画基本上都是用描述性的语言，比如，高维，海量这种模糊的术语（如图1.1），而对大数据的本质结构并没有清晰的数学刻画。

为了回答这个问题，我们来看看数据分析解决问题的步骤：

- (1) 首先要给数据一个抽象的表示；
- (2) 其次基于表示进行建模，建立数学模型；
- (3) 接着估计模型的参数，也就是计算或设计解此模型的算法；
- (4) 然后编出程序、进行测试、调整得到最终解答；
- (5) 最后为了应对大规模的数据所带来的问题，我们还需要设计一些高效的实现手段，包括硬件层面和算法层面。

这一过程与传统计算机科学解决数据计算问题的过程是相似的。传统计算机科学处理问题也涉及对数据进行表示，并建立一个数学模型以及设计一个解此模型的算法。其中构建数学模型的实质是分析问题，从中提取操作的对象，并找出这些操作对象之间含有的关系，然后用数学的语言加以描述。这里有两种情况要考虑：

- (1) 对于数值计算问题：所用的数学模型是用数学方程描述，所涉及的运算对象一般是简单的整型、实型和逻辑型数据，因此程序设计者的主要精力集中于程序设计技巧上，而不是数据的存储和组织上。
- (2) 计算机科学应用的更多领域是“非数值型计算问题”，处理的对象是类型复杂的数据，它们的数学模型无法用数学方程描述，而是用数据结构描述，因此程序设计需要设计出合适的数据结构来对数据进行有效的存储和组织。众所周知，数据结构最早是由美国计算机科学家、图

N	经典的数据结构	离散关系
1	逻辑结构	集合、线性、树形、图形（常用数据结构：数组、栈、队列、链表、树、图、堆）
2	物理结构	顺序、链接、索引、散列
3	运算结构（结构算法 ）	检索、插入、删除、更新和排序

数据结构：在同一类有限的数据集中，研究数据元素离散关系和数据运算

图 1.2: 经典数据结构

灵奖得主唐纳德·克努特（Donald Ervin Knuth）于 1968 年在其《计算机程序设计艺术》系统提出。传统计算机科学中经典的数据结构，用一句可以概括为：在同一类有限的数据集中，研究数据元素离散关系和数据运算，其具体内容包括如图 1.2 所示。而计算机科学算法（Algorithm）是指对解决方案的准确而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制，它也依赖于数据结构。由此我们，可以看出传统计算机科学的核心——算法与程序设计以及其依赖的数据结构，这些内容都是建立在离散结构基础之上的。而离散结构主要指离散对象之间的数学结构，所以又称离散数学，已成为传统计算机科学的核心数学基础，所以计算机科学是以“离散数学”为重点的数学体系。离散数学这个名称最终在 1974 年由美国 IEEE 计算机协会典型课程分委员会正式提出，并于 1976 年把它列为计算机科学的核心课程。离散数学主要包括传统的逻辑学，集合论（包括函数），数论基础，算法设计，组合分析，离散概率，关系理论，图论与树，抽象代数（包括代数系统、群、环、域等），布尔代数，计算模型（语言与自动机）等等，主要用于描述经典数据的物理结构和逻辑结构，以及增、删、改、查等运算结构。

回到现今的数据科学与工程面临的数据处理问题，与传统计算机科学一样，大部分也都是“非数值型计算问题”。对于大数据，它们的数学模型也无法用数学方程来描述，我们进行程序设计和建立数学模型的目的是（1）更高效的存储和组织大数据；（2）发现大数据中有别于传统离散关系的新数据关系，如相关关系（包括相似关系、顺序关系、类别关系）或因果关系；（3）由这些新的数据关系引出或定义新的数据运算结构，比如我们常见的分类运算（如电商希望对其客户数据进行建模分析来实现客户分类）、聚类、回归、降维（能够对数据进行可视化）和排序等运算。对数据进行这些关系发现和数据运算获得的结果可以归结为从数据中获得信息和知识。如果这一套流程全部依赖于计算机程序来完成，并自动化的辅助人类决策，就形成所谓的人工智能，更准确说是数据驱动的人工智能。在上述三个目的中，大数据的存储和组织形式与计算机科学中经典的数据和存储形式并没有很大的差异和变化（增加了并行处理等模式），所以这一部分仍然依赖于经典的数据结构以及相应的离散数学基础。然而，对于第二和第三个目的，其实属于数据分析的范畴，仅仅具有经典的数据结构和相应的离散数学基础是不够的，需要形成一套新的大数据结构以及相应的数学基础来支撑。

N	数据数学结构	相关关系或因果关系
1	代数结构	向量、矩阵、张量
2	度量结构	欧氏距离、范数
3	网络结构	有向图、无向图
4	拓扑结构	Klein瓶
5	函数结构	线性函数、分片函数

运算结构：分类、回归、聚类、降维、密度估计和排序等

图 1.3: 大数据的数学和运算结构

如果我们把大数据特性中海量高维数据集扩展为无限数据集；多模数据集归结为多元数据集；高速到达数据归结为快速增长的数据集；噪声、缺失、非平衡、稀疏归结为奇异性，则数据分析中大数据所依赖的大数据结构可粗略的总结为：在多元无限快速增长的数据集中，研究数据的相关关系或因果关系和数据运算，见图 1.3。

对上述定义的大数据结构的研究事实上是现今机器学习的主要内容。而支撑这套数据结构的数学基础已突破了传统的离散数学，更多的是与矩阵计算、概率与统计、信息论和优化理论等连续数学相关（注意统计学如今在国内外都属于与数学并行的一级独立学科，但是因为其很多理论基础植根于数学，所以我们在这里仍然把它归为应用数学的范畴），因此需要形成一套新的数据科学与工程的数学基础来支撑对大数据结构的研究。从上述角度看，数据科学与工程的数学基础之于数据科学类似于离散数学之于计算机科学。我们需要在这些新的大数据结构维度上考虑问题。

那么，这些基础具体包括哪些内容呢？我们在 1.4 节会详细给出。下面我们首先在 1.2 节通过数据科学或人工智能中两类常见的应用场景，也即视觉感知和自然语言处理的例子，来展示人工智能的很多应用任务处理都可以归结为上述提到的大数据结构中各种数据关系和运算任务问题。然后在 1.3 节我们会介绍这些数据分析运算任务的理论基础，也即机器学习的理论背景以及相关涉及的数学问题，并由此在 1.4 节给出数据科学与工程所需的数学内容框架。

1.2 从图像感知到自然语言处理

下面我们通过介绍两类场景案例来展示数据驱动的图像感知和自然语言处理问题是如何转变成一个基本的数据分析计算任务的。第一个案例是感知任务分析，涉及图像识别，代表了近年来以数据驱动的人工智能研究的核心进展。第二个案例是信息检索和文本分类，代表了数据科学与机器学习被广泛认可的一个成功应用。这些案例将简要的表明数据驱动的人工智能应用中数据分析任务涉及的代数表示和概率建模以及各种优化问题。其中文本分类涉及凸优化问题——源于逻辑回归模型或支持向量机的使用；而感知任务通常涉及高度非线性和非凸优化问题

——源于深度神经网络的使用。这两个案例将在全书的很多地方被提及，作为在全书介绍很多重要的数学概念和结论的应用案例。

1.2.1 猫、分类和神经网络

计算机视觉旨在识别和理解图像/视频中的内容。其诞生于 1966 年 MIT AI Group 的“the summer vision project”。当时，人工智能其他分支的研究已经有一些初步成果。由于人类可以很轻易地进行视觉认知，MIT 的教授们希望通过一个暑期项目解决计算机的视觉问题。当然，计算机视觉没有被一个暑期内解决，但计算机视觉经过 50 余年发展已成为一个十分活跃的研究领域。如今，互联网上超过 70% 的数据是图像/视频，全世界的监控摄像头已超过人口数，每天有超过八亿小时的监控视频数据生成。如此大的数据量亟待自动化的视觉理解与分析技术。

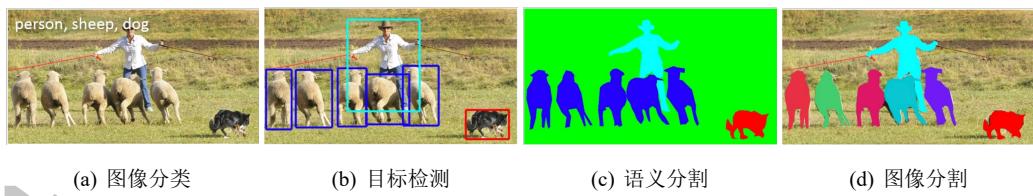


图 1.4: 计算机四大视觉任务

图1.4是计算机视觉领域的四大基本任务，包括图像分类、目标检测、语义分割和实例分割。给定一张输入图像，图像分类任务旨在判断该图像所属类别；目标检测既要识别出图中的物体，又要知道物体的位置，即图像分类和目标定位任务；语义分割除了识别物体类别与位置外，还要标注每个目标的边界，将物体进行像素级别的分割提取，但不区分同类物体；实例分割任务，除了识别物体类别与位置外，还要标注每个目标的边界，且区分同类物体。



图 1.5: 计算机视觉任务（以猫为例）

更为具体地，以猫为例，图1.5a 对猫进行像素级别的语义分割，区分背景物体和前景物体；

图1.5b对猫进行目标检测，即分类和定位；图1.5c对多个物体进行目标检测，既标识了图中的猫，也标识了图中的狗；图1.5d则是针对多个物体的语义分割。这些问题都有关图像分类，因此本节重点介绍图像分类问题。所谓图像分类问题，就是已有固定的分类标签集合，然后对于输入的图像，从分类标签集合中找出一个分类标签，最后把分类标签分配给该输入图像。

如例1，已知一个类别标签集合 $\{\text{airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck}\}$ ，计算机如何判断图1.6是属于哪个类别呢？需要注意的是，计算机可以使用RGB位图表示彩色图像，RGB位图用三维数组表示，数组元素的取值范围为 $[0, 255]$ 的整数，数组的大小是宽度 \times 高度 \times 通道数，RGB图像的通道数即红、绿、蓝三个通道。这张猫的图像就可以用大小为 $32 \times 32 \times 3$ 的三维数组进行表示。

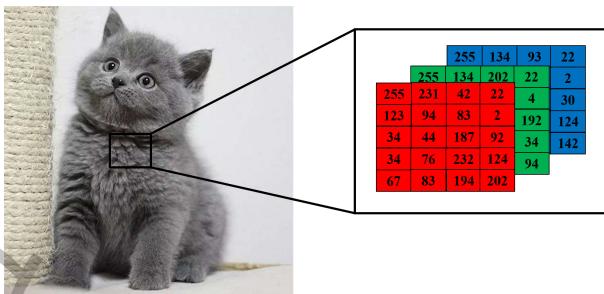


图 1.6: 图像的数据表示

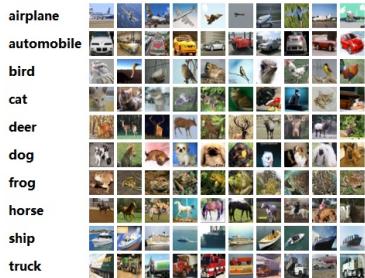


图 1.7: CIFAR-10

在机器学习中，我们经常采用基于数据驱动的方法对图像进行分类，即给计算机大量图像数据和标签，然后实现学习算法，让计算机学习到每个类别的特征。具体方法流程如下：

- 1 输入：输入是 N 个图像的集合（即训练集），每个图像的标签是所有分类标签中的一种；
- 2 学习：使用训练集来学习每个类的特征，这一步也被称为是在训练分类器或学习一个模型；
- 3 评价：让分类器预测未曾见过的测试图像的标签，将预测标签与真实标签进行对比，来评价分类器的质量。通常使用测试集的准确率、精确率、召回率和F1-score等指标来评价分类器。

对图像分类的基本流程有了一定了解之后，我们开始对数据集进行划分和预处理。在这个例子中，我们选取的是在图像分类任务中常用的一个数据集，CIFAR-10，该数据集包含60000张 $32 \times 32 \times 3$ 的图像，共有10个类别，每个类别有6000张图像和对应标签。我们对数据集进行划分，将每个类别的5000张图片作为训练集，剩下的1000张图片作为测试集。除了划分数据集之外，我们常对数据进行预处理，例如重塑、归一化和降维。重塑可以将 $32 \times 32 \times 3$ 的3维数组转化为一个 3072×1 的一维数组，即3072维的向量。归一化操作可以保证计算距离时各个维度的量纲保持一致。如果数据是高维数据，我们也会考虑使用降维方法对数据进行降维，比如PCA降维，如图1.8所示，数据从二维向量降到一维向量。

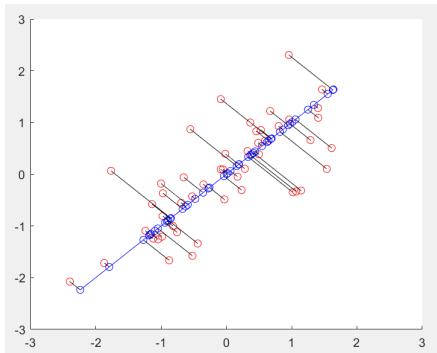


图 1.8: PCA 降维

在对数据进行预处理之后，我们要正式开始构建模型并进行学习。在这里，我们将简单介绍三种类型的分类器，分别是 K 最近邻分类算法、线性分类算法和卷积神经网络算法。

K 最近邻分类算法是数据挖掘分类技术中最简单的方法之一。所谓 K 最近邻，就是 K 个最近的邻居的意思，说的是每个样本都可以用它最接近的 K 个邻居来代表。KNN 就是通过测量不同特征值之间的距离来进行样本分类。

在 CIFAR-10 中，首先将测试图像和训练图像转化为两个 3072 维的向量 I_1 和 I_2 ，然后计算它们之间的 L_1 距离：

$$d_1(I_1, I_2) = \sum_{p=1}^{3072} |I_1^p - I_2^p|$$

当然，我们也经常选择 L_2 距离：

$$d_2(I_1, I_2) = \sqrt{\sum_{p=1}^{3072} (I_1^p - I_2^p)^2}$$

test image				training image				pixel-wise absolute value differences			
56	32	10	18	10	20	24	17	46	12	14	1
90	23	128	133	8	10	89	100	82	13	39	33
24	26	178	200	12	16	178	170	12	10	0	30
2	0	255	220	4	32	233	112	2	32	22	108

$\xrightarrow{\text{add}} 456$

图 1.9: 以图片中的一个颜色通道为例

如图1.9所示，以图片中的一个颜色通道为例。两张图像通过 L_1 距离进行比较，逐个像素求差值，再将所有差值求和。如果两张图像完全一样，则 L_1 距离为 0；如果两张图像差异极大，则 L_1 值将会非常大。

但同时我们会有疑问，KNN 算法中的 K 值该如何选取呢？计算距离时是选择 L_1 距离还是 L_2 距离呢？这些选择被称为超参数。在数据驱动的机器学习算法设计中，超参数十分常见，但如何选取往往需要通过验证集进行参数调优。

对于 KNN 算法，它在 CIFAR-10 数据集上可以得到近 40% 的准确率，实现起来非常简单，但对磁盘存储有要求。分类器必须记住所有训练数据并将其存储起来，以便于未来测试数据用于比较。这在存储空间上是低效的，数据集大小也很容易就以 GB 计。对一个测试图像进行分类时，需要和所有训练图像逐一比较，算法计算资源耗费高。因此我们寻求一种更强大的方法来解决图像分类问题。

该方法就是线性分类器，它可以很自然地延伸到神经网络和卷积神经网络上。这种方法主要有两部分组成：一个是评分函数，它是原始图像数据到类别分值的映射。另一个是损失函数，它是用来量化预测分类标签与真实标签之间一致性的。该方法可以转化为一个最优化问题，在最优化过程中，将通过更新评分函数的参数来最小化损失函数值。

在本模型中，我们从最简单的概率函数开始，一个线性映射：

$$f(\mathbf{W}, \mathbf{b}; \mathbf{x}_i) = \mathbf{W}\mathbf{x}_i + \mathbf{b}$$

在此公式中，假设每个图像数据集都被拉成为一个长度为 D 的列向量，大小为 $[D \times 1]$ 。其中 $[K \times D]$ 的矩阵 \mathbf{W} 和大小为 $[K \times 1]$ 的列向量 \mathbf{b} 为该函数的参数。仍然以 CIFAR-10 为例， \mathbf{x}_i 就包含了第 i 个图像的所有像素信息，这些信息被拉成为一个 $[3072 \times 1]$ 的列向量， \mathbf{W} 大小为 $[10 \times 3072]$ ， \mathbf{b} 的大小为 $[10 \times 1]$ 。因此，3072 个数字输入函数，函数输出 10 个不同类别的得分。参数 \mathbf{W} 被称为权重， \mathbf{b} 被称为偏差向量。在此可以预告一下，卷积神经网络映射图像像素值到分类分值的方法和线性分类器一样，但是映射 f 就要复杂的多，其包含的参数也更多。

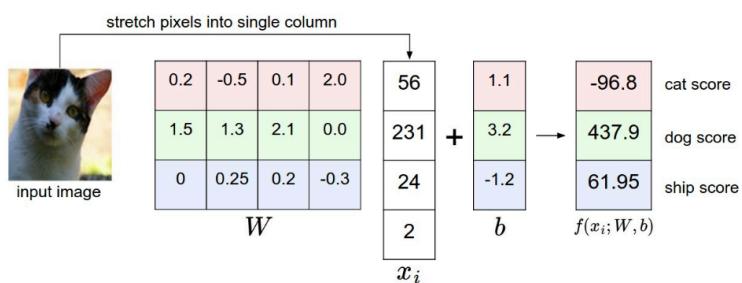


图 1.10: 评分函数可视化。假设图像只有 4 个像素（也不考虑 RGB 通道），有 3 个分类（cat、dog、ship）。首先将图像像素拉伸为一个列向量，与 \mathbf{W} 进行矩阵乘法，然后得到各个分类的分值

为了便于可视化。假设图像只有 4 个像素，有 3 个分类，红色代表猫，绿色代表狗，蓝色代表船。首先将图像像素拉伸为一个列向量，与 \mathbf{W} 进行矩阵乘法，得到各个分类的分值。需要注意的是，这个 \mathbf{W} 结果不佳：猫分类的分值非常低。从图 1.10 中看，算法认为这个图像是一只狗。

这是就需要使用损失函数来衡量我们对结果的不满意程度，当评分函数输出结果与真实结果之间差异越大，损失函数输出越大，反之越小。

Softmax 是最常用的分类器之一，使用 softmax 函数将一组 $(-\infty, +\infty)$ 的得分 f 转换为一组 $(0, 1)$ 的概率，并且这组概率的和为 1。每张训练图像属于类别 i 的概率得分可以用公式表示：

$$p_i = \frac{e^{f_i}}{\sum_{j=1}^K e^{f_j}}$$

根据预测类别的概率得分，使用交叉熵函数作为损失函数，计算真实标签与预测标签之间的损失。将标签 y 转换成 one-hot 向量 y ，例如真实标签为 4，则 $y = [0, 0, 0, 0, 1]$ 。每张训练图像对应的交叉熵损失用公式表示为：

$$l = - \sum_{c=1}^K y_c \log(p_c) = -y_c \log(p_c)$$

显然每张图像都只需要计算一个类别的概率得分和真实标签的交叉熵。因此第 i 张图像的交叉损失函数又可以表示为：

$$l_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_{j=1}^K e^{f_j}}\right) = -f_{y_i} + \log\left(\sum_{j=1}^K e^{f_j}\right).$$

定义了损失函数后，我们需要确定最优化目标，最优化的目标即对所有训练集的图像的损失和最小

$$\min Loss = \min \sum_{i=1}^N l_i = \min - \sum_{i=1}^N \log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right).$$

为了寻找能使得损失函数值最小化的参数 W 的过程，可以考虑多个策略：

1. 随机搜索。从随机权重开始，然后迭代取优，从而获得更低的损失值。
2. 随机本机搜索。从随机权重开始，然后生成一个随机的 δW ，只有当 $W + \delta W$ 的损失值变低，才可以更新。
3. 跟随梯度。从数学上计算最陡峭的方向，然后向着最陡峭的方向下降。

梯度下降法如图 1.11 所示：

我们已经了解到基于参数的评分函数、损失函数和最优化过程之间是如何运作的，这样我们将会回到第一个部分，基于参数的函数映射，然后将其拓展为一个远比线性函数复杂的函数——卷积神经网络。之前提到过，卷积神经网络映射图像像素值到分类分值的方法和线性分类器一样，但是映射 f 要复杂的多，其包含的参数也更多。而损失函数和最优化过程在两个部分将会保持相对稳定。

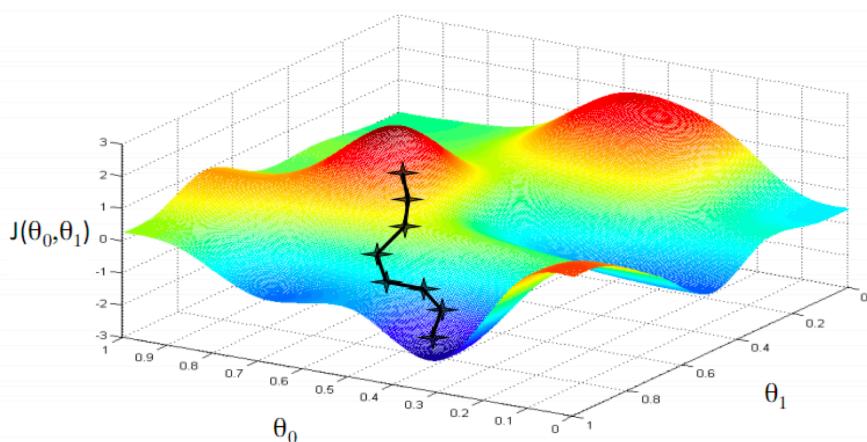


图 1.11: 梯度下降算法

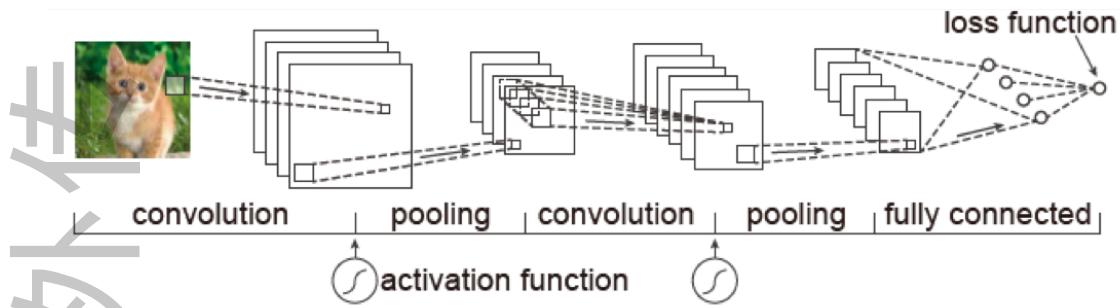


图 1.12: 一个典型的 CNN 架构图

一个典型的卷积神经网络架构如图1.12所示，输入一张图像，经过一系列卷积层、非线性层、池化层和完全连接层，最终得到类别概率输出。在一个简单的卷积神经网络中，每层都使用一个可以微分的函数将激活数据从一个层传递到另一个层。在本例中，一个用于 CIFAR-10 图像数据分类的卷积神经网络的结构可以是 [输入层-卷积层-ReLU 层-池化层-全连接层]。

本例中图像宽高均为 32，有 3 个颜色通道，则图像的原始像素值为 $[32 \times 32 \times 3]$ 。在卷积层中，神经元与输入层中的一个局部区域相连，每个神经元都计算自己与输入层相连的小区域与自己权重的内积。卷积层会计算所有神经元的输出。如果我们使用 12 个滤波器（也叫作核），得到的输出数据体的维度就是 $[32 \times 32 \times 12]$ 。ReLU 层将会逐个元素地进行激活函数操作，比如使用以 0 为阈值的 $\max(0, x)$ 作为激活函数。该层对数据尺寸没有改变，还是 $[32 \times 32 \times 12]$ 。由于线性模型的表达能力不够，在该层中使用的激活函数都是非线性函数。池化层在空间维度（宽

度和高度)上进行降采样操作, 数据尺寸变为 $[16 \times 16 \times 12]$, 主要是对ReLU层的输入进行压缩, 使得特征图变小, 提取主要特征。全连接层将会计算分类评分, 数据尺寸变为 $[1 \times 1 \times 10]$, 其中10个数字对应的就是CIFAR-10中10个类别的分类评分值。正如其名, 全连接层与常规神经网络一样, 其中每个神经元都与前一层中所有神经元相连接。

由此看来, 卷积神经网络一层一层地将图像从原始像素值转换成最终的分类评分值。其中有的层含有参数, 有的没有。具体说来, 卷积层和全连接层对输入执行变换操作的时候, 不仅会用到激活函数, 还会用到很多参数。而ReLU层和池化层则是进行一个固定不变的函数操作。卷积层和全连接层中的参数会随着梯度下降被训练, 这样卷积神经网络计算出的分类评分就能和训练集中的每个图像的标签吻合了。

至此, 我们了解了图像识别任务及其所涉及的机器学习任务和数学基础。

1.2.2 文本、词向量和朴素贝叶斯

我们知道, 计算机视觉主要是让计算机具有“看”客观世界的能力, 而语音识别主要是让计算机“听”外界的声音, 自然语言处理主要解决如何让计算机理解人类语言, 更好地进行人机交互。因此自然语言处理是人工智能的另一大核心研究主题。下面我们进一步通过自然语言处理相关的例子来了解其涉及的数据分析任务和相关的数学基础。

目前自然语言处理的应用主要有自动问答、机器翻译和信息检索等。而这些任务又大致可归结为四大类任务: 文本分类(如: 舆情监测, 新闻分类)、序列标注(如: 分词, 词性标注, 命名实体识别)、文本匹配(如: 搜索引擎, 自动问答)和文本生成(如: 机器翻译, 文本摘要)。下面我们将以文本分类为例来介绍自然语言处理的建模流程。

文本分类也称为自动文本分类, 是指给定文档 p (可能含有标题 t), 将文档分类为 n 个类别中的一个或多个。是自然语言处理领域一个比较经典的任务。实现这个任务传统的机器学习方法有逻辑回归模型和svm等, 最新的深度学习方法有FastText和TextCNN等。文本分类的应用也很广泛, 包括常见的垃圾邮件识别、以及近年来兴起的情感分析等。

文本分类的流程如下图1.13所示, 包括: 文档的输入, 对文档进行预处理, 然后对其进行文本表示, 文本表示完成后, 就可以设计一个分类器来对文档进行分类。

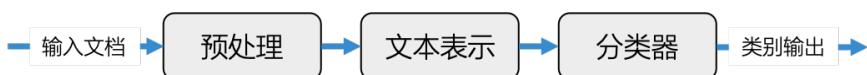


图 1.13: 文本分类流程

下面我们以电影评论分类为例, 来介绍文本表示和分类器设计这两个任务是如何进行的, 涉及到哪些数学基础。

例 1.2.1. 以文本分类中的影评分类为例，介绍自然语言处理的建模流程。影评分类数据如下所示：

电影影评	类别
the plot of this movie is funny, excellent!!!	1
this movie is awful indeed.	0

图 1.14: 两条电影影评数据

这里主要有两个问题需要考虑：第一个问题是如何在计算机中表示电影评论数据（为简化处理，忽略影评数据中的标点符号）；第二个问题是基于影评数字化表示，对其进行分类建模。在这个例子考虑如下的影评数据样本示例，共有两条影评。一类是正类影评，类别是 1。另外一类是负类影评，用 0 表示。在这里，我们仅展示了两条影评作为样例，实际应用中影评数据集可以很大，比如 keras 上的 IMDB 数据集内部集成了 5 万条严重两级分化的数据。影评分类不仅可以让我们知道观众的喜好和反馈，还可以用于指导电影工业的制片和放映排片，甚至可以当成电影票房预估的影响因素之一。对于影评分类问题的第一步也是最基础的一步就是如何表示文本，然后在基于文本表示的基础上，对文本分类进行建模。

文本表示属于语言表示，在方法上可以从两个维度进行区分。一个维度是按不同粒度进行划分，语言具有一定的层次结构，语言表示可以分为字、词、句子、篇章等不同粒度的表示。另一个维度是按表示形式进行划分，可以分为离散表示和连续表示两类。离散表示是将语言看成离散的符号，而将语言表示为连续空间中的一个点，包括分布式表示和分散式表示。文本表示的目的是指将字词处理成向量或矩阵，以便计算能处理，因此文本表示是自然语言处理的开始环节。当前主流的文本表示方法大致有 5 种，分别是独热编码 (one-hot)、词袋模型、TF-IDF(是在词袋模型上进行了改进)、共现矩阵以及在深度学习中比较火的词嵌入表示。前四种属于离散表示，特点是离散、高维和稀疏。后一种是分布式表示，特点是连续、低维、稠密。

独热编码 (one-hot) 又称为一位有效编码，主要是采用 N 位状态寄存器来对 N 个状态进行编码，每个状态都由有独立的寄存器位，并且在任意时候只有一位有效；在自然语言处理领域中，通过将每个单词转换成一个个独热表示便于后续的处理。通过统计语料中所有不重复单词并得到不重复词表的大小为 V 。

one-hot 向量是最简单的词向量，用一个 $\mathbb{R}^{|V| \times 1}$ 向量来表示每个单词，将所有的词排序，每个词对应下标由 0 和 1 组成，下面给出例 1.2.1 的 one-hot 表示：

$$w^{the} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{plot} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{of} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots, w^{indeed} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

在例1.2.1中共有 10 个不重复的单词，所以词汇表的大小 $|V| = 10$ 。将每个词表示成一个 V 维的向量，向量的元素由 0 和 1 组成，且在每个词的独热表示中，只有一个位置数值为 1，其他位置的数值为 0；比如 plot 这个词在词表中处于第 2 个位置，所在 10 维的向量中，它在第二个位置元素为 1，其它都为 0。这样每个单词被表示成完全独立的实体，但任意两个词向量没有体现相似性的概念，即：

$$(w^{the})^T w^{plot} = (w^{the})^T w^{of} = 0$$

也就是说两个向量的点乘的结果都为 0，这样无法衡量词间的相似性也不能区分词的重要性。这里涉及向量的点乘和数据比较，在后面的章节会讲解。

词袋模型表示也被称为计数向量表示。在这种表示方法中，把文本看做是一个词袋，统计每个单词的个数，而忽略文本的语序、语法和句法；使用词袋模型表示文本，有两个步骤，以之前的影评数据作为语料：

第一步：统计语料中所有不重复的词并构建相应的索引词表 V ，由 10 个单词组成； $V = \{1 : “the”, 2 : “plot”, 3 : “of”, 4 : “this”, 5 : “moive”, 6 : “is”, 7 : “funny”, 8 : “excellent”, 9 : “awful”, 10 : “indeed”\}$ 。

第二步：是在词表 V 的基础上，将每个文本表示成词表大小的向量。具体的做法是：统计文本中每个单词的出现次数，并将该次数作为向量在词表索引号的值；最后得到了一个基于计数频次的文本的向量化表示。这个词表一共包含 10 个不同的单词，利用词表的索引号，例1.2.1中两个影评文本可以用两个 10 维向量表示：文本 1 表示为：[1, 1, 1, 1, 1, 1, 1, 1, 0, 0]，文本 2 可以表示为：[0, 0, 0, 1, 1, 0, 0, 1, 1]。

TF-IDF，即词频表示。是一种统计方法，用来评估一个字词对于一个文件集或一个语料库中的其中一份文件的重要程度，这表明同一个词在不同文章出现时，其重要性是不一样的；TF-IDF 的主要思想是：如果某个词或短语在一篇文档中出现的频率高，并且在其他文档中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类；词袋模型是基于计数得到的，而 TF-IDF 则是基于频率统计得到的。TF-IDF 的分数代表了词语在当前文档和整个语料库中的相对重要性。TF-IDF 分数由两部分组成：第一部分是词语频率 (TF)，第二部分是逆文档频率 (IDF)

$$TF(\text{单词}) = \frac{\text{该词在当前文档出现次数}}{\text{当前文档中的词语总数}}$$

$$IDF(\text{单词}) = \ln \frac{\text{文档总数}}{\text{出现该词语的文档总数}}$$

词语频率越高，那么这个词对这篇文档就越重要；IDF 就越大，那么包含某个词的文档越少，说明这个词具有很好的类别区分能力；TF-IDF 加权的各种形式常被搜索应用，作为文件与用户查询之间相关程度的度量或评级。

下面以影评数据为例，简单介绍下 TF-IDF 的计算过程，在实际过程中通常要复杂的多；以“plot”为例，计算其在文本 1 的 tf-idf 值：

$$tf_{plot, \text{文本 } 1} = \frac{1}{8} idf_{plot, \text{文本 } 1} = \ln \frac{2}{1} = \ln 2$$

$$\text{tf-idf}_{\text{plot, 文本 1}} = \text{tf}_{\text{plot, 文本 1}} \cdot \text{idf}_{\text{plot, 文本 1}} = \frac{1}{8} \cdot \ln 2 \approx 0.0866$$

以此类推，计算每个文档中的每个词的 tf-idf 值，并将 tf-idf 值放入到词袋向量中的相应位置，得到最终的表示；TF-IDF 是在词袋模型上进行的改进。词袋模型中文本向量的每个位置的值是通过统计词表索引中该位置的词出现的次数，而在 TF-IDF 则是计算每个位置的词的 tf-idf 值。我们在 2.1 节还会继续举用向量进行词袋模型和词频表示的例子。

one-hot 向量可以表示每个词，但是这样其实无法衡量词间的相似性，也不能区分词的重要性，这种现象可以通过共现矩阵得到一定的缓解。共现矩阵通过统计一个事先指定大小的窗口内的单词共现次数，以单词周边的共现词的次数做为当前单词的向量表示。基于影评语料记录每个单词在目标单词的特定大小的窗口（取窗口大小为 1，即只考虑与该单词邻接的词）中出现的次数，得到的关联矩阵 X ，称为共现矩阵：

	<i>the plot of this movie is funny excellent awful indeed</i>									
<i>the</i>	0	1	0	0	0	0	0	0	0	0
<i>plot</i>	1	0	1	0	0	0	0	0	0	0
<i>of</i>	0	1	0	1	0	0	0	0	0	0
<i>this</i>	0	0	1	0	2	0	0	0	0	0
$X =$ <i>movie</i>	0	0	0	2	0	2	0	0	0	0
<i>is</i>	0	0	0	0	2	0	1	0	1	0
<i>funny</i>	0	0	0	0	0	1	0	1	0	0
<i>excellent</i>	0	0	0	0	0	0	1	0	0	0
<i>awful</i>	0	0	0	0	0	1	0	0	0	1
<i>indeed</i>	0	0	0	0	0	0	0	0	1	0

比如，*this* 这个单词，左边相邻是 *of*，右边相邻是 *movie*，*of* 只出现 1 次，而 *movie* 出现两次，所以 *this* 这个词的词向量表示就是第 4 行或第 4 列的一个向量。该矩阵是一个对称矩阵，矩阵的每一行或者每一列都可以表示成该行或该列索引单词的词向量。对称矩阵在数据科学和机器学习领域具有重要的应用，很多数据表示和模型最后都归结为对称矩阵建模。共现矩阵很多元素是 0，因此这个矩阵也称为“稀疏矩阵”，稀疏矩阵问题在数据压缩和机器学习领域也有着重要的应用，这些内容我们在后面课程中会介绍。

共现矩阵这种方法在一定程度上缓解了 one-hot 向量相似度为 0 的问题，但由于其稀疏性依旧没有解决数据稀疏和维度灾难的问题，尤其是当语料库非常大时，这个矩阵会非常大，非常稀疏，对其进行计算研究会很困难。一个自然而然的解决思路是对原始词向量进行降维，从而得到一个稠密的连续词向量。降维属于无监督学习的一个主要应用，数学上会用到奇异值分解，我们在下一小节和第 4 章会讲解。

在这里可以注意到，对称矩阵、稀疏矩阵、奇异值分解是这门数学基础课程的核心概念。本书在后面会重点讲述。

前面考虑都是离散稀疏的表示，下面我们就来看看连续的分布式表示，词嵌入表示。它是文本的分布式表示，是过去一些年深度学习方法处理自然语言问题的重大突破之一。词嵌入表示可以理解成是一种映射，通过将文本空间中的单词通过一定的方式映射到另外一个数值向量空间，在该数值空间中，意义相似的单词具有类似的表示形式，即它们在这个数值空间中相对其他意义不同的词的距离会更近。常见的词嵌入表示包括：Word2vec、Glove、Fasttext 和 Bert 等。本节重点介绍一下 Word2vec 的词嵌入过程。Word2Vec 又包括连续词袋（cbow）和连续跳跃元语法（skip-gram）两种模型。下面以连续词袋模型为例，简单介绍下词嵌入的过程。

CBOW 模型基于上下文来预测当前的词，从而学习到词嵌入，其中上下文是由一个邻近词窗口来定义。如下示意图 1.15 表示了简单的 CBOW 模型。在该模型中，假设窗口大小为 $2i+1$ ，每个词向量 $w_t \in \mathbb{R}^{|V|}$, n 为输入向量的个数, C 是上下文单词的个数, $V \in \mathbb{R}^{|V| \times n}$ 和 $U \in \mathbb{R}^{n \times |V|}$ 是两个权重矩阵。

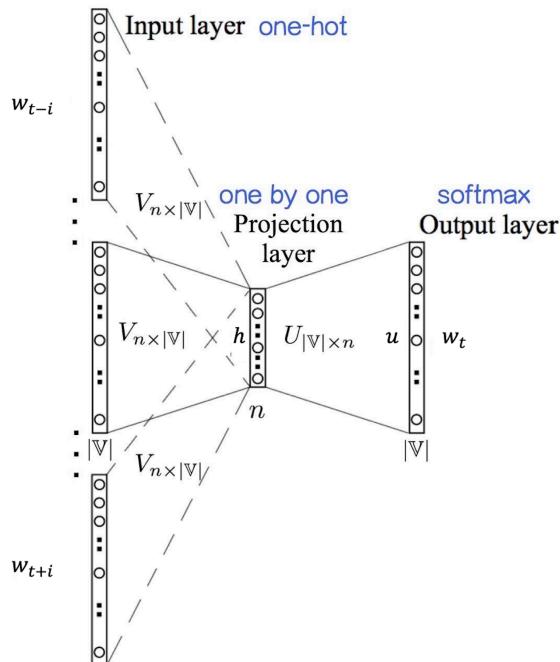


图 1.15: CBOW 模型。 w_t 为目标词，其余词 $w_i, i \neq t$ 为上下文 $w_{context}$

第一步：计算隐层 \mathbf{h}

$$\mathbf{h} = \frac{1}{C} V^T \cdot \left(\sum_{i=1}^C \mathbf{w}_i \right)$$

第二步：计算输出层输出

$$\mathbf{u}_j = \mathbf{U}^T \cdot \mathbf{h}$$

草稿勿外传

$$y_j = p(\mathbf{w}_t | \mathbf{w}_{context}) = \frac{\exp(\mathbf{u}_j)}{\sum_{j'=1}^K \exp(\mathbf{u}_{j'})}$$

这里涉及到矩阵乘法、求平均、以及非线性激活函数 softmax。softmax 又称归一化指数函数，是逻辑函数的一种推广，它能将一个含任意实数的 K 维的向量 \mathbf{z} 的“压缩”到另一个 K 维实向量，使得每一个元素的范围都在 $(0,1)$ 之间，并且所有元素的和为 1。在人工神经网络最后一层经常使用 softmax 函数作为分类函数，这些神经网络通常取对数损失函数或交叉熵损失函数，给出了多项 Logistic 回归的非线性变量。从 softmax 层得到的输出可以看做是一个概率分布。

一开始，权重矩阵是随机初始化的，一般需要通过定义损失函数对模型进行优化，才能得到矩阵 V 和 U 的参数。这个损失函数一般为交叉熵损失，用它衡量预测分布和实际分布的差异，并对差异通过梯度下降和反向传播算法进行学习优化，得到最终的词向量表示矩阵 V 和 U 。

交叉熵 (Cross Entropy) 是 Shannon 信息论中一个重要概念，主要用于度量两个概率分布间的差异性信息。语言模型的性能通常用交叉熵和复杂度 (perplexity) 来衡量。我们在第 8 章会讲到熵的概念。最小化优化问题和梯度下降法我们会在第 10-12 章来讲解。

在给出文本表示后，主要考虑使用两类方法对文本分类问题进行数学建模：传统方法和神经网络方法。在表示上，使用 TF-IDF 对文档进行表示，然后逻辑回归 (Logistics Regression, LR) 模型对文本分类进行数学建模为传统方法。在表示上，使用词向量 word2vec 对单词进行表示，然后使用循环神经网络 (Recurrent Neural Network, RNN) 对词向量特征进行进一步表达并用 softmax 映射输出进行非线性分类建模为神经网络方法。

逻辑回归是一种分类模型，它假设数据标签服从伯努利分布，使用条件概率 $P(y=1|x)$ 进行建模，其中 x 就是影评评论的 TF-IDF 表示，参数模型如下：

$$P(y=1|\mathbf{x}; \mathbf{w}) = \frac{\exp^{(\mathbf{w}^T \mathbf{x} + b)}}{1 + \exp^{(\mathbf{w}^T \mathbf{x} + b)}}$$

其中 \mathbf{w} 是权重参数向量，它的维数与 \mathbf{x} 的维数相同， b 是偏置项。对于逻辑回归的参数模型，使用“极大似然法”来构建对数损失：

$$L = -\frac{1}{m} \sum_{i=1}^m \ln(P(y_i|\mathbf{x}; \mathbf{w}))$$

其中：

$$P(y_i|\mathbf{x}; \mathbf{w}) = P(y=1|\mathbf{x}; \mathbf{w})^{y_i} (1 - P(y=1|\mathbf{x}; \mathbf{w}))^{1-y_i}$$

最后使用优化算法 (如梯度下降法) 对参数进行估计。

Word2Vec 是在大量无监督语料上使用浅层神经网络训练出来的词嵌入模型，它将单词映射成低维稠密向量，仅仅是缓解词了词语相似度的表达但是未能彻底解决语言学中的一词多义问题。因此基于深度神经网络的建模方法先通过深度网络对词向量进行进一步的特征抽取。在这里主要使用 RNN 来进行表示学习。

对于序列数据建模 (文本，语音，股票等)，RNN 引入了隐状态 h (hidden state) 的概念。经过 RNN 编码后， h 可以提取序列数据的特征。RNN 架构图如下图 1.16 所示：第 t 时刻的输入以

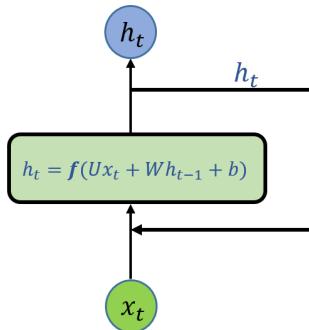


图 1.16: RNN 结构图

及第 \$t-1\$ 时刻的隐藏状态 \$h_{t-1}\$ 经非线性变换 \$f\$ 得到 \$h_t\$。

RNN 按时间展开可以得到下图1.17。在处理文本数据时,图1.17中的 \$\mathbf{x}_1\$ 可以看做是第一个单词的词向量,\$\mathbf{x}_2\$ 可以看做是第二个单词的词向量,依次类推在处理语音数据时,此时 \$\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\cdots,\mathbf{x}_n\$ 是每帧的声音信号隐藏状态 \$\mathbf{h}_i\$ 编码了第 \$i\$ 以及之前时刻的数据特征

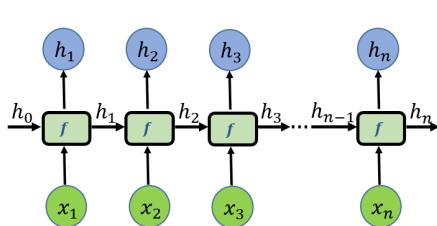


图 1.17: RNN 按时间步展开

在文本分类问题中,对于一个包含 \$n\$ 的单词的文本 \$W = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n)\$, 我们使用 RNN 对文本进行序列建模编码, 如下图1.18所示, 取第 \$n\$ 时刻的隐藏状态 \$\mathbf{h}_n\$ 来表示文本并使用其进行文本分类。

得到文本表示后,先使用线性变换对获得的特征进行加权组合,然后用 softmax 进行映射输出:

$$\begin{pmatrix} \text{logit}^{(0)} \\ \text{logit}^{(1)} \end{pmatrix} = \mathbf{G}\mathbf{h}_n + \mathbf{t} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1d} \\ g_{21} & g_{22} & \cdots & g_{2d} \end{bmatrix} \mathbf{h}_n + \begin{bmatrix} t_{11} \\ t_{21} \end{bmatrix}$$

$$P(\text{负类}|\mathbf{x}; \mathbf{w}) = P(y = 0|\mathbf{x}; \mathbf{w}) = \frac{\exp^{\text{logit}^{(0)}}}{\exp^{\text{logit}^{(0)}} + \exp^{\text{logit}^{(1)}}}$$

$$P(\text{正类}|\mathbf{x}; \mathbf{w}) = P(y = 1|\mathbf{x}; \mathbf{w}) = \frac{\exp^{\text{logit}^{(1)}}}{\exp^{\text{logit}^{(0)}} + \exp^{\text{logit}^{(1)}}}$$

传外切语情高稿草

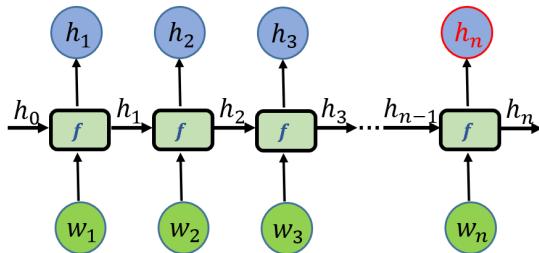


图 1.18: RNN 对文本进行编码

这里 \mathbf{G} 是 $2 \times d$ 的参数矩阵、 \mathbf{t} 是 2×1 的列向量， \mathbf{w} 是模型参数，由 RNN 中的 $\mathbf{U}, \mathbf{W}, \mathbf{b}$ 以及 softmax 分类层中的 \mathbf{G}, \mathbf{t} 组成 $\mathbf{w} = (\mathbf{U}, \mathbf{W}, \mathbf{b}, \mathbf{G}, \mathbf{t})$ 。

得到各个类别的概率后，使用“极大似然法”来构建对数损失：

$$L = -\frac{1}{m} \sum_{i=1}^m \ln(P(y_i | \mathbf{x}; \mathbf{w}))$$

其中：

$$P(y_i | \mathbf{x}; \mathbf{w}) = P(y = 1 | \mathbf{x}; \mathbf{w})^{y_i} (1 - P(y = 1 | \mathbf{x}; \mathbf{w}))^{1-y_i}$$

最后使用优化算法（如梯度下降法）对参数 $\mathbf{w} = (\mathbf{U}, \mathbf{W}, \mathbf{b}, \mathbf{G}, \mathbf{t})$ 进行估计。

可以看出，无论是传统方法还是深度学习方法，最后一步损失函数和优化问题可能是相同的，但是文本表示和中间的特征建模可能是不一样，而这也是自然语言处理中最重要的一部分。

这样我们就完成了图像和文本分类这样两个计算机视觉和自然语言处理任务，从数据驱动方法的角度看，这些任务最后都归结为数据分析中各种基本运算，如分类、回归、或降维等等，通常首先要把数据进行恰当的表示，然后进行任务建模和求解；实现这些运算的方法理论支撑是机器学习以及相应的数学基础，包括表示、建模和求解的过程中涉及向量、矩阵、概率分布、交叉熵和优化算法，这些内容来源于线性代数，概率和信息论、优化理论，而这正是本课程的核心内容。因此下一节我们将对机器学习做一个简要概览，并给出所需数学的具体框架。

1.3 从数据分析到数学基础

上一节我们已经通过两个例子讨论了数据分析和智能处理任务可以归结为数据分析中各种运算任务，如分类和降维等。我们把这种智能称为数据智能。机器学习其实是为数据智能提供重要的数据分析技术支撑。本节我们将对机器学习的理论背景以及其涉及的相关数学问题进行介绍，然后给出本课程相应的数学内容框架。我们首先给机器学习做一个概览，然后从数据、模型、学习三个角度来引出所需的数学基础。注意，我们这本书本质不是讲人工智能和机器学习，而是讲人工智能、机器学习和数据分析背后所需的数学基础，因此我们要对我们数学服务的领域背景有一个了解。

1.3.1 数据分析和机器学习概览

数据分析主要用于对数据的预测与分析，特别是对未知新数据的预测与分析。对数据的预测可以让计算机更加智能化，或者说是计算机的某些性能得到提高；对数据的分析可以让人们获取新的知识，给人们带来新的发现。

在数据分析中，我们假设存在一个未知的通用数据集，其中包含所有可能的数据对以及它们在现实世界中出现的概率分布。在实际应用中，由于内存不足或其他一些不可避免的原因，我们观察到的只是通用数据集的一个子集。此获取的数据集通常称为训练集（训练数据），用于学习通用数据集的属性和知识。数据分析的基本问题就是基于可获得的训练数据集，构建一个数学模型，通常是概率统计模型，不光用来刻画训练数据集中的数据关系，而且还能用于预测或发现未知数据之间的关系。数据分析总的目标就是考虑构建什么样的模型和如何构建模型，以使模型对数据进行准确的预测与分析，同时也要考虑尽可能的提高建模的效率。

在数据科学与工程领域，这种基于训练数据来构建模型并用于未知数据的预测和分析，可以归结机器学习，它是数据分析的核心。机器学习就是关于如何用计算机基于数据构建概率统计模型并运用模型对数据进行预测与分析的一门学科。应该说，这个定义只是机器学习一种定义而已，机器学习从上世纪 50 年代感知机被提出以来，到目前为止并没有一个统一的定义。

而近年来热门的深度学习和机器学习又有什么关系呢？粗略的说，深度学习是主要使用深度神经网络为工具的机器学习算法，也即通过多层非线性变换对高复杂度数据建模的算法的合集。深度学习是机器学习的一个研究分支，机器学习是人工智能的一部分，它们之间的关系如图1.19所示。

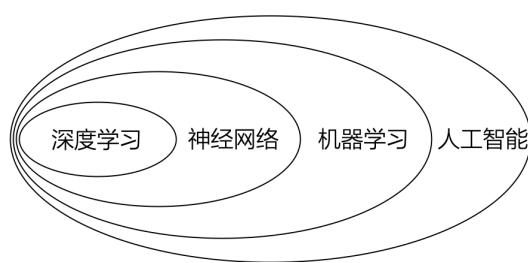


图 1.19: AI 中四个概念的包含关系

传统机器学习通常被称为浅层学习，深度学习属于深层学习，深度学习和机器学习的差异主要是在数据规模、模型深度和计算能力需求上的差异。

从数据科学的角度看，机器学习也是数据全生命周期的核心环节，在数据科学中具有重要的地位，如图1.20所示。

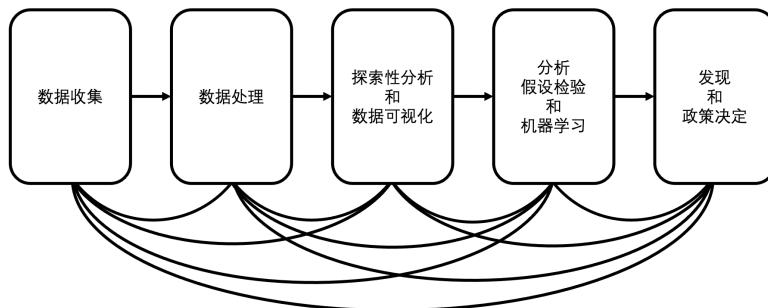


图 1.20: 数据分析与机器学习在数据全生命周期所处的阶段

下面我们来了解机器学习中的一些基本术语。我们通过一个商家对其客户进行分类的例子来考察机器学习的典型过程。给定一些数据，如图1.21左边表格这样一些客户情况数据，包括客户的基本信息特征和商家对其的类别标记，也即是否是好的客户？这些数据，我们称之为训练数据。商家希望从这些训练数据中训练出一个模型，以便来了一个新客户，能够对其进行预测分类，看是不是好的客户，从而为其提供相应的服务。这个模型根据训练数据的大小，可以建成传统的浅层机器学习模型，比如说决策树和支持向量机，也可以是深度神经网络；可以是概率模型，也可以是非概率模型，按照一定准则来建立和选取模型。在训练出模型后还要有测试数据来测试模型是不是好的模型，也就在新的数据上表现是否好，如果不好的话，我们还需要调整训练，这就是所谓的学习。

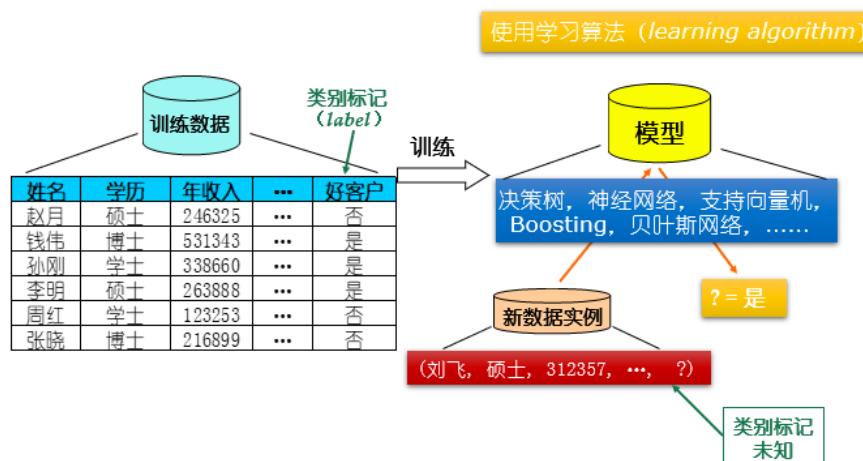


图 1.21: 典型的机器学习过程

从这个过程我们可以看出，一个机器学习系统主要由数据、模型和学习三部分组成，其中

数据包括训练数据和测试数据；模型包括确定性模型和不确定性模型，也对应于非概率模型和概率模型，学习部分包括模型选择的策略和模型学习的算法。其中，模型、策略和算法也称为机器学习方法的三要素。

据此，我们可以把机器学习方法可以概括如下：从给定的、有限的（在大数据时代，虽然数据规模很大，但大多数时候数据量总是有限的）、用于学习的训练数据集合出发，假设数据是独立同分布产生的；并且假设要学习的模型属于某个函数的集合，称为假设空间；应用某个评价准则，从假设空间中选取一个最优的模型，使他对已知的训练数据及未知的测试数据在给定的评价准则下有最优的预测；最优的模型选取由算法实现。

实现机器学习方法的步骤如下：（1）得到一个有限的训练数据集合；（2）确定包含所有可能的模型的假设空间，即学习模型的集合；（3）确定模型选择的准则，即学习的策略；（4）实现求解最优模型的算法，即学习的算法；（5）通过学习方法选择最优模型；（6）利用学习的最优模型对新数据进行预测和分析。

这里面预测和分析是机器学习的主要任务，也是大数据计算的主要任务。根据预测目标输出不同，可以分为：分类、回归、标注、聚类、降维和概率密度估计等。分类也即当输出变量取有限个离散值时，预测问题便成了分类问题，这时输出变量可以是连续变量，也可以是离散的。回归，当输出变量取连续值时，预测问题就成了回归问题。标注可以看成是分类的扩展，输入的是观测序列，输出是标记序列。聚类是数据实例集合当中相似的数据实例分配到相同的类，不相似的数据分配到不同的类。降维是将训练数据中的样本实例从高维空间转换到低维空间。概率密度估计简称概率估计，假设训练数据由一个概率模型生成，由训练数据学习模型的结构和参数，这几类任务都是无标记信息的。

实现这些任务按照是否从有无标记数据中学习，可以归结为监督学习、无监督学习和半监督学习等等，既包括众多经典的统计学习方法，如感知机、逻辑回归和支持向量机，也包括近年来火热的深度神经网络。

监督学习是指从有标记数据中学习预测模型的机器学习问题。标记数据表示输入输出的对应关系，预测模型对给定的输入产生相应的输出。监督学习的本质是学习输入到输出的映射的统计规律，这个映射以概率函数、代数函数或人工神经网络为基函数模型，采用迭代计算方法，学习结果为函数。监督学习方法的应用包括分类、标注与回归问题，这些方法在自然语言处理、信息检索、文本数据挖掘等领域有着极其广泛的应用。

无监督学习是指从无标记的数据中学习预测模型的机器学习问题，无标记数据是自然得到的数据，预测模型表示数据的类别、转换或概率。无监督学习的本质是学习数据的统计规律和潜在结构。无监督学习方法的应用主要包括聚类、降维、概率密度估计和图分析等。无监督学习可以用于数据分析或者监督学习的前处理。

半监督学习是指利用标记数据和未标记数据学习预测模型的机器学习问题。通常有少量标记数据，大量未标记数据，因为标记数据的构建往往需要人工，成本较高，未标记数据的收集不需要太多成本。半监督学习旨在利用未标记数据中的信息，辅助标记数据，进行监督学习，以

较低的成本达到较好的学习效果。

此外，还有主动学习和强化学习。主动学习是指机器不断主动给出实例让教师进行标记，然后利用标记数据学习预测模型的机器学习问题。通常的监督学习使用给定的标记数据，往往是随机得到的，可以看作是“被动学习”，主动学习的目标是找出对学习最有帮助的实例让教师标记，以较小的标记代价，达到较好的学习效果。主动学习和前面的半监督学习更接近监督学习。

强化学习是指智能系统在与环境的连续互动中学习最优行为策略的机器学习问题。假设智能系统与环境的互动基于马尔科夫决策过程，智能系统能观测到的是与环境互动得到的数据序列，强化学习的本质是学习最优的序贯决策。

机器学习的目标是找到好的模型，使得学到的模型能很好的适用于“未知的测试数据”，而不仅仅是训练数据，我们称模型适用于未知数据的能力为泛化(generalization)能力。一般而言训练数据越多越有可能通过学习获得强泛化能力的模型。

在给出了这些机器学习的基本术语之后，下面我们分别对机器学习系统中的数据、模型和学习三部分展开介绍。

1.3.2 数据

我们在1.1节大数据结构描述中已经提到数据科学中要处理的数据类型包括：图像、视频、文本、语音、网页、图数据、时间序列、以及传统的表格数据等，在1.2节我们已经处理过图像和文本数据。数据科学中我们面临的大数据通常具有高维、海量、多模、高速、噪声、稀疏和非平衡性等特性。这些特性都是我们建模时要根据具体数据情况进行考虑的，这里面最基本的就是如何根据数据类型和数据特性对数据进行表示。而且从前面我们对大数据的结构定义中可以看出，数据分析和机器学习处理任务首要的问题就是要对数据进行恰当的表示。数据表示包括数据表示为向量、输入数据和输出结果的表示和范围、输入数据变量和输出结果变量的基本假设三个部分。

1、数据表示为向量

我们可以考虑一个例子。比如有一个人力资源数据。假设数据按表格1.1存放，表的每一行表示某个人，每一列表示人的某个特征，如何把表格转换成可以由计算机读取并以数字表示的数据？

如果没有其他说明，应缩放数据集的所有列，使其均值为0和方差为1。

这里我们可以使用一些指导原则，比如：(1)首先可以将类变量转化为数字，在表1.1中性别列（类变量）可以被转换为表示“男性”的数字0和表示“女性”的1，或可以分别用数字-1,+1表示；(2)其次，可以运用领域知识，在构建表示时使用特定领域知识通常很重要，例如知道大学学位从学士学位到硕士学位到博士学位，或者知道邮政编码不仅仅是一串字符，而实际上是某一个区域的编码。在表1.2中，将表1.1中的数据转换为数字格式，每个邮政编码表示为两个数字，即纬度和经度；(3)还有就是利用合理的单位，可能直接读入机器学习算法的数值数据都

姓名	性别	学位	邮编	年龄	年薪
赵月	女	硕士	710001	34	246325
钱伟	男	博士	518051	44	531343
孙刚	男	学士	410013	52	338660
李明	男	硕士	100010	31	263888
周红	女	学士	150010	25	123253

表 1.1: 人力资源数据

性别	学位	纬度	经度	年龄	年薪 (万)
-1	2	34.2304	108.9343	34	24.6325
+1	3	22.5329	113.9303	44	53.1343
+1	1	28.2351	112.9313	25	33.8660
+1	2	39.9316	116.4101	52	26.3888
-1	1	45.7570	126.6425	31	12.3253

表 1.2: 转换后的人力资源数据

应该仔细考虑单位，合理缩放和约束。本例中，年薪在转化后可以以千为单位。在这样一些数据表示的指导原则下，我们可以将人力资源数据表转换成如表1.2这样计算机可读取的数据。比如性别就转换成-1, +1 这样一列数据，学位就用 1、2、3 来表示，邮编就用经纬度来表示，年薪全部转换成以千为单位。在转换成计算机读取的数据后，接下来如果我们要使用这些数据建立机器学习模型，我们需要给这些数据赋予数学结构。

假设特定的领域专家已经适当地转换了数据，我们知道表1.2中每一行都是代表某个人的特征，比如说第一行代表赵月的 6 个特征，每个人的所有特征形成一个一元的六维数组，作为计算机的输入。如果总共有 n 个人，每个人都 D 个特征的话，就形成了 n 个一元 D 维数组，我们把它记为 \mathbf{x}_n 。这个一元数组，我们把它称为向量，也即每个输入 \mathbf{x}_n 是 D 维向量，其被称为特征、属性或协变量。这样每个输入 \mathbf{x}_n 就是一个 D 维的向量，数据就表示为向量。除了人力资源数据的可以表示为向量外，其它复杂的结构化对象，例如，图像、句子、电子邮件消息、时间序列、分子形状和图形等也都可以表示成向量。

当数据集中 N 个输入 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 经过合适转换后的数据，按行排成一个 $N \times D$ 的二元数组，后面我们称之为矩阵，这些矩阵也被称为特征或属性矩阵，记作 $\mathbf{X} \in \mathbb{R}^{N \times D}$ 。这里 $\mathbb{R}^{N \times D}$ 表示 $N \times D$ 维向量空间，我们在第 2 章会介绍。特征矩阵每一行是某个个体 \mathbf{x}_n ，称为机器学习中的实例 (instance) 或数据点。一般，使用 N 来表示数据集中的实例数，并使用小写 $n = 1, \dots, N$ 来索引实例，下标 n 指的是数据集中总共 N 个实例中的第 n 个实例；使用 D 来表示数据集中

总的特征数，每列表示关注的特征，用 $d = 1, \dots, D$ 索引特征。我们刚刚介绍的这个表示只是输入数据的表示。

对于监督学习问题，还有一个与每个输入实例 \mathbf{x}_n 相关联的输出标签 y_n 。这时，数据集被写为一组实例标签对或输入输出对： $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n), \dots, (\mathbf{x}_N, y_N)\}$ ，实例标签对或输入输出对也称为样本或样本点。图1.22表示一维输入 x 和对应标签 y 的实例。

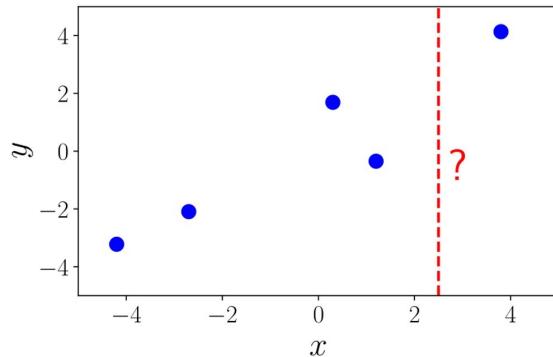


图 1.22: 线性回归的实例数据 (x_n, y_n) ：
 $\{(-4.200, -3.222), (-2.700, -2.093),$
 $(+0.300, +1.690), (+1.200, -0.348),$
 $(+3.800, +4.134)\}$, 注意 $x = 2.5$ 处的函数值不属于训练数据

对于无监督学习，通常使用大量的无标注数据学习或训练，这时每一个样本是一个实例。训练数据集表示为 $\mathbf{x}_1, \dots, \mathbf{x}_N$ ，其中 $\mathbf{x}_i, i = 1, 2, \dots, N$ 是样本。无监督学习每个输入是一个实例，由特征向量表示。每一个输出是对输入的分析结果，由输入的类别、转换或概率表示。模型可以实现对数据的聚类、降维或概率估计。

将数据表示为向量 \mathbf{x}_n 需要使用线性代数中的概念。数据表示为向量属于数据的代数表示。除了把数据表示为向量，我们还可以把数据表示为矩阵或更高阶的张量。此外，有些数据集具有隐含的对称性，这也可以用代数的方法表达出来。我们将在本书第 2 章中会详细介绍向量和矩阵的基本概念和运算。

注记 1. 因为数据用向量表示，所以很多时候我们可以处理数据来更好的表示数据。这主要有两种方式：找到原始特征向量的低维近似向量和使用原始特征向量的非线性高维组合。找到原始特征向量的低维近似向量本质上属于下面接下来要提到的无监督学习中的降维任务，可以通过主成分分析方法来实现。寻找主成分与第 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，我们将看到一个明确的特征映射 $\phi(\cdot)$ ，它允许我们使用更高维的表示 $\phi(\mathbf{x}_n)$ 来表示 \mathbf{x}_n 。高维表示可以将新特征构造为原始特征的非线性组合，可以使学习问题更容易。我们

在本书第 2 章也会讨论特征映射，并且展示该特征映射如何导向内核。近年来，深度学习方法 (Goodfellow 等, 2016) 已经显示出使用数据本身来学习这些特征的前景，并且在计算机视觉、语音识别和自然语言处理等领域已经非常成功。我们不会在本书的这一部分介绍神经网络，但读者可参考 5.6 节的反向传播的数学描述，那是训练神经网络的关键概念。

注记 2. 数据除了代数表示之外，还有图表示。比如社交网络数据，具有网络结构，可以用图来表示。有些数据本身没有图结构，但可以附加一个图结构。比方说度量空间的点集，我们可以根据点与点之间的距离来决定是否把两个点连接起来，这样就得到一个图结构。

注记 3. 在许多机器学习算法中，通常需要对数据进行表计，如比较两个向量的相关性或相似性，这需要用到一些几何度量，比如距离。我们在本书第 3 章和第 7 章我们会介绍计算两个实例之间的相似性或距离，具有相似特征的实例应该具有相似的输出或标签。两个向量的比较要求我们构造一个几何模型（在第 3 章中解释），并需要用第 10 章中的技术优化所得到的学习问题。

2、输入数据和输出结果的表示和范围

在监督学习中，将模型输入数据与输出结果的所有可能取值的集合，分别称为输入空间与输出空间，并且通常将输入实例 x_n 和输出标签 y_n 分别看作定义在输入空间和输出空间上的随机变量 X 和 Y 的取值。输入与输出空间可以是有限元素的集合，也可以是在集合上通过附加各种数学运算结构，如加法或数乘运算，变成一个基本的数学空间，最常见的就是欧氏空间。输入与输出空间，可以是同一个空间，也可以是不同的空间，但通常输出空间远远小于输入空间，甚至是输入空间的子空间。我们在第 2、3 章会给出欧氏空间和子空间的相关概念。

在监督学习中，每个具体的输入实例 x_n ，如果由特征向量表示，这时所有特征向量存在的空间称为特征空间，特征空间的每一维对应于一个特征；有时假设输入空间与特征空间为相同的空间，对它们不予区分；有时假设输入空间与特征空间为不同的空间，将实例从输入空间映射到特征空间，模型实际上都是定义在特征空间上的。

3、输入数据变量和输出结果变量的基本假设

在机器学习中，通常会将输入与输出看作是定义在输入（特征）空间与输出空间上的随机变量的取值。输入输出变量用大写字母表示，习惯上输入变量写作 X ，输出变量写作 Y 。输入与输入输出变量的取值，用小写字母表示，输入变量的取值写作 x ，输出变量的取值写作 y 。变量可以是标量和向量，都用相同类型字母表示。除特别声明外，本书中向量均为列向量。

输入变量 X 和输出变量 Y 有不同的类型，可以是连续的，也可以是离散的。可以根据输入输出变量的不同类型对预测任务给予不同的名称：当输入变量与输出变量均为连续变量的预测问题称为回归问题；当输出变量为有限个离散变量的预测问题称为分类问题；输入变量与输出变量均为变量序列的预测问题，称为标注问题。

对于监督学习，通常假设输入与输出的随机变量 X 和 Y 遵循联合概率分布 $P(X, Y)$ 。 $P(X, Y)$ 表示分布函数和分布密度函数。注意在学习过程中，假定这些联合概率分布存在，但对学习系

统来说，联合概率分布的具体定义是未知的。训练数据与测试数据被看作是依联合概率分布 $P(X, Y)$ 独立同分布产生的。特别统计机器学习假设数据存在一定的统计规律， X 和 Y 具有联合概率分布，就是监督学习关于数据的基本假设。

关于随机变量、联合概率分布等概念我们会在第 7 章给出。

从刚才数据表示的内容可以看出，这一部分主要涉及到线性代数和概率论，因此线性代数和概率论是数据表示的数学基础。

1.3.3 模型

获得数据的合适向量表示之后，我们就可以开始构建数据分析模型，模型是一个数据分析或机器学习系统最重要的部分。机器学习首要考虑的问题是学习什么样的模型。机器学习的模型可以分为非概率模型（也称为确定性模型）和概率模型，随具体的学习方法而定。

在监督学习中，非概率模型取函数形式 $y = f(x)$ ，概率模型取条件概率分布形式 $P(y|x)$ ，其中 x 是输入， y 是输出。在无监督学习中，非概率模型取函数形式 $z = g(x)$ ，概率模型取条件概率分布形式 $P(z|x)$ 或 $P(x|z)$ ，其中 x 是输入， z 是输出。在监督学习中，概率模型是生成模型，非概率模型是判别模型。

我们以后会知道，机器学习中常见的决策树、朴素贝叶斯、隐马尔可夫模型、条件随机场、概率潜在语义分析、潜在狄利克雷分配、高斯混合模型是概率模型。感知机、支持向量机、近邻、AdaBoost、左均值、潜在语义分析，以及神经网络是非概率模型。逻辑回归既可看作是概率模型，又可看作是非概率模型。

1、模型是函数

当模型是一种函数时，给定特定输入实例（特征向量）时，会生成输出。现在考虑将输出视为单个数字，即实值标量输出。这可以写作

$$f : \mathbb{R}^D \rightarrow \mathbb{R}, \quad (1.1)$$

其中输入向量 x 是 D 维（具有 D 个特征），函数 $f(x)$ 返回实数。图1.23表示一个可用于计算输入值 x 的预测值的函数。

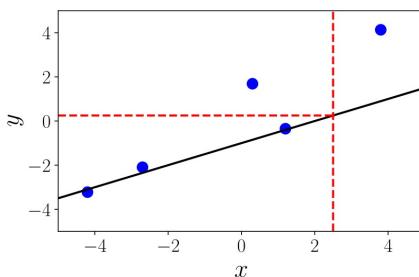
我们知道函数类型主要有线性函数和非线性函数。他们可以用来表示机器学习中的线性模型和非线性模型。我们后面会知道，机器学习中常见的感知机、线性支持向量机、左近邻、左均值、潜在语义分析都是线性模型。核函数支持向量机、AdaBoost、神经网络都是非线性模型。

深度学习（deep learning）实际是复杂神经网络的学习，也就是复杂的非线性模型的学习。

我们来看两个具体的例子。我们考虑仿射函数

$$f(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x} + \theta_0, \quad (1.2)$$

当 $\theta_0 = 0$ 时退化为标准的线性函数。仿射函数在平面上就是一条直线，如图1.23所示。仿射函数或线性函数表达的模型较为简单，但又具有一定的数据建模能力，所以仿射或线性函数在可

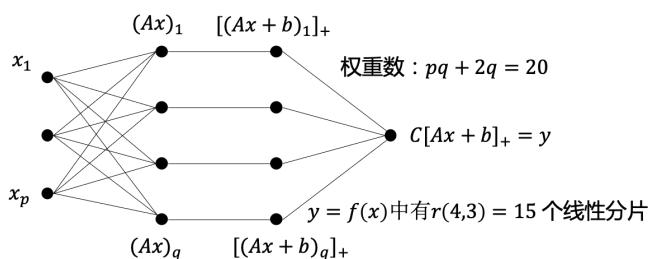
图 1.23: 实例函数在 $x = 2.5$ 时的预测: $f(2.5) = 0.25$.

以解决问题的一般性和所需的数学知识量之间取得了很好的平衡。但是很多时候数据中具有非线性特征，而线性函数不能表达数据的非线性特征，这时就要用非线性函数来建模。

考虑深度学习中的函数

$$f(\mathbf{x}) = f_L(f_{L-1}(\dots f_2(f_1(\mathbf{x}))). \quad (1.3)$$

其中 $f_i(\mathbf{x}) = \text{ReLU}(A_i \mathbf{x} + b_i) = (A_i \mathbf{x} + b_i)_+ = \max(A_i \mathbf{x} + b_i, 0)$ 是非线性函数，是非线性激活函数 ReLU 和仿射变换的复合。 ReLU 是神经网络中一个非常重要的非线性激活函数，定义了神经网络在线性变换后的输出。图 1.24 展示了数据向量 \mathbf{x} 的分段线性函数的神经网络构造。除了 ReLU ，还有 1.2 节提到的 softmax 也是非线性激活函数，我们在后续的章节还会详细介绍一些常用的非线性激活函数。

图 1.24: 数据向量 x 的分段线性函数的神经网络构造

函数建模是属于确定性建模。关于模型涉及的线性和非线性函数的性质我们在本书第 2 章、第 3 章、第 6 章和第 10 章都会提到。

2、模型是概率分布

我们经常认为数据是对某些真实潜在影响的噪声观察，希望通过应用机器学习可以识别来自噪声的信号。这要求有一种形式来量化噪声的影响。我们也希望有模型表达某种不确定性，例如，量化我们对特定测试数据点的预测值的置信度，这就需要引入概率，概率论提供了量化不

确定性的描述。概率建模的主要工具有：有限维参数的特殊分布，也即多元概率分布；和图的描述，也即概率图模型。

图1.25说明了函数作为高斯分布的预测不确定性。给定一些数据，我们可以利用线性回归对 $x=2.5$ 处的 y 值进行预测得到一个预测值。但是真实的 y 值实际上服从一个正态分布。我们试图预测出 y 最可能的取值。在黄点处概率最大，在其他预测值服从正态分布。

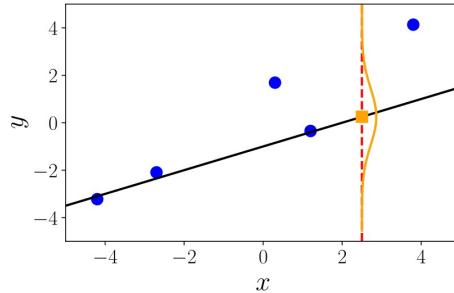


图 1.25: 实例函数 (黑色实心对角线) 及其在 $x=2.5$ 时的预测不确定性 (绘制为高斯分布)

我们将在本书第 7 章和第 9 章介绍概率的基础理论和相关模型，包括概率模型的图语言描述。

3、监督学习模型的假设空间

对于监督学习来说，学习的目的在于学习一个由输入到输出的映射，这一映射由模型来表示。换句话说，学习的目的就在于找到最好的这样的模型。监督学习的模型可以是概率模型和非概率模型，也即由条件概率分布 $P(X, Y)$ 或决策函数 $Y = f(X)$ 表示，随具体的学习方法而定。对具体的输入进行相应的输出预测时，写作 $P(y|x)$ 或 $y = f(x)$ 。

监督学习模型属于由输入空间到输出空间的映射集合，这个集合称为假设空间，用 \mathcal{F} 来表示。

假设空间可以定义为决策函数的集合：

$$\mathcal{F} = \{f | Y = f(X)\}, \quad (1.4)$$

其中， X 和 Y 是定义在输入空间 \mathcal{X} 和输出空间 \mathcal{Y} 上的变量。这时 \mathcal{F} 通常是由一个参数向量决定的函数族：

$$\mathcal{F} = f | Y = f_{\theta}(X), \theta \in \mathbb{R}^n, \quad (1.5)$$

参数向量 θ 取值于 n 维欧氏空间 \mathbb{R}^n ，称为参数空间。

假设空间也可以定义为条件概率的集合：

$$\mathcal{F} = P | P(Y|X), \quad (1.6)$$

其中， X 和 Y 是定义在输入空间 \mathcal{X} 和输出空间 \mathcal{Y} 上的随机变量。这时 \mathcal{F} 通常是由一个参数向量决定的条件概率分布族：

$$\mathcal{F} = P|P_\theta(Y|X), \theta \in \mathbb{R}^n, \quad (1.7)$$

参数向量 θ 取值于 n 维欧氏空间 \mathbb{R}^n ，也称为参数空间。

假设空间的确定意味着学习的范围的确定。假设空间因为是由函数或概率构成的空间，与数学中的泛函分析和概率分析的基本概念如范数、度量密切相关，我们将在第 3 章和第 7 章会提及。

本书中称由决策函数表示的模型为非概率模型，由条件概率表示的模型为概率模型。为了简便起见，当论及模型时，有时只用其中一种模型。

1.3.4 学习

学习的目标是找到一个模型及其相应的参数，使得模型在未知数据上表现良好。在讨论机器学习系统的学习部分时，有三个不同的学习阶段：(1) 训练或参数估计；(2) 超参数调整或模型选择；(3) 预测或推理。其中预测阶段是在未知的测试数据上使用经过训练的模型进行预测。换句话说，参数和模型选择已经固定，模型应用到表示新数据点的向量。根据预测模型是函数模型或者是概率模型，分别对应于机器学习的两个主要流派：优化方法流派和贝叶斯流派。当预测模型使用概率模型时，预测阶段称为推理。因此，在学习阶段，参数估计和模型选择是关键。这里会涉及到模型选择的策略。

1、策略

训练或参数估计阶段是根据训练数据调整预测模型，我们希望找到对训练数据表现良好的预测模型，因此我们需要考虑按照什么样的准则学习或选择最优的模型。前面我们已经定义了模型的假设空间，统计机器学习的目标在于从假设空间中选取最优模型。

这里主要有两种策略：根据某种质量指标找到最好的预测模型（有时称为寻找点估计）或使用贝叶斯推断。寻找点估计可用于函数模型和概率模型两种类型的预测模型，但贝叶斯推断只用于概率模型。对于非概率模型，我们遵循所谓的经验风险最小化准则，经验风险最小化提供了一个优化问题来寻找好的参数。对于统计模型，最大似然原理可以被用于找到一组好的参数。我们还可以使用贝叶斯推断或潜变量对概率模型中参数的不确定性进行建模。关于最大似然和贝叶斯推断在本书的第 7 章和第 9 章会涉及。

下面我们重点论述监督学习模型的选择策略——经验风险最小化准则。首先引入损失函数与风险函数的概念。损失函数度量模型一次预测的好坏，风险函数度量平均意义上模型预测的好坏。

1.1 损失函数和风险函数

监督学习问题是在假设空间 \mathcal{F} 中选取模型作为决策函数，对于给定的输入 X ，由 $f(X)$ 给出相应的输出 Y ，这个输出的预测值 $f(X)$ 与真实值 Y 可能一致也可能不一致，用一个损失函数

或代价函数来度量预测错误的程度。损失函数是 $f(X)$ 和 Y 的非负实值函数，记作 $L(Y, f(x))$ 。

统计机器学习常用的损失函数有以下几种：

(1) 0 – 1 损失函数 (0 – 1 loss function)

$$L(Y, f(X)) = \begin{cases} 1 & Y \neq f(X), \\ 0 & Y = f(X) \end{cases}, \quad (1.8)$$

(2) 平方损失函数

$$L(Y, f(X)) = (Y - f(X))^2, \quad (1.9)$$

(3) 绝对损失函数

$$L(Y, f(X)) = |Y - f(X)|, \quad (1.10)$$

(4) 对数损失函数或对数似然损失函数

$$L(Y, P(Y|X)) = -\log P(Y|X), \quad (1.11)$$

这些函数在很多机器学习模型中都有重要应用。比如在分类问题中，可以使用 0-1 损失函数的正负号来进行模式判断，函数值本身的大小并不是很重要，0 – 1 损失函数比较的是预测值 $f(x_i)$ 与真实值 y_i 的符号是否相同。其他损失函数都有类似相应的应用，我们在后面章节会介绍。

损失函数作为衡量预测值和真实值的误差当然是越小，模型就越好。那么这个误差到底有多大呢？怎么来衡量呢？由于模型的输入、输出 (X, Y) 是随机变量，遵循联合分布 $P(X, Y)$ ，所以损失函数的期望是

$$R_{exp}(f) = E_P[L(Y, f(X))] = \int_{\mathcal{X} \times \mathcal{Y}} L(y, f(x))P(x, y)dxdy, \quad (1.12)$$

这是理论上模型 $f(X)$ 关于联合分布 $P(X, Y)$ 的平均意义下的损失，称为风险函数或期望损失。

学习的目标就是选择期望风险最小的模型。但是由于联合分布 $P(X, Y)$ 是未知的， $R_{exp}(f)$ 不能直接计算。实际上，如果知道联合分布 $P(X, Y)$ ，可以从联合分布直接求出条件概率分布 $P(Y|X)$ ，也就不需要学习了。正因为不知道联合概率分布，所以才需要进行学习。这样一来，一方面根据期望风险最小学习模型要用到联合分布，另一方面联合分布又是未知的，所以从数学上看，监督学习就成为一个病态问题。这个问题可以通过概率中的大数定律以及经验风险最小化准则来解决。

1.2 经验风险和经验风险最小化准则

给定一个训练数据集

$$T = (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

模型 $f(X)$ 关于训练数据集的平均损失称为经验风险或经验损失，记作 R_{emp} ：

$$R_{emp}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)), \quad (1.13)$$

期望风险 $R_{exp}(f)$ 是模型关于联合分布的期望损失，经验风险 $R_{emp}(f)$ 是模型关于训练样本集的平均损失。根据大数定律，当样本容量 N 趋于无穷时，经验风险 $R_{emp}(f)$ 趋于期望风险

$R_{exp}(f)$ 。所以一个很自然的想法是用经验风险估计期望风险。但是，由于现实中训练样本数目有限，甚至很小，所以用经验风险估计期望风险常常并不理想，要对经验风险进行一定的矫正。这就关系到监督学习的一个基本策略：经验风险最小化准则。

在假设空间、损失函数以及训练数据集确定的情况下，经验风险函数式(6.1)就可以确定。经验风险最小化（empirical risk minimization, ERM）的策略认为，经验风险最小的模型是最优的模型。根据这一策略，按照经验风险最小化求最优模型就是求解最优化问题：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)), \quad (1.14)$$

其中， \mathcal{F} 是假设空间。

当样本容量足够大时，经验风险最小化能保证有很好的学习效果，在现实中被广泛采用。比如，极大似然估计就是经验风险最小化的一个例子。当模型是条件概率分布、损失函数是对数损失函数时，经验风险最小化就等价于极大似然估计。但是，当样本容量很小时，经验风险最小化学习的效果就未必很好，会产生“过拟合”现象。这时需要采用所谓的结构风险最小化准则或正则化来进行模型选择。下面我们来描述过拟合，以及所谓的结构风险最小准则和正则化，它作为经验风险最小化的补充，使其能够很好地概括最小化预期风险。

1.3 过拟合、结构风险最小化和正则化

回想一下，我们训练机器学习模型的目的是使我们能够很好地处理未知测试数据。这种未知测试数据称为测试集。假定预测器 f 有足够丰富的函数类，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的损失是很好的，但实际上，我们只有一组有限的数据，因此我们将数据分成训练和测试集。训练集用于拟合模型，测试集用于评估泛化性能。我们使用下标 $train$ 和 $test$ 来分别表示训练和测试集。

事实证明，经验风险最小化可能导致过度拟合，即预测与训练数据过于吻合，使其不能很好地推广到测试数据（Mitchell, 1997）。当我们具有很少的数据和复杂的假设函数类时，这种一般现象是具有非常小的训练损失但是有很大的测试损失。对于特定模型 f （参数固定），当来自训练数据 $\mathbf{R}_{emp}(f, \mathbf{X}_{train}, \mathbf{y}_{train})$ 的风险估计低估期望风险 $\mathbf{R}_{true}(f)$ 时，会发生过度拟合现象。由于我们通过使用测试集 $\mathbf{R}_{emp}(f, \mathbf{X}_{test}, \mathbf{y}_{test})$ 上的经验风险来估计期望风险 $\mathbf{R}_{true}(f)$ ，如果测试风险远大于训练风险，这表明是过度拟合。

因此，我们可以通过引入所谓的结构风险最小化策略来防止过拟合。结构风险最小化等价于正则化。结构风险是在经验风险上加上表示模型复杂度的正则化项或罚项。在假设空间、损失函数以及训练数据集确定的情况下，结构风险定义为：

$$R_{srm}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f), \quad (1.15)$$

其中 $J(f)$ 为模型的复杂度，是定义在假设空间 \mathcal{F} 上的泛函。模型 f 越复杂，复杂度 $J(f)$ 就越大；反之，模型 f 越简单，复杂度 $J(f)$ 就越小。也就是说，复杂度表示了对复杂模型的惩罚。 $\lambda \geq 0$ 是系数，用以权衡经验风险和模型复杂度。结构风险小需要经验风险与模型复杂度同时小。结构风险小的模型往往对训练数据以及未知的测试数据都有较好的预测。

比如，贝叶斯估计中的最大后验概率估计就是结构风险最小化的一个例子。当模型是条件概率分布、损失函数是对数损失函数、模型复杂度由模型的先验概率表示时，结构风险最小化就等价于最大后验概率估计。

结构风险最小化的策略认为结构风险最小的模型是最优的模型。所以求最优模型，就是求解最优化问题：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f), \quad (1.16)$$

上述最优化问题一般也称为正则化。因此，正则化是结构风险最小化策略的实现。其中正则化项或惩罚项 $\lambda J(f)$ 用来以某种方式偏向于寻找经验风险的最小化，这使得优化问题更难以返回过于灵活的模型。正则化项可以取不同的形式，一般是模型复杂度的单调递增函数，模型越复杂，正则化值就越大。比如，正则化项可以是模型参数向量的范数。例如，回归问题中，损失函数是平方损失，正则化项可以是参数向量的 L_2 范数：

$$L(w) = \frac{1}{N} \sum_{i=1}^N (f(x_i; w) - y_i)^2 + \frac{\lambda}{2} \|w\|^2, \quad (1.17)$$

这里， $\|w\|$ 表示参数向量 w 的 L_2 范数。正则化项也可以是参数向量的 L_1 范数：

$$L(w) = \frac{1}{N} \sum_{i=1}^N (f(x_i; w) - y_i)^2 + \frac{\lambda}{2} \|w\|_1, \quad (1.18)$$

这里， $\|w\|$ 表示参数向量 w 的 L_1 范数。第 1 项的经验风险较小的模型可能较复杂（有多个非零参数），这时第 2 项的模型复杂度会较大。正则化的作用是选择经验风险与模型复杂度同时较小的模型。

这样，监督学习问题就变成了经验风险或结构风险函数的最优化问题(6.4)和(1.16)。这时经验或结构风险函数是最优化的目标函数。

上面主要是针对监督学习的策略。因为无监督学习的基本任务主要包括聚类、降维和概率模型估计等，所以对于无监督学习模型的策略，在不同的问题中有不同的形式，但也都可以表示为目标函数的优化。比如，聚类中样本与所属类别中心距离的最小化，降维中样本从高维空间转换到低维空间过程中信息损失的最小化，概率模型估计中模型生成数据概率的最大化。我们考虑如下线性回归和 PCA 降维的例子，如图1.26所示。

对于一个监督学习问题，设其数据集为 $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ ，以一元线性回归作为模型，根据经验风险最小化可得：

$$\min_{(k,b) \in \mathbb{R}^2} \frac{1}{N} \sum_{i=1}^N |kx_i + b - y_i|$$

对于一个无监督学习问题，设其数据集为 $\{(x_1^1, x_1^2), (x_2^1, x_2^2), \dots, (x_n^1, x_n^2)\}$ ，使用 PCA 对其降维，将原来位置到新位置的距离看做信息损失（即原来位置到 1 维直线的距离）可得：

$$\min_{(a,b,c) \in \mathbb{R}^3} \frac{1}{N} \sum_{i=1}^N \frac{|ax_i^{(1)} + bx_i^{(2)} + c|}{\sqrt{a^2 + b^2}}.$$

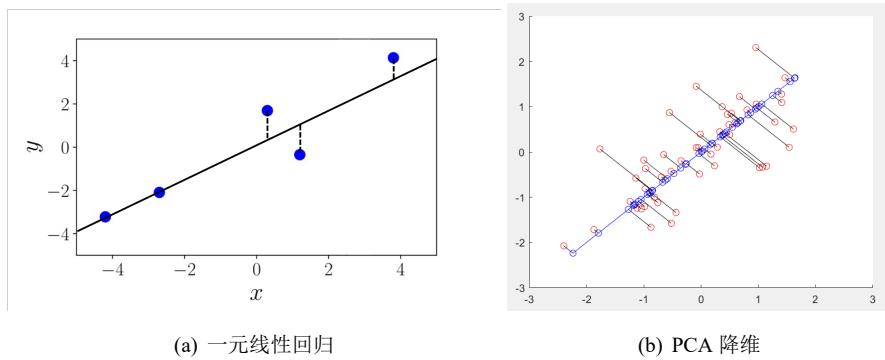


图 1.26: 监督学习与无监督学习

从上面的经验风险最小化策略和结构风险最小化策略中，我们可以看出利用经验风险代替期望风险的数学理论基础就是概率统计中的大数定律，我们将在本书第7章进行详细的介绍。然后要使得经验风险最小或结构风险最小，这里需要求解最优化问题，其中最优化问题的目标函数是由各种向量或矩阵损失函数和正则化项构成的，正则化项通常可以是模型参数向量或参数矩阵的范数。关于范数、损失函数和目标函数的有关定义、性质和计算，包括微分，将在本书第3章和第6章以及第10章介绍。

注记 4. 机器学习中还有一种常用的模型选择策略是交叉验证 (cross validation)。如果给定的样本数据充足，进行模型选择的一种简单方法是随机地将数据集切分成三部分，分别为训练集 (*training set*)、验证集 (*validation set*) 和测试集 (*test set*)。训练集用来训练模型，验证集用于模型的选择，而测试集用于最终对学习方法的评估。在学习到的不同复杂度的模型中，选择对验证集有最小预测误差的模型。由于验证集有足够的数据，用它对模型进行选择也是有效的。但是，在许多实际应用中数据是不充足的。为了选择好的模型，可以采用交叉验证方法。交叉验证的基本想法是重复地使用数据：把给定的数据进行切分，将切分的数据集组合为训练集与测试集，在此基础上反复地进行训练、测试以及模型选择。

2、算法

算法是指学习模型的具体计算方法。统计学习基于训练数据集，根据学习策略，从假设空间中选择最优模型，最后需要考虑用什么样的计算方法求解最优模型。这时，统计机器学习问题归结为最优化问题，统计学习的算法成为求解最优化问题的算法。如果最优化问题有显式的解析解，这个最优化问题就比较简单。但通常解析解不存在，这就需要用数值计算的方法求解。如何保证找到全局最优解，并使求解的过程非常高效，就成为一个 important 问题。统计学习可以利用已有的最优化算法，有时也需要开发独自的最优化算法。这里需要考虑优化问题是不是凸的，是不是精确可解，有没有对偶等。算法通常是迭代算法，通过迭代达到目标函数的最优化，比

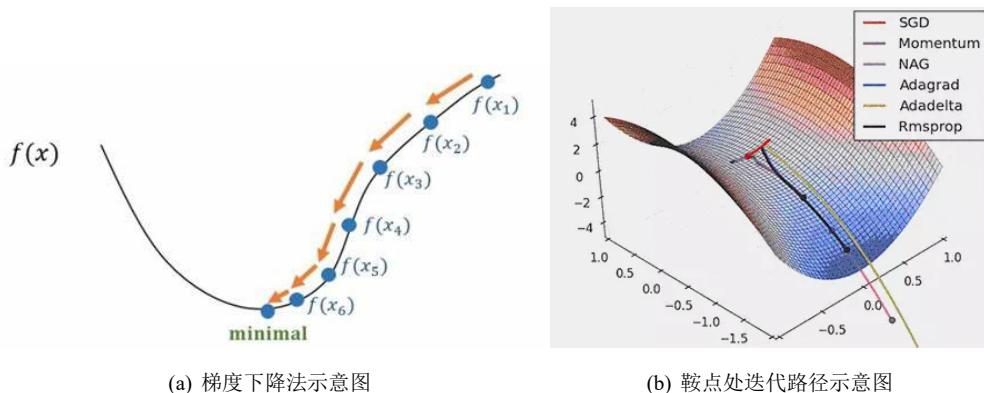


图 1.27: 优化算法

如，梯度下降算法、随机梯度下降算法、牛顿法等等。对于梯度下降法，简单说，就是沿负梯度寻找方向迭代的寻找函数值最小的点。在梯度下降算法中，一个优化算法中要包含三个要素，起点，步长，以及下降方向。三要素的选取决定了算法表现是否良好。梯度下降的迭代公式是：

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k \nabla f(\mathbf{x}^{(k)}).$$

图1.27给出了梯度下降和在深度学习中一些常用的优化算法在鞍点处的迭代路径示意图。

关于最优化的求解理论和方法将在本书第10章至12章进行详细的介绍。因为目标函数通常是向量函数和矩阵函数，所以优化问题求解会涉及矩阵的分解和方程组的求解以及向量函数和矩阵函数的微分，这些内容将在本书第4章至第6章详细介绍。

3、模型评估、泛化能力

统计机器学习的目的是使学到的模型不仅对已知数据而且对未知数据都能有很好的预测能力。不同的学习方法会给出不同的模型。当损失函数给定时，基于损失函数的模型的训练误差和模型的测试误差就自然成为学习方法评估的标准。注意，统计学习方法具体采用的损失函数未必是评估时使用的损失函数。当然，让两者一致是比较理想的。

假设学习到的模型是 $Y = \hat{f}(x)$ ，训练误差是模型 $Y = \hat{f}(x)$ 关于训练数据集的平均损失：

$$R_{emp}(\hat{f}) = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}(x_i)), \quad (1.19)$$

其中 N 是训练样本容量。

测试误差是模型 $Y = \hat{f}(x)$ 关于测试数据集的平均损失：

$$e_{test} = \frac{1}{N'} \sum_{i=1}^{N'} L(y_i, \hat{f}(x_i)), \quad (1.20)$$

其中 N' 是测试样本容量。

训练误差的大小，对判断给定的问题是不是一个容易学习的问题是有意义的，但本质上不重要。测试误差反映了学习方法对未知的测试数据集的预测能力，是学习中的重要概念。显然，给定两种学习方法，测试误差小的方法具有更好的预测能力，是更有效的方法。通常将学习方法对未知数据的预测能力称为泛化能力。

学习方法的泛化能力是指由该方法学到的模型对未知数据的预测能力，是学习方法本质上重要的性质。现实中采用最多的方法是通过测试误差来评价学习方法的泛化能力。但这种评价是依赖于测试数据集的。因为测试数据集是有限的，很有可能由此得到的评价结果是不可靠的。统计学习理论试图从理论上对学习方法的泛化能力进行分析，并定义了泛化误差。也就，如果学到的模型是 $Y = \hat{f}(x)$ ，那么用这个模型对未知数据预测的误差即为泛化误差

$$R_{exp}(\hat{f}) = E_P[L(Y, \hat{f}(X))] = \int_{\mathcal{X} \times \mathcal{Y}} L(y, \hat{f}(x)) P(x, y) dx dy, \quad (1.21)$$

泛化误差反映了学习方法的泛化能力，如果一种方法学习的模型比另一种方法学习的模型具有更小的泛化误差，那么这种方法就更有效。事实上，泛化误差就是所学到的模型的期望风险。

学习方法的泛化能力分析往往是通过研究泛化误差的概率上界进行的，简称为泛化误差上界。具体来说，就是通过比较两种学习方法的泛化误差上界的大小来比较它们的优劣。泛化误差上界通常具有以下性质：它是样本容量的函数，当样本容量增加时，泛化上界趋于 0；它是假设空间容量 (capacity) 的函数，假设空间容量越大，模型就越难学，泛化误差上界就越大。关于统计机器学习方法泛化误差上界的估计和证明常常会用到概率不等式，比如 Hoeffding 不等式等。本书我们对这些误差上界的估计的结果和证明不作详细介绍，读者可以参考瓦普尼克的《统计学习理论》一书。

机器学习的学习过程主要包括模型选择的策略和算法求解两部分。此外，也涉及模型评估和泛化能力的考量。一般统计机器学习方法之间的不同，主要来自其模型、策略、算法的不同。确定了模型、策略和算法，统计机器学习的方法也就确定了。这就是将其称为统计学习方法三要素的原因。统计机器学习方法的三要素，再加上数据，就构成了一个机器学习系统主要的要素。

1.3.5 机器学习的应用

我们之前把大数据的运算结构定义为分析数据时所要实现的各种计算任务，包括分类、聚类、回归、排序、降维和密度估计等，它们都属于机器学习的各种应用。具体上，监督学习的应用主要在三个方面：分类问题、标注问题和回归问题。

1) 分类运算。通过训练集训练出来一个模型，用于判断新输入数据的类型，即找一个函数判断输入数据所属的类别，而在训练的过程中，一定需要有标签的数据，即训练集本身就带有标签。简单来说，用已知的数据来对未知的数据进行划分。这是一种有监督学习。分类可以是二类别问题（是/不是），也可以是多分类问题（在多个类别中判断输入数据具体属于哪一个类

别)。分类问题的输出不再是连续值，而是离散值，用来指定其属于哪个类别。分类问题在现实中应用非常广泛，比如垃圾邮件识别，手写数字识别，人脸识别，语音识别等。

2) 标注运算。标注 (tagging) 也是一个监督学习问题。可以认为标注问题是分类问题的一个推广，标注问题又是更复杂的结构预测问题的简单形式。标注问题的输入是一个观测序列，输出是一个标记序列或状态序列。标注问题的目标在于学习一个模型，使它能够对观测序列给出标记序列作为预测。注意，可能的标记个数是有限的，但其组合所成的标记序列的个数是依序列长度呈指数级增长的。

3) 回归运算。回归是从一组数据出发，确定某些变量之间的定量关系式；即建立数学模型并估计未知参数。回归的目的是预测数值型的目标值，它的目标是接受连续数据，寻找最适合数据的方程，并能够对特定值进行预测。这个方程称为回归方程，而求回归方程显然就是求该方程的回归系数，求这些回归系数的过程就是回归。

回归分析，即量化因变量受自变量影响的大小，建立线性回归方程或者非线性回归方程，从而达对因变量的预测，或者对因变量的解释作用。

分类和回归的区别在于输出变量的类型。定量输出称为回归，或者说是连续变量预测；定性输出称为分类，或者说是离散变量预测。举个例子：预测明天的气温是多少度，这是一个回归任务；预测明天是阴、晴还是雨，就是一个分类任务。

无监督学习的主要应用主要在三个方面：聚类、降维和密度估计。

1) 聚类运算。聚类将数据集中的样本划分为若干个通常是不相交的子集，每个子集称为一个簇 (cluster)，每个簇对应一个潜在概念或类别。当然这些类别在执行聚类算法之前是未知的，聚类过程是自动形成簇结构，簇所对应的概念语义由使用者命名。

聚类和分类是有区别的。对于一组数据，若根本不知道数据之间的关系，不知道他们是否属于同一类，抑或属于不同类别，也不知道到底可以分为多少类。这个时候，就需要聚类算法来对数据进行一个关系分析，通过聚类，我们可以把未知类别的数据，分为一类或者多类，这个过程是不需要标签的，这是一种无监督学习。

聚类既能作为一个单独过程，用于寻找数据内在的分布结构，也可作为分类等其他学习任务的前驱过程。如在一些商业应用中需对新用户的类型进行判别，但定义用户类型对商家来说可能不太容易，此时可先对用户进行聚类，根据聚类结果将每个簇定义为一个类，然后再基于这些类训练分类模型，用于判别新用户的类型。

2) 降维运算。降维就是指采用某种映射方法，将原高维空间中的数据点映射到低维度的空间中。降维的本质是学习一个映射函数 $f : x \mapsto y$ ，其中 x 是原始数据点的表达，目前多使用向量表达形式， y 是数据点映射后的低维向量表达，通常 y 的维度小于 x 的维度（当然提高维度也是可以的）。 f 可能是显式的或隐式的、线性的或非线性的。

目前大部分降维算法处理向量表达的数据，也有一些降维算法处理高阶张量表达的数据。之所以使用降维后的数据表示是因为在原始的高维空间中，包含有冗余信息以及噪声信息，在实际应用例如图像识别中造成了误差，降低了准确率；而通过降维，我们希望减少冗余信息所造成

的误差，提高识别（或其他应用）的精度。又或者希望通过降维算法来寻找数据内部的本质结构特征。在很多算法中，降维算法成为了数据预处理的一部分，如 PCA。事实上，有一些算法如果没有降维预处理，其实是很难得到很好的效果的。

3) 密度估计。密度估计是机器学习的基本问题之一，其目的是根据训练样本确定样本 x 的概率分布。密度估计包括参数估计与非参数估计。当我们把机器学习应用于数据时，我们通常想要用某种方式表示数据。一种直接的方法是用数据点本身来表示数据。然而，如果数据集很大，或者我们对表示数据的特征很感兴趣，这种方法可能没有帮助。在密度估计中，我们用密度来紧凑地表示数据，例如高斯分布或贝塔分布。例如，我们可能在寻找一个数据集的均值和方差，以便用高斯分布函数紧凑地表示数据。均值和方差可以使用极大似然或极大后验估计来得到。然后我们可以用高斯分布的均值和方差来表示数据背后的分布。比如如果我们要从中进行采样，我们认为数据集是这种分布的典型实现。

无监督学习更综合，与领域相关的应用还包括话题分析和图分析。

(4) 话题分析。话题分析是文本分析的一种技术。给定一个文本集合，话题分析旨在发现文本集合中每个文本的话题，而话题由单词的集合表示。注意，这里假设有足够数量的文本，如果只有一个文本或几个文本，是不能做话题分析的。话题分析可以形式化为概率模型估计问题，或降维问题。

(5) 图分析。很多应用中的数据是以图的形式存在，图数据表示实体之间的关系，包括有向图、无向图、超图。图分析的目的是发掘隐藏在图中的统计规律或潜在结构。链接分析是图分析的一种，包括 PageRank 算法，主要是发现有向图中的重要结点。PageRank 算法是无监督学习方法。

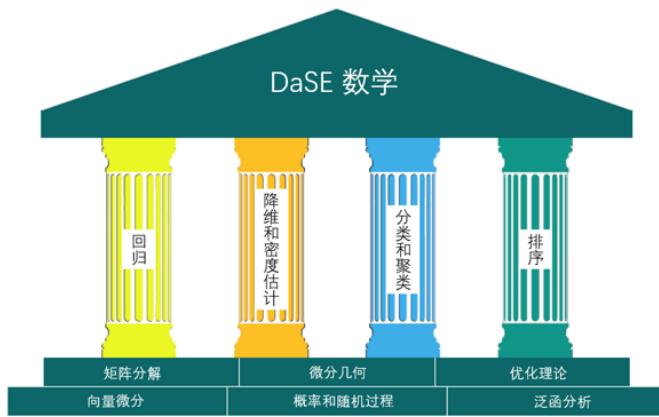
与话题分析和 PageRank 的计算有关的例子将在本书的第 2 章和第 5 章会进行详细介绍。除了上述监督学习和无监督任务外，还有一些计算任务可以建模成不同形式的问题。比如排序运算。

排序或机器学习排序 (MLR) 是指应用机器学习为信息检索系统构建排序模型，通常通过一个二次排序函数实现。排序学习可以是监督、半监督或强化学习，用于构建信息检索系统的排名模型。训练数据通常为包含部分排序信息的列表，该排序通常表示为对每个物体都使用一个数字或序号表示的分数，或者是二元判断（相关或不相关）。排序模型的最终目的是得到可靠的排序，即便列表中的物体未曾出现过。常用的排序学习方法主要有：逐个的 (PointWise)，逐对的 (PairWise) 和逐列的 (ListWise)。

1.4 数据分析和机器学习所需数学内容框架

由上述机器学习概览我们可知，在对数据建立了模型之后，模型求解大多被定义为一个优化问题或后验抽样问题，具体地，频率派方法其实就是一个优化问题。而贝叶斯模型的计算则

往往牵涉蒙特卡罗 (Monte Carlo) 随机抽样方法。因此数据科学与工程或机器学习的数学基础主要依赖于以矩阵分析、概率统计和优化为主的数学体系。除此之外，还涉及到更高等的数学基础，如统计机器学习所需的泛函分析基础，特别是再生核希尔伯特空间和 Mercer 定理；用于描述数据高维结构的几何基础，包括张量、拓扑学和微分流形（嵌入定理）以及随机过程。这些内容非常多，散落在数学的各个不同分支的教材里面，不方便一本书一本书的去学习，需要设计一本新的类似于计算机科学中“离散数学”这样的统一的“数据科学与工程数学基础”教材来覆盖这些方面的内容（如图1.28所示）。



本书就是针对这个需求进行设计的。全书的内容组织结构如图1.29所示：第一章是绪论，接下来是线性代数和矩阵计算部分：包括向量和矩阵基础，度量与投影，矩阵分解，矩阵计算问题，向量和矩阵微分；然后就是概率和信息论部分：包括概率基础、信息论基础、概率模型和参数估计。第三部分是优化理论，包括优化基础、最优化条件和对偶理论、优化算法等。线性代数可用于数据表示，概率和信息论可以用于描述数据的随机分布关系，这两部分一起为数据的表示和数学模型提供了数学基础；线性代数也是概率和优化部分内容的基础！优化理论部分则提供了数据的数值优化模型和方法。

注意本书主要包括数据的低维表示、数据的概率和随机表示、数据的数值优化方法，主要面向数据科学与工程专业的本科生。对于包括数据的高维几何表示、随机过程和高等优化算法等，我们这里并不涉及，我们计划在未来《数据科学与工程的高等数学基础》这本教材中来介绍这些内容。

本章剩余的部分将对全书涉及的主要概念提供一个简要概览，并对相关内容所涉及的学科做一个简要的历史回顾。大多数概念在这里是非正式的，更严格的定义和例子描述将在随后的章节中详细给出。

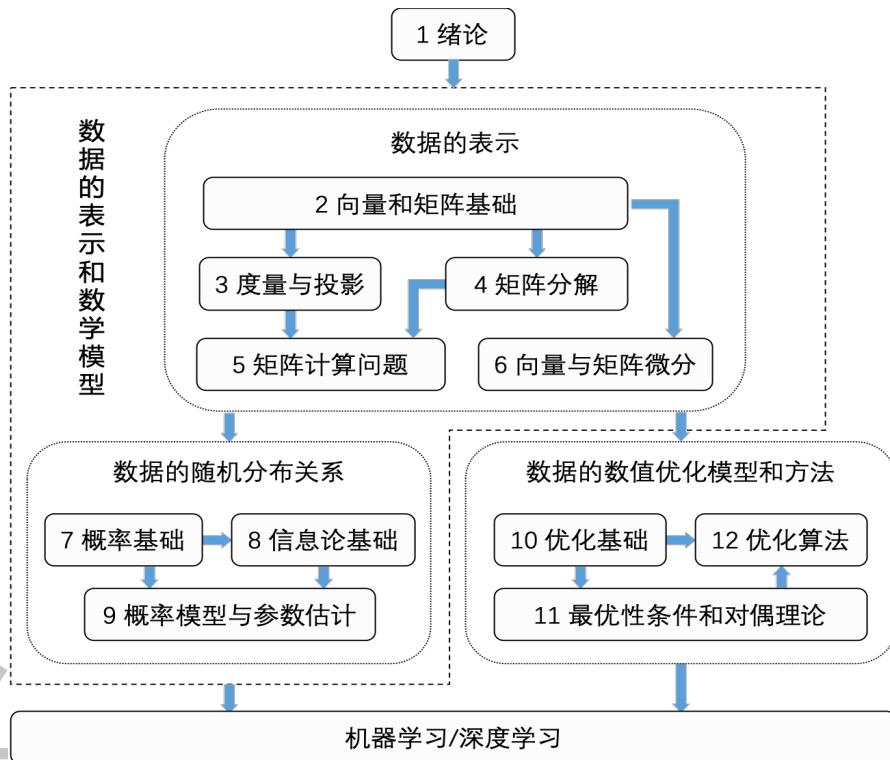


图 1.29: 数据科学与工程的数学基础内容组织结构流程图

1.5 数据科学与工程数学的历史

正如我们在 1.1 节所提到的那样，数据科学与工程的数学基础涉及到几乎所有数学的分支，包括代数、几何、分析和概率的理论与计算方法，因此我们不能对涉及到的所有数学的发展做一个简要的历史回顾。下面主要对本书所涉及的数学知识，包括线性代数、概率和优化的早期历史做一个简要的介绍。

1.5.1 早期阶段：线性代数的诞生

线性代数作为一个涉及解决数值问题的算法的领域，它的起源或许可以追溯到中国古代方程。与本书第五章提到的求解线性方程组的高斯消去法相同的方法早在公元一世纪的中国古代数学经典《九章算术》中就出现了，被称为直除法。

图1.30是 16 世纪出版物中的 9×9 矩阵。

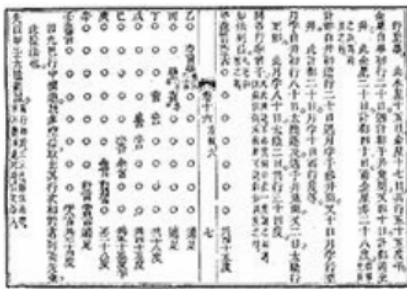


图 1.30: 古代中国的线性代数文本

1.5.2 概率论的起源

概率论是一门研究随机现象的数学规律的学科。它起源于十七世纪中叶，来自赌徒的问题刺激着当时的数学家们思考概率问题。费马、帕斯卡、惠更斯等首先对这个问题进行了研究与讨论，科尔莫戈罗夫等数学家对它进行了公理化。后来，由于社会和工程技术问题的需要，促使概率论不断发展，隶莫弗、拉普拉斯、高斯等著名数学家对这方面内容都进行了研究。概率论发展到今天，和以它作为基础的数理统计学科一起，在自然科学，社会科学，工程技术，军事科学及生产生活实际等诸多领域中起着不可替代的作用。

1.5.3 优化作为理论工具

在 19 世纪，高斯建立在线性代数的早期结果上来创造了一种求解最小二乘问题的方法，该方法依赖于求解相关的线性方程（即正规方程）。他用这种方法准确预测了小行星 Ceres 的轨迹。

在 17 世纪和 19 世纪之间，优化问题对理论力学和物理学的发展至关重要。大约在 1750 年，Maupertuis 引入最小作用原理，根据该原理，自然系统的运动可以被描述为涉及“能量”的某种成本函数的最小化问题。这种基于优化的（或变分的）方法是经典力学的基础。

意大利数学家 Giuseppe Lodovico (Luigi) Lagrangia，也称拉格朗日，是这一发展的关键人物，他的名字与优化中的核心概念对偶有关。优化理论在物理学中发挥了核心作用。随着计算机的诞生，它开始进入物理学以外的领域，在各种实际应用中发挥重要作用。

1.5.4 数值线性代数的出现

随着计算机在 40 年代后期问世，数值线性代数飞速发展。早期的贡献者包括 Von Neumann, Wilkinson, Householder 和 Givens。

早期的挑战是算法不可避免地传播数值误差。这导致了对算法稳定性和相关扰动理论的大量研究活动。在这种背景下，研究人员认识到某些自 19 世纪起物理领域遗留下来的某些问题求

得数值解的困难，例如一般方阵的特征值分解。最近产生的分解算法，例如奇异值分解，被认为在许多应用中起着核心作用。

优化在线性代数的发展中起着关键作用。在 70 年代，实用的线性代数与软件有着密切联系。用 FORTRAN 编写的高效软件包，例如 LINPACK 和 LAPACK，在 80 年代推出。这些软件包后来被应用到并行编程环境中。线性代数的一个关键发展阶段是科学计算平台的出现，如 Matlab, Scilab, Octave, R 等。这些平台将早期开发的 FORTRAN 软件包隐藏在用户友好的界面之后，并且使用非常接近自然数学符号的编码符号来解决线性方程式变得非常容易。

线性代数的相关应用已经有很多成功案例。PageRank 算法由著名的搜索引擎用于对网页进行排名，它依赖于幂法算法来解决特殊类型的特征值问题。目前数值线性代数领域的大部分研究工作涉及解决超大规模问题。两个研究方向很普遍。一个涉及解决分布式平台上的线性代数问题。另一项重要工作涉及采样算法。

1.5.5 线性和二次规划的出现

线性规划模型由 George Dantzig 在 40 年代引入，涉及军事领域中的 0-1 规划问题。将线性代数的范围扩展到不等式产生了著名的单纯形算法。线性规划的另一个重要的早期贡献者是苏联数学家 Leonid Kantorovich。

二次规划在许多领域都很受欢迎，例如金融，其中目标中的线性项是指投资的预期负收益，而平方项对应于风险（或收益的方差）。该模型由 H. Markowitz（他当时是兰德公司的 Dantzig 的同事）在 50 年代引入，以模拟投资问题。马科维茨在 1990 年因此获得诺贝尔经济学奖。在 60 年代到 70 年代，很多注意力都集中在非线性优化问题上。提出了寻找局部最小值的方法。与此同时，研究人员认识到这些方法不能找到全局最小值，甚至无法收敛。因此，当时认为，线性优化在数值上易于处理，而一般非线性优化不是。这有具体的实际后果：线性编程求解器可以可靠地用于日常操作（例如，用于航空公司机组人员管理），但非线性求解器需要专家对他们进行测试。在 60 年代，凸分析随着其发展成为优化进展的重要理论基础。

1.5.6 凸规划的出现

在 60 年代至 80 年代，美国的大多数优化研究都集中在非线性优化算法和应用上，苏联则将研究重点更多地放在优化理论上。由于非线性问题很难，苏联研究人员回到线性规划模型，并在理论上研究如下问题：什么使线性程序变得容易？它是否真的是客观和约束函数的线性，还是其他一些更通用的结构？是否存在非线性但仍易于解决的问题？

在 80 年代后期，苏联的两位研究人员 Yurii Nesterov 和 Arkadi Nemirovski 发现，使优化问题“容易”的一个关键特性不是线性，而是实际的凸性。他们的结果不仅是理论上的，而且是算法上的，因为他们引入了所谓的内点方法来有效地解决凸问题。粗略地说，凸问题很容易（包括线性规划问题）而非凸的很难。其实并非所有的凸问题都很容易，但它们的（相当大的）子集

是容易的。相反，只有少部分一些非凸问题实际上很容易解决（例如一些路径规划问题可以在线性时间内解决）。自 Nesterov 和 Nemirovski 的开创性工作以来，凸优化已成为推广线性代数和线性规划的有力工具：它具有可靠性（它总是收敛于全局最小值）和易处理性（它在合理的时间内完成）。

1.5.7 现阶段

目前，人们对从工程设计，统计学和机器学习到金融和结构力学等各个领域的优化技术应用非常感兴趣。与线性代数一样，最近与凸优化软件包，例如 CVX 或 YALMIP，可以非常容易地为中等大小的问题建立原型模型。

由于非常大的数据集的出现，目前正努力研究实现机器学习，图像处理等中出现的极大规模凸问题的解决方案。在这种情况下，90 年代对内点方法的初步关注已被早期算法（主要是 50 年代开发的所谓“一阶”算法）的重新审视和开发所取代，这些算法迭代非常容易。

1.6 本教材的使用建议

第 1 章绪论是你必须要了解的，它带领你快速概览从模式分析（包括图像感知和自然语言处理）、到数据分析与机器学习、到数学基础的整个内容逻辑链条，让你做到心中有数。

如果你具备工科的《高等数学》、《线性代数》、《概率论和数理统计》的基础知识，你可以用较少的时间来学习本教材的第 2 章（向量和矩阵基础）和第 7 章（概率论基础）的内容。但是我建议你最好要学，因为我们提供了一个从数据的视角来介绍这部分内容的尝试，里面也包含像数据的向量和矩阵表示（跟数据度量相关）、向量和矩阵函数（跟数据模型相关）、相关系数（跟特征选择有关）等传统线性代数和概率论不作为重点的内容。也介绍了很多基础概念，如投影和数据分析、机器学习任务的联系，然后迅速进入本课程其它对应章节更高级的内容学习。

如果你不具备工科线性代数、概率论和数理统计的基础知识，也不用害怕，你只需要更用心学习本教材第 2 章和第 7 章的内容即可。本教材第 2 章和第 7 章会为你提供一个足够本课程使用的简明的线性代数和概率论与数理统计基础知识，并配备足量的习题供你练习巩固。本教材每一章内容会配备大量的习题供你练习巩固所学内容。

本教材线性代数和矩阵计算、概率与信息论基础、优化基础三部分内容虽然通过数据分析与机器学习的处理流程有机统一在一起，但相对独立。因此你也可以重点选择其中某一板块内容进行学习，如果涉及教材前面介绍的知识点，但你又不了解这个知识点，你可以快速回溯这个知识点所在章节进行学习，比如优化求解计算涉及对某些特殊向量函数和矩阵函数进行微分，你可以回溯到第 6 章来进行补充学习。

本教材内容尽量做到详略得当，我们希望这本教材能够带领你领略数据科学、人工智能和机器学习涉及的不一样的数学世界。

习题

A 组

习题 1.1. 卷积神经网络是一类典型的处理图像的模型，其中卷积是其中一种非常重要的函数操作。试计算下列输入和卷积核做卷积的结果。

$$input = \begin{pmatrix} 1 & 3 & 0 & -1 \\ 3 & 0 & -1 & 2 \\ 1 & -1 & 2 & 0 \end{pmatrix}, Kernel = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix}$$

习题 1.2. 现有一组图片数据集，任务目标是将这些图片分类。其中图片中包含的类别有：猫、狗、鹦鹉、人。试试用 *one-hot* 向量将类别表示为向量。

习题 1.3. 现有文本集：

- *I know.*
- *You know.*
- *I know that you know.*
- *I know that you know that I know.*

试计算，该文本集各个单词的 *TF-IDF* 值。

习题 1.4. 设数据集为 x_1, x_2, \dots, x_n ，其中被分为两类 y_1, y_2 ，试写出线性分类器的评分函数的形式。并尝试使用 *0-1* 损失函数和平方损失函数来写出这个线性分类器的损失函数。

习题 1.5. 现有一个数据集有 5 个数据，分别被分类为

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

而一个模型给出的评分分别为

$$\begin{pmatrix} 2 \\ 8 \end{pmatrix}, \begin{pmatrix} 1 \\ 9 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 5 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

试给出此时模型给各个数据的概率评分以及交叉熵损失的值。

习题 1.6. 设数据集为 x_1, x_2, \dots, x_n ，其中被分为两类 y_1, y_2 。如果使用线性分类器，给出一个考虑结构风险的损失函数的公式。

B 组

习题 1.7. 利用 *python* 将一张黑白图片或彩色图片转化为矩阵或张量，并使图片水平翻转。

习题 1.8. 利用 *python* 统计 *IMDB* 影评数据集 *data.txt* 文件中，各单词出现的次数并计算每篇影评中各单词的 *tf*、*idf* 以及 *tf-idf*。

参考文献

- [1] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. 2011.
- [2] Laura Balzano and Stephen J. Wright. Local convergence of an algorithm for subspace identification from partial data. *Foundations of Computational Mathematics*, 15(5):1279–1314, 2015.
- [3] Samuel Burer and Renato D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.
- [4] E. Cands, X. Li, Y. Ma, and J. Wright. Robust principal component analysis?: Recovering low-rank matrices from sparse errors. In *Sensor Array & Multichannel Signal Processing Workshop*, 2010.
- [5] Emmanuel J. Cands and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717, 2009.
- [6] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Sparse and low-rank matrix decompositions. *Ifac Proceedings Volumes*, 42(10):1493–1498, 2009.
- [7] Yudong Chen and Martin J. Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *Computer Science*, 2015.
- [8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [9] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.
- [10] Y. Le Cun, B. Boser, J. S. Denker, R. E. Howard, W. Hubbard, L. D. Jackel, and D. Henderson. Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 2(2):396–404, 1990.
- [11] Y. Le Cun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to document recognition. *Proceedings of IEEE*, pages 2278–2324, 1998.
- [12] Li Deng, G. Hinton, and B. Kingsbury. New types of deep neural network learning for speech recognition and related applications: an overview. In *IEEE International Conference on Acoustics*, 2013.
- [13] Susan Dumais. Inductive learning algorithms and representations for text categorization. *Computer Engineering & Design*, pages 148–155, 2006.
- [14] N. I. Fisher and P. K. Sen. Probability inequalities for sums of bounded random variables. *Publications of the American Statistical Association*, 58(301):13–30, 1963.
- [15] Gilles Gasso, Aristidis Pappaioannou, Marina Spivak, and Lon Bottou. Batch and online learning algorithms for nonconvex neyman-pearson classification. *Acm Transactions on Intelligent Systems & Technology*, 2(3):1–19, 2011.

- [16] Friedman Jerome, Hastie Trevor, and Tibshirani Robert. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.
- [17] Deng Jia, Ding Nan, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Li Yuan, Hartmut Neven, and Hartwig Adam. Large-Scale Object Classification Using Label Relation Graphs. 2014.
- [18] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Proc Conference on Machine Learning, 1998.
- [19] Alan F Karr. Exploratory Data Mining and Data Cleaning. 2003.
- [20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In International Conference on Neural Information Processing Systems, 2012.
- [21] Yann Lecun, Lon Bottou, Genevieve B Orr, and Klaus Robert Müller. Efficient backprop. In *Neural Networks: Tricks of the Trade, This Book Is An Outgrowth of A Nips Workshop*, 1998.
- [22] David D. Lewis, Yiming Yang, Tony G. Rose, and Li Fan. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(2):361–397, 2004.
- [23] Hunacek Mark. The princeton companion to applied mathematics edited by higham nicholas j. , pp. 1016, ?69.95 (hard), isbn 978-0-691-15039-0, princeton university press (2015). *Mathematical Gazette*, 101(550):1016–170, 2017.
- [24] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. 2014.
- [25] Parrilo P A Recht B, Fazel M. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471C501, 2010.
- [26] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951.
- [27] D E Rumelhart, G E Hinton, and R JWilliams. Learning Internal Representations by Error Propagation. 1988.
- [28] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.
- [29] Berwin A Turlach, William N Venables, and Stephen J Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.
- [30] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.
- [31] Vladimir Vapnik. Estimation of dependences based on empirical data. *Journal of the Royal Statistical Society*, 41(3), 2006.

- [32] Vladimir N Vapnik. Statistical learning theory. Annals of the Institute of Statistical Mathematics, 55(2):371–389, 2003.
- [33] Chervonenkis A J. Vapnik V N. Theory of pattern recognition. Nauka, Moscow, 1974.
- [34] Blum Avrim, Hopcroft John, and Kannan Ravindran, Foundation of Data Science, Thursday 4th January, 2018.
- [35] Hastie Trevor, Tibshirani Robert and Friedman Jerome. 2016. The Elements of Statistical Learning. 2nd. Springer.
- [36] Goodfellow I, Bengio Y, Courville A. 深度学习 [M]. 人民邮电出版社, 2017.
- [37] 周志华. 机器学习 [M]. 清华大学出版社, 2016.
- [38] 欧高炎、朱占星、董彬、鄂维南。数据科学导引, 高等教育出版社, 北京, 2017.
- [39] 李航, 统计学习方法, 清华大学出版社, 北京, 2019.
- [40] 左孝凌, 离散数学的形成、发展及其在计算机科学中的作用与地位, 自然杂志, 1984, 7 (6) : 414-417.

传外勿请稿草

第二章 向量和矩阵基础

本章我们将按数据的向量和矩阵表示、数据的向量和矩阵空间、数据空间的关系以及数据空间上代数结构建立的过程来具体介绍数据科学与工程所涉及的向量和矩阵的计算所需的基本知识。向量是一个一元数组，可以看作为空间中的一个点，通过横向或纵向不同的排列方式，又可分为行向量或列向量。矩阵是一个二元数组，既可以看作一些一元数组构成的数表，也可以看作为输入空间和输出空间之间的线性变换，在数学上通常与线性方程组密切相关；在数据分析领域中是建立各种线性和非线性数据模型的基础。在向量和矩阵概念的基础上，我们可以定义向量的加、减和数乘等运算，也可以定义矩阵的加、减、乘积、数乘、逆和迹等运算，并引出有关迹、行列式、二次型、特征值和特征向量等矩阵的基本特征。由于在实际问题中，我们通常面对的是数据向量集合和数据矩阵集合构成的空间，也即在向量空间中来考虑问题。为此我们需要引入保持向量空间结构的运算——线性映射，以及在向量空间上引入新的代数结构，如内积和范数等来度量向量之间的距离和相似性等。此外，数据的高维空间表示常常给数据分析带来困难，这往往需要“拉回”到低维子空间才能看清其内蕴结构。我们引入正交和投影的概念，帮助我们理解这一运算的本质。

本章的内容概览图如下：

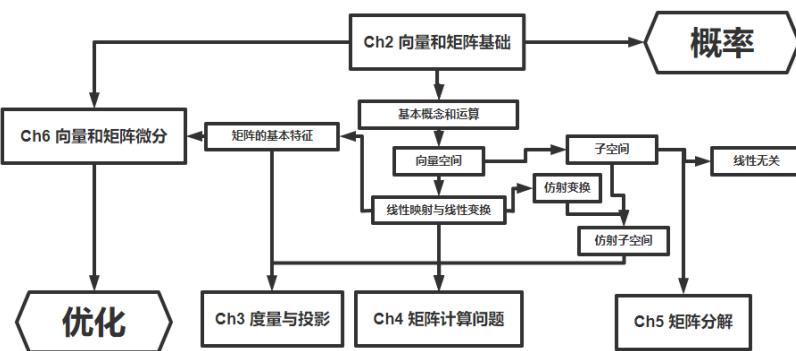


图 2.1: 本章导图

2.1 向量与矩阵的概念与运算

本节我们首先从数据的角度来谈谈向量和矩阵的基本概念，然后逐渐过渡到向量和矩阵的各种运算。

在中学解析几何中，我们已经看到，有些事物不能用一个数来刻画。例如，为了刻画一点在平面上的位置需要两个数，一点在空间的位置需要三个数，也就是需要它们的坐标。又比如，力学中的力、速度、加速度等，由于它们既有大小、又有方向，也不能用一个数刻画它们，在确定坐标系后它们可以用三个数来刻画。几何中向量的概念正是它们的抽象。但是还有不少东西用三个数来刻画是不够的。比如几何中，要刻画一个球需要刻画球的大小和位置，需要知道它中心的坐标（三个数）以及它的半径，也就是说，一个球需要 4 个数来刻画。至于一个刚体位置的确定就需要 6 个数了。

在数据科学、模式分析和人工智能领域，我们涉及到怎么刻画来自现实世界或网络世界各种实际的对象，比如一篇文章、一幅图像、一段语音和一条交易记录等，来作为计算机编码的输入依据和算法处理的对象。为了准确理解这些来自传感器中记录的数值或者网络活动过程中记录的数值数据，不混淆这些数值代表的实际意义，我们需要按照一定的顺序来记录或排列这些数值，让它形成一个有序的数组，如果这个数组中每个数值的次序只由一个单独的索引（比如竖着排列的顺序）就可以确定，则这个数组就称为一个一元有序数组。如果有 n 个数值，就称为一元 n 维有序数组。我们把这些例子中都涉及的数组抽象出来，就形成了向量的概念。特别，在数据科学、模式分析、人工智能和机器学习的语境下，我们可以粗略的称之为数据向量。

如果这个数组中每个数值的次序由两个索引（而非一个，比如横的顺序和竖的顺序）所确定，则这个数组就称为一个二元的有序数组，有时会称为数表。如果这个数表总共有 $m \times n$ 个数值，按照 m 行（横的）和 n 列（竖的）的形式排列则就形成为一个大小为 $m \times n$ 的数表。我们把这些 $m \times n$ 的数表都涉及的二元有序数组抽象出来，就形成了矩阵的概念。

2.1.1 向量与矩阵的基本概念：数据表示的观点

基于词袋模型的文本表示

在信息检索领域，比如我们想实现在文本中对某些关键词进行快速查找，那么文本表示是最基础最重要的第一步。这里仅以基于词项频率的词袋模型为例来介绍文本表示。词袋模型是指将所有词语装进一个袋子里，不考虑其词法和语序的问题，每个词语都是独立的。而词项频率指词项（索引的单位）在文本（词项序列）中出现的频率，简称词频。下面我们来看几段具体的文本。

例 2.1.1. 用向量表示文本

下面是纽约时报网络版在 2010 年 12 月 7 日的四则新闻提要：

(a) *Suit Over Targeted Killing in Terror Case Is Dismissed.* A federal judge on Tuesday dismissed a lawsuit that sought to block the United States from attempting to kill an American citizen, Anwar Al-Awlaki, who has been accused of aiding Al Qaeda.

(b) *In Tax Deal With G.O.P, a Portent for the Next 2 Years.* President Obama made clear that he was willing to alienate his liberal base in the interest of compromise. Tax Deal suggests new path for Obama. President Obama agreed to a tentative deal to extend the Bush tax cuts, part of a package to keep jobless aid and cut pay roll taxes.

(c) *Obama Urges China to Check North Koreans.* In a frank discussion, President Obama urged China's president to put the North Korean government on a tighter leash after a series of provocations.

(d) *Top Test Scores From Shanghai Stun Educators.* With China's debut in international standardized testing Shanghai students have surprised experts by outscoring counterparts in dozens of other countries.

我们希望用一元数组来表示这四则新闻标题，一元数组中的每一个元素对应一个特定项在文档中出现的次数。

首先将四则新闻标题中的单词进行简化，比如去除名词复数变为单数，例如将 (b) 中 *Years* 改为 *Year*；动词改为现在时，例如将 (a) 中 *Killing* 改为 *kill*。现在假设这个特定项为一本字典 V (*dict*)，字典 V 中的单词为 $\{aid, kill, deal, president, tax, china\}$ ，我们想知道每则新闻标题中的单词在字典中出现的频率，比如 *aid* 或 *kill* 在新闻 (a)、(b)、(c)、(d) 中出现的次数。

非常容易可以看出，在新闻 (a) 中 *aid* 共出现了 1 次，*kill* 共出现了 2 次，而字典 V 中的其它单词并没有出现，通过一元数组表示这一结果，即

$$\mathbf{a} = (1, 2, 0, 0, 0, 0)^T.$$

将一元数组 \mathbf{a} 归一化 (\mathbf{a} 中每个单词除以总共出现的次数)，便可以得到这则新闻标题在字典 V 中出现的相对频率，即

$$\mathbf{a}' = \left(\frac{1}{3}, \frac{2}{3}, 0, 0, 0, 0\right)^T.$$

将其它三则新闻也用一元数组表示，即分别为

$$\mathbf{b}' = \left(\frac{1}{10}, 0, \frac{3}{10}, \frac{1}{5}, \frac{2}{5}, 0\right)^T,$$

$$\mathbf{c}' = \left(0, 0, 0, \frac{1}{2}, 0, \frac{1}{2}\right)^T,$$

$$\mathbf{d}' = (0, 0, 0, 0, 0, 1)^T.$$

这样我们就把上述每一个新闻提要按照词频表示成一个一元六维的数组，这些数组是由一些具有意义的数值构成的，我们可以将它抽象出来，赋予新的定义，即向量。

向量的基本概念

当面对一个问题时，无论是在数学、计算机还是数据科学中，首先都需要搞清楚问题所在的定义域。例如在 2.1.1 中，如果问题是每则新闻标题在字典中出现的频率，那么表示向量（例

如向量 \mathbf{a}) 的元素都是非负整数, 也即其定义域是非负整数集; 而如果问题是每则新闻标题在字典中出现的相对频率, 那么表示向量 (例如向量 \mathbf{a}') 的元素是分数, 也即其定义域是分数集。一般地, 常见的定义域包括全体有理数构成的有理数集、全体实数构成的实数集和全体复数构成的复数集等。这些数集有着各自不同的性质, 但也有很多共同的代数性质。而有些数集也具有与有理数、实数、复数的全体所共有的代数性质, 为了在讨论中能够把它们统一起来, 由此引出一个更为一般的概念:

定义 2.1.1. 设 \mathbb{K} 是由一些数组成的集合, 如果 0 与 1 都在 \mathbb{K} 里且 \mathbb{K} 中任意两个数的和差积商 (除数不为零) 仍在 \mathbb{K} 里, 则称 \mathbb{K} 是一个数域。

常见的有理数集、实数集、复数集都可定义为数域, 它们分别称为有理数域 \mathbb{Q} , 实数域 \mathbb{R} , 复数域 \mathbb{C} 。通过引入数域的概念, 可以对向量进行形式化的定义:

定义 2.1.2. 由数域 \mathbb{K} 中 n 个数组成的有序数组 (a_1, a_2, \dots, a_n) 称为 \mathbb{K} 上的 n 维向量, 即 $\mathbf{a} = (a_1, a_2, \dots, a_n)$, 其中第 i 个数 a_i 称为 \mathbf{a} 的第 i 个分量。

几何上的向量可以认为是它的特殊情形, 即 $n = 2, 3$ 且 \mathbb{K} 为实数域的情形。在 $n > 3$ 时, n 维向量就没有直观的几何意义了。我们之所以仍然称它为向量, 一方面是由于它包括通常的向量作为特殊情形, 另一方面也由于它与通常的向量一样可以定义运算, 并且有许多运算性质是共同的, 因而采取这样一个几何的名词有好处。

以后我们用小写字母 $\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$ 代表向量。

定义 2.1.3. 如果 n 维向量

$$\mathbf{a} = (a_1, a_2, \dots, a_n), \mathbf{b} = (b_1, b_2, \dots, b_n)$$

的对应分量 $a_i = b_i (i = 1, 2, \dots, n)$, 则称向量 \mathbf{a} 与 \mathbf{b} 相等, 记作 $\mathbf{a} = \mathbf{b}$ 。

定义 2.1.4. 分量全为零的 n 维向量 $(0, 0, \dots, 0)$ 称为零向量, 记作 $\mathbf{0}$, 向量 $-\mathbf{a} = (-a_1, -a_2, \dots, -a_n)$ 称为向量 $\mathbf{a} = (a_1, a_2, \dots, a_n)$ 的负向量, 记作 $-\mathbf{a}$ 。

由若干个维数相同的向量组成的集合, 称为向量组。例如在例 2.1.1 中, 新闻标题集合 $\{\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{d}'\}$ 是由 4 个 6 维向量组成的向量组。

在科学和工程中遇到的向量可以分为以下三种:

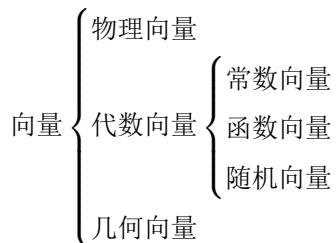
(1) 物理向量: 泛指既有大小, 又有方向的物理量, 如速度、加速度、位移等。
(2) 几何向量: 为了将物理向量可视化, 常用带方向的 (简称“有向”) 线段表示。这种有向线段称为几何向量。例如, $\mathbf{v} = \overrightarrow{AB}$ 表示的有向线段, 其起点为 A , 终点为 B 。

(3) 代数向量: 几何向量可以用代数形式表示。例如, 若平面上的几何向量 $\mathbf{v} = \overrightarrow{AB}$ 的起点坐标 $A = (a_1, a_2)$, 终点坐标 $B = (b_1, b_2)$, 则该几何向量可以表示为代数形式 $\mathbf{v} = \begin{bmatrix} b_1 & -a_1 \\ b_2 & -a_2 \end{bmatrix}$ 。
这种用代数形式表示的几何向量称为代数向量。

根据元素取值种类的不同，代数向量又可分为以下三种：

- (1) 常数向量：向量的元素全部为实常数或者复常数，如 $\mathbf{a} = [1, 5, 4]^T$ 等。
- (2) 函数向量：向量的元素包含了函数值，如 $\mathbf{x} = [1, x^2, \dots, x^n]^T$ 等。
- (3) 随机向量：向量的元素为随机变量或随机过程，如 $\mathbf{x}(n) = [x_1(n), \dots, x_m(n)]^T$ ，其中 $x_1(n), \dots, x_m(n)$ 是 m 个随机过程或随机信号。

下图归纳了向量的分类。



实际应用中遇到的往往是物理向量，而几何向量是物理向量的可视化，代数向量则可看作是物理向量的运算化工具。

用矩阵表示词项-文档集合和图像

除了向量，矩阵在数学中，特别是线性代数中也起到了举足轻重的地位，线性方程组、线性映射、线性变换都与矩阵密不可分。而在数据科学中，矩阵也是最为常见的数据表现形式之一，自然语言处理和图像处理都离不开矩阵的表示。例如，我们通常可以用矩阵来表示文本向量集。

例 2.1.2. 用矩阵表示文本向量集

在例 2.1.1 中，每则新闻标题都由一个 6 维向量表示，那么这四则新闻标题组成的新闻集可以由 4 个这样的 6 维向量组成的向量集表示。换言之，这个新闻集可以按列组成一个 6×4 的二元数组。即

$$\mathbf{A} = \begin{pmatrix} \frac{1}{3} & \frac{1}{10} & 0 & 0 \\ \frac{2}{3} & 0 & 0 & 0 \\ 0 & \frac{3}{10} & 0 & 0 \\ 0 & \frac{1}{5} & \frac{1}{2} & 0 \\ 0 & \frac{2}{5} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 1 \end{pmatrix}$$

在图像中，二元数组则更为常见，因为在计算机中读取图像的过程中，其本身就已经转化为二元数组的形式。

例 2.1.3. 计算机中存储的图像

在计算机中，如果只保留图像的灰度，那么该图像可以表示为二元数组，其中二元数组中的每个输入包含图像中相应像素的强度值（在 $[0, I]$ 中为“double”类型值，其中 0 表示黑色， I

表示白色；或“int”类型值，介于 0 至 255 之间)。图 2.2 显示了一张灰度图，具有 500 个水平像素和 600 个垂直像素。

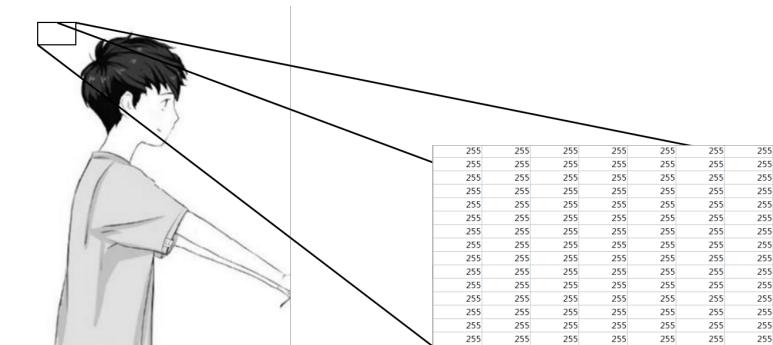


图 2.2: 图像的表示

矩阵的基本概念

与向量一样，矩阵也可以给出形式化的定义：

定义 2.1.5. 由数域 \mathbb{K} 中的 $m \times n$ 个数 a_{ij} ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$) 排成 m 行、 n 列的表

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

称为 \mathbb{K} 上的 $m \times n$ 矩阵，记为 $\mathbf{A} = (a_{ij})_{m \times n}$ 或 $\mathbf{A}_{m \times n}$ ，表中的每一个数都称为矩阵 \mathbf{A} 的一个元素。若 $m = n$ ，则 $n \times n$ 矩阵 $\mathbf{A} = (a_{ij})_{m \times n}$ 也称为 n 阶方阵。

从定义上可以看出，矩阵是一个二维数组，而向量是矩阵的一种特殊形式，因为它的 $m = 1$ 或者 $n = 1$ ，其中 $1 \times n$ 的矩阵称之为行向量，而 $m \times 1$ 的矩阵称之为列向量。

当 \mathbf{A} 的每个元素是实数，也即 $a_{ij} \in \mathbb{R}$ 时，则称 \mathbf{A} 为实矩阵。所有 n 维实矩阵构成的集合记为 $\mathbb{R}^{m \times n}$ 。

当 \mathbf{A} 的每个元素是复数，也即 $a_{ij} \in \mathbb{C}$ 时，则称 \mathbf{A} 为复矩阵。所有 n 维复矩阵构成的集合记为 $\mathbb{C}^{m \times n}$ 。

在科学和工程中遇到的矩阵可以分为以下三种：

(1) 常数矩阵：矩阵的元素全部为实常数或者复常数。

(2) 函数矩阵：矩阵的元素包含了函数值。

(3) 随机矩阵：矩阵的元素为表示概率的非负实数。

例如, Laplace 矩阵(图与网络中常用的矩阵)是常数矩阵, 而 Markov 矩阵(状态转移矩阵)为随机矩阵。

在引入了向量和矩阵定义之后, 接下来我们给出向量和矩阵的基本运算。

2.1.2 向量的运算

加法和数乘是向量之间的两种基本代数运算, 统称为向量的线性运算。

定义 2.1.6. 设向量 $\mathbf{a} = (a_1, a_2, \dots, a_n), \mathbf{b} = (b_1, b_2, \dots, b_n)$, 则向量 $(a_1 + b_1, a_2 + b_2, \dots, a_n + b_n)$ 称为向量 \mathbf{a} 与 \mathbf{b} 的和, 记作 $\mathbf{a} + \mathbf{b}$, 即

$$\mathbf{a} + \mathbf{b} = (a_1 + b_1, a_2 + b_2, \dots, a_n + b_n).$$

利用向量的加法及负向量, 类似可定义向量的减法。

定义 2.1.7. 设向量 $\mathbf{a} = (a_1, a_2, \dots, a_n)$, k 是数域 \mathbb{K} 中的数, 则向量 $(ka_1, ka_2, \dots, ka_n)$ 称为数 k 与向量 \mathbf{a} 的乘积, 简称数乘, 记作 $k\mathbf{a}$, 即

$$k\mathbf{a} = (ka_1, ka_2, \dots, ka_n).$$

定理 2.1.1. 设 $\mathbf{a}, \mathbf{b}, \mathbf{c}$ 是 \mathbb{K} 上的 n 维向量, λ, μ 是数域 \mathbb{K} 中的数, 则向量的加法和数乘运算满足下列交换律、结合律和分配律

- (1) $\mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a}$,
- (2) $(\mathbf{a} + \mathbf{b}) + \mathbf{c} = \mathbf{a} + (\mathbf{b} + \mathbf{c})$,
- (3) $\mathbf{a} + \mathbf{0} = \mathbf{a}$,
- (4) $\mathbf{a} + (-\mathbf{a}) = \mathbf{0}$,
- (5) $1\mathbf{a} = \mathbf{a}$,
- (6) $\lambda(\mu\mathbf{a}) = (\lambda\mu)\mathbf{a}$,
- (7) $(\lambda + \mu)\mathbf{a} = \lambda\mathbf{a} + \mu\mathbf{a}$,
- (8) $\lambda(\mathbf{a} + \mathbf{b}) = \lambda\mathbf{a} + \lambda\mathbf{b}$.

应当注意的是, 两个向量只有维数相同时, 才能进行加法和减法运算。

2.1.3 矩阵的运算

接下来, 我们要介绍矩阵的基本运算, 在加法和数量乘法之外, 还包括乘积、分块、逆和转置等运算。

矩阵的加法

定义 2.1.8. 设 $A = (a_{ij})_{m \times n}$, $B = (b_{ij})_{m \times n}$ 。令 $C = (c_{ij})_{m \times n}$, 其中 $c_{ij} = a_{ij} + b_{ij}$ ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$), 则称 C 为 A 与 B 的和, 记作 $C = A + B$ 。

A 的负矩阵记为 $-A = (-a_{ij})_{m \times n}$ 。

从而将 A 与 B 的差记为 $A - B = A + (-B) = (a_{ij} - b_{ij})_{m \times n}$ 。

定理 2.1.2. 设元素全为零的矩阵称为零矩阵, 记作 O , 则矩阵的加法满足下列规律:

- (1) 交换律 $A + B = B + A$,
- (2) 结合律 $(A + B) + C = A + (B + C)$,
- (3) $A + O = A$,
- (4) $A + (-A) = O$.

矩阵的乘积

定义 2.1.9. $A = (a_{ij})_{m \times r}$, $B = (b_{ij})_{r \times n}$, 令 $C = (c_{ij})_{m \times n}$, 其中 $c_{ij} = \sum_{k=1}^r a_{ik}b_{kj}$ ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$), 则 C 称为 A 与 B 的乘积, 记作 $C = AB$ 。

需要特别注意的是, 矩阵 A 与 B 的乘积矩阵 C 的第 i 行第 j 列的元素 c_{ij} 等于 A 的第 i 行与 B 的第 j 列的对应元素乘积的和。只有 A 的列数与 B 的行数相时, 乘积 AB 才有意义。一般地, $AB \neq BA$ 。

定理 2.1.3. 矩阵的乘积满足下列规律:

- (1) 结合律 $(AB)C = A(BC)$,
- (2) 左分配律 $A(B + C) = AB + AC$, 右分配律 $(B + C)A = BA + CA$.

定义 2.1.10. 主对角线上的元素全是 1, 其余元素全是 0 的 n 阶方阵

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

称为 n 阶单位矩阵, 记为 I_n 或简记为 I 。 $AI = IA = A$ 。

有了矩阵的乘积, 便可以定义矩阵的幂。

定义 2.1.11. 设 $A = (a_{ij})_{n \times n}$, 则 A 的 k 次幂定义为 k 个 A 连乘, 记作 A^k , 即 $A^k = AA \cdots A$ (k 个因子)。

定义 2.1.12. 设 $A = (a_{ij})_{m \times n}$ 是 $m \times n$ 矩阵, $x = (x_1, x_2, \dots, x_n)^\top$ 是 n 维列向量, 令 $b = (b_1, b_2, \dots, b_m)$, 其中 $b_i = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = \sum_{k=1}^n a_{ik}x_k (i = 1, \dots, m)$, 则 b 称为矩阵 A 与向量 x 的乘积, 记作 $b = Ax$.

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

矩阵的数量乘积

定义 2.1.13. $A = (a_{ij})_{m \times n}, \lambda \in \mathbb{K}$, 则 λ 与 A 的数量乘积或标量乘积定义为 $\lambda A = (\lambda a_{ij})_{m \times n}$.

定理 2.1.4. 矩阵的数量乘积满足下列规律:

- (1) $1A = A$,
- (2) $(\lambda\mu)A = \lambda(\mu A)$,
- (3) $(\lambda + \mu)A = \lambda A + \mu A$,
- (4) $\lambda(A + B) = \lambda A + \lambda B$,
- (5) $\lambda(AB) = (\lambda A)B = A(\lambda B)$.

矩阵的分块

当要处理一些维数比较高的矩阵时, 我们可以把一个大矩阵看成是一些小矩阵组成的。就如同矩阵是由数组成的一样, 在运算中, 把这些矩阵当作数一样来计算, 这就是矩阵的分块。

例 2.1.4. 设矩阵 A

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} I_2 & \mathbf{0} \\ A_1 & I_2 \end{pmatrix}$$

其中, I_2 表示 2×2 的单位矩阵, $\mathbf{0}$ 表示零矩阵, 而

$$A_1 = \begin{pmatrix} -1 & 2 \\ 1 & 1 \end{pmatrix},$$

有另一矩阵 B

$$B = \begin{pmatrix} 1 & 0 & 3 & 2 \\ -1 & 2 & 0 & 1 \\ 1 & 0 & 4 & 1 \\ -1 & -1 & 2 & 0 \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

其中,

$$\mathbf{B}_{11} = \begin{pmatrix} 1 & 0 \\ -1 & 2 \end{pmatrix}, \mathbf{B}_{12} = \begin{pmatrix} 3 & 2 \\ 0 & 1 \end{pmatrix}$$

$$\mathbf{B}_{21} = \begin{pmatrix} 1 & 0 \\ -1 & -1 \end{pmatrix}, \mathbf{B}_{22} = \begin{pmatrix} 4 & 1 \\ 2 & 0 \end{pmatrix}$$

在计算 \mathbf{AB} 时, 把 A, B 都看成是由这些小矩阵组成的, 即按 2 维矩阵来计算, 于是有

$$\begin{aligned} \mathbf{AB} &= \begin{pmatrix} \mathbf{I}_2 & \mathbf{O} \\ \mathbf{A}_1 & \mathbf{I}_2 \end{pmatrix} \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{A}_1 \mathbf{B}_{11} + \mathbf{B}_{21} & \mathbf{A}_1 \mathbf{B}_{12} + \mathbf{B}_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 3 & 2 \\ -1 & 2 & 0 & 1 \\ -2 & 4 & 1 & 1 \\ -1 & 1 & 5 & 3 \end{pmatrix} \end{aligned}$$

初等矩阵

定义 2.1.14. 所谓数域 \mathbb{K} 上矩阵的初等行变换是指下列三种变换:

- (1) 以 \mathbb{K} 中一个非零的数乘矩阵的某一行;
- (2) 把矩阵的某一行的 c 倍加到另一行, 这里 c 是 \mathbb{K} 中任意一个数;
- (3) 互换矩阵中两行的位置。

定义 2.1.15. 由单位矩阵 I 经过一次初等行变换得到的矩阵称为初等矩阵。

同样, 如果对矩阵做初等列变换也能得到相应的初等矩阵。由矩阵的初等矩阵和矩阵乘积的联系, 我们不加证明便可以得到如下定理:

定理 2.1.5. 对一个 $m \times n$ 的矩阵 A 作一初等行变换就相当于在 A 左边乘上相应的 $m \times m$ 的初等矩阵; 对 A 作一初等列变换就相当于在 A 右边乘上相应的 $n \times n$ 的初等矩阵。

例 2.1.5. 设矩阵 A

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

对矩阵 A 作一初等行变换 (互换第 1 和第 3 行的位置), 则有

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

矩阵的逆

定义 2.1.16. 设 A 是数域 \mathbb{K} 上的 $n \times n$ 矩阵, 如果存在 \mathbb{K} 上的 $n \times n$ 矩阵 B , 使得 $AB = BA = I$, 则称 A 为可逆矩阵, 简称 A 可逆, 而 B 则称为 A 的逆矩阵, 记作 A^{-1} , 即 $AA^{-1} = A^{-1}A = I$.

例 2.1.6. 2×2 矩阵的逆

考虑一个 2×2 矩阵:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

如果有另一个 2×2 矩阵:

$$B = \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}$$

则有

$$AB = \begin{pmatrix} a_{11}a_{22} - a_{12}a_{21} & 0 \\ 0 & a_{11}a_{22} - a_{12}a_{21} \end{pmatrix} = (a_{11}a_{22} - a_{12}a_{21})I$$

当 $a_{11}a_{22} - a_{12}a_{21} \neq 0$ 时, 有 $A^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}}B$ 其中, $a_{11}a_{22} - a_{12}a_{21} \neq 0$ 其实是 2×2 矩阵 B 的行列式, 有关行列式的概念将在以后的章节中详细介绍。

值得注意的是, 如果 A 可逆, 则 A^{-1} 也可逆, 且 $(A^{-1})^{-1} = A$ 。进而可知, 若 A 可逆, 则其逆矩阵是唯一的。

定理 2.1.6. 矩阵的逆满足如下性质:

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

$$(2) (A^{-1})^T = (A^T)^{-1}, 特别要注意的一般地, (A + B)^{-1} \neq A^{-1} + B^{-1}$$

上面我们已经定义了初等矩阵和初等变换, 并且知道用初等行变换可以化简矩阵。如果同时用行与列的初等变换, 那么矩阵还可以进一步化简。为了方便, 我们引入:

定义 2.1.17. 矩阵 A 和 B 是等价的, 如果 B 可以由 A 经过一系列初等变换得到。

等价是矩阵间的一种关系。不难证明, 它具有反身性、对称性与传递性。根据定理 2.1.5, 对一矩阵作初等变换就相当于用相应的初等矩阵去乘以这个矩阵。因此, 矩阵 A, B 等价的充分必要条件是有初等矩阵 $P_1, \dots, P_l, Q_1, \dots, Q_t$, 使得

$$A = P_1 \cdots P_l B Q_1 \cdots Q_t,$$

由此可得:

定理 2.1.7. n 阶矩阵 A 为可逆的充分必要条件是它能表成一些初等矩阵的乘积:

$$A = P_1 P_2 \cdots P_m. \quad (2.1)$$

将 (2.1) 改写一下, 有

$$\mathbf{P}_m^{-1} \cdots \mathbf{P}_2^{-1} \mathbf{P}_1^{-1} \mathbf{A} = \mathbf{I}.$$

则有

定理 2.1.8. 可逆矩阵总是可以经过一系列初等行变换化为单位矩阵。

由此可以得到求逆矩阵的初等变换法。设 \mathbf{A} 是 $n \times n$ 可逆矩阵, 在 \mathbf{A} 的右边写上 $n \times n$ 单位矩阵 \mathbf{I} , 构成一个 $n \times 2n$ 矩阵 $(\mathbf{A} \ \mathbf{I})$, 再对 $(\mathbf{A} \ \mathbf{I})$ 进行一系列初等行变换, 把它的左半部分 \mathbf{A} 化为单位矩阵 \mathbf{I} , 则它的右半部分 \mathbf{I} 就化为 \mathbf{A} 的逆矩阵 \mathbf{A}^{-1} , 即

$$(\mathbf{A} \ \mathbf{I}) \xrightarrow{\text{初等行变换}} (\mathbf{I} \ \mathbf{A}^{-1}).$$

例 2.1.7. 考虑一个 2×2 矩阵:

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

则

$$(\mathbf{A} \mathbf{I}) = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 3 & 4 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 1 & 0 \\ 0 & -2 & -3 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & -2 & 1 \\ 0 & -2 & -3 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & -2 & 1 \\ 0 & 1 & \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

即

$$\mathbf{A}^{-1} = \begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

矩阵的转置

定义 2.1.18. 设 $\mathbf{A} = (a_{ij})_{m \times n}$, 把 \mathbf{A} 的行、列互换所得到的矩阵称为 \mathbf{A} 的转置, 记作 \mathbf{A}^T , 或 \mathbf{A}' , 即 $\mathbf{A}^T = \mathbf{A}' = (a_{ji})_{n \times m}$ 。

定理 2.1.9. 矩阵的转置满足下列规律:

$$(1) (\mathbf{A}^T)^T = \mathbf{A},$$

$$(2) (\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T,$$

$$(3) (\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T,$$

$$(4) (\lambda \mathbf{A})^T = \lambda \mathbf{A}^T.$$

2.1.4 线性方程组

线性方程组是线性代数研究的中心对象, 许多问题最后都可以归结为线性方程组的求解, 包括数据分析领域很多优化问题的求解最后都可以转为线性方程组的求解。前面我们已经给出了矩阵和向量的乘积, 下面我们通过一个例子说明, 线性方程组可以利用这一运算表示成紧凑的形式。

例 2.1.8. 某公司生产产品 N_1, \dots, N_n , 其需要的资源分别为 R_1, \dots, R_m 。为了生产一单位的 N_j 需要 a_{ij} 单位的 R_i , 其中 $i = 1, \dots, m; j = 1, \dots, n$ 。目的是找到最佳的生产计划, 也即如果有 b_i 单位的 R_i 可供使用, 那么应该生产多少 (设为 x_j) 单位的产品 N_j 使得恰好用尽资源。如果我们生产 x_1, \dots, x_n 单位的对应产品, 我们一共需要 $a_{i1}x_1 + \dots + a_{in}x_n$ 单位资源 R_i 。最优生产计划 $(x_1, \dots, x_n) \in \mathbb{R}^n$, 因此它必须满足方程组

$$a_{11}x_1 + \dots + a_{1n}x_n = b_1$$

⋮

$$a_{m1}x_1 + \dots + a_{mn}x_n = b_m$$

上述方程组可以写成矩阵的形式

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

用一个紧凑形式表示即为

$$Ax = b \quad (2.2)$$

我们称(2.2)为线性方程组的一般形式, 并且 $\mathbf{x}_1, \dots, \mathbf{x}_n$ 是该线性方程组的未知数。满足(2.2)的每一个 n 元 $(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^n$ 是线性方程组的一个解。下面利用初等变换和逆法来求一个简单的线性方程组, 关于一般线性方程组(2.2)的解集和求解方法我们会在第四章中给出详细的介绍。

例 2.1.9. 求

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 7 \end{pmatrix}$$

解法一: 我们记增广矩阵为

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 7 \end{pmatrix}$$

我们使用行初等变换法求解这个线性方程组。即

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 7 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 3 \\ 1 & -2 & -2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

我们将左边化为单位阵, 最右边的一列即为解。所以解

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

解法二：上面，我们介绍了矩阵的逆。如果系数矩阵可逆，则可以在方程两边同时左乘系数矩阵 A 的逆 A^{-1} 即

$$A^{-1}Ax = A^{-1}b \rightarrow Ix = A^{-1}b \rightarrow x = A^{-1}b$$

对于上面那个线性方程组，他的系数矩阵的逆我们已经求过了。

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}^{-1} = \begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

那么

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} 3 \\ 7 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

注记 5. 前面我们讨论过，当一个数组中每个数值只有一个索引时，可以表示为向量；当一个数组每个数值都具有两个索引时，可以表示为矩阵。然而，在某些情况下，我们也会讨论每个数值的索引超过两个的情况，这时就形成了一个有序的三元或更高元的数组，一般地，一个数组中的元素分布在若干元（二元以上）索引的规则网格中，我们称之为张量。例如，图2.3彩色图像的表示，是一个 $m \times n \times 3$ 矩阵，可以看成一个张量。

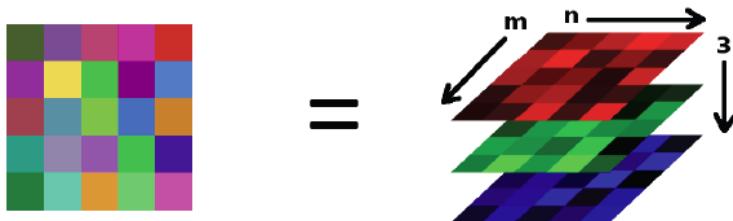


图 2.3: 彩色图像的张量表示

我们把现实世界各种实际的对象用向量和矩阵来进行表示，称为数据的低维结构表示；用张量来表示，称为数据的高维结构表示。此外，现实世界还有一些更抽象的非数组型结构化对象，如网络结构，不能用一个向量或张量来表示，而需要用图，甚至更抽象的代数结构，如集合、半群和群等结构来表示。在本书中，我们只涉及数据的低维结构表示。

注记 6. 有了数据的向量表示以后，我们可以处理这些数据向量获得它潜在的更好的表示。我们主要讨论两种寻找更佳表示的方式：(1) 寻找原始特征向量的低维近似。从数据处理的角度，这可以通过主成分分析来获得，从数学的角度来看，这涉及奇异值分解，我们在第 4 章将会介绍这部分内容。(2) 寻找原始特征向量的高维表示。从数据的角度看，寻找高维表示的目的在于我们可以利用它构造新的特征作为原始特征的非线性组合，这反过来会使得数据分析和机器学习的问题更容易处理。从数学的角度看，特征映射和本章接下来要讲的向量空间之间的线性映射和 3.4 节的非线性函数密切相关。

2.2 向量空间

在中学时，我们就知道现实世界的很多对象按照某些特定的属性可以形成集合，在数据科学中，比如一些文本可以构成集合，一些图像可以构成集合，因为文本可以用向量表示、图像可以用矩阵表示。那么，向量和矩阵也可以构成集合。我们接下来介绍数域 \mathbb{K} 上的所有向量或矩阵等抽象对象构成的集合的性质。

由上一节向量和矩阵的基本运算，我们发现向量和矩阵虽然是两个不同的数学对象，但是它们都有加法和数量乘法这两种运算。当然随着对象的不同，这两种运算的定义也是不同的，为了抓住它们的共同点，把它们统一起来加以研究，我们引入向量空间的概念，也称为线性空间。注意这里的向量空间显然是广义的向量空间，它首先是一个集合，集合里面的元素可以是我们上一讲提到向量，也可以是矩阵，也可以是其它对象，这里的向量比几何中所谓的向量的涵义要丰富的多。

向量空间具有重要的应用。从数学的角度看，它可以作为方程组的解空间。从数据科学、人工智能和机器学习的角度看，它也是这些学科领域数据问题处理空间的出发点，数据科学、人工智能和机器学习中很多基本的处理任务都可以放在向量空间及其子空间中来考虑。比如，在数据科学中，我们常常得到海量的高维数据，但是这些数据中，经常是只有几个维度的数据和我们的预测或决策问题有关。也就是说，我们只需要考虑高维空间中的一个低维子空间就可以解决我们的问题。还有一些情形，某些维度的数据和另外一些维度的数据没有关联，因此我们可以分别处理几个小的子空间来帮助我们解决最终的问题。这可以通过对数据做降维（有时也称特征选择）或做特征抽取来实现。

2.2.1 向量空间的基本概念：数据处理空间的出发点

我们首先来看一个鸢尾花数据集降维的具体例子。Iris（鸢尾花）数据集是机器学习中常用的分类实验数据集，是一类多重变量分析的数据集。该数据集包含 150 个数据，分为 3 类，每类 50 个数据，每个数据包含 4 个属性。可通过花萼长度，花萼宽度，花瓣长度，花瓣宽度 (sepal length, sepal width, petal length, petal width) 4 个特征预测鸢尾花卉属于 Setosa, Versicolour, Virginica 三个种类中的哪一类。如果我们想可视化或者在低维空间上找到数据分类的特征依据，通常我们把这个数据集看成一个 4 维的向量空间，然后选取 2 维或 3 维子空间对数据进行降维。图 2.4 是对降维结果的一个直观展示。

通过这个例子我们可以看出，向量空间和子空间的概念是对数据进行了表示以后，数据处理的最基本的出发点。下面我们给出向量空间和子空间的形式化定义。所谓“空间”，就是指满足一定结构或具有一定性质的集合。向量空间是线性代数研究的一种基本结构。在向量空间中，一定结构就是定义了加法运算和数乘运算，一定性质指这两种运算满足封闭性。

定义 2.2.1. 设 \mathbb{V} 是由 n 维向量组成的非空集合， \mathbb{K} 是一个数域。在 \mathbb{V} 上定义了的加法，在

sepal_length	sepal_width	petal_length	petal_width	species
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
7	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.9	3.1	4.9	1.5	Iris-versicolor
5.5	2.3	4	1.3	Iris-versicolor
6.5	2.8	4.6	1.5	Iris-versicolor
5.7	2.8	4.5	1.3	Iris-versicolor
6.3	3.3	6	2.5	Iris-virginica
5.8	2.7	5.1	1.9	Iris-virginica
7.1	3	5.9	2.1	Iris-virginica
6.3	2.9	5.6	1.8	Iris-virginica
6.5	3	5.8	2.2	Iris-virginica
7.6	3	6.6	2.1	Iris-virginica
4.9	2.5	4.5	1.7	Iris-virginica
7.3	2.9	6.3	1.8	Iris-virginica

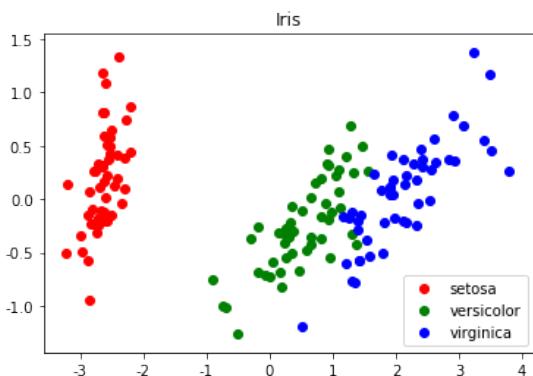


图 2.4: Iris 数据集(左), PCA 降维可视化(右)

\mathbb{K} 与集合 \mathbb{V} 上定义了数乘，并且 $\forall \mathbf{a}, \mathbf{b} \in \mathbb{V}$ 及任意数 $k \in \mathbb{K}$, 有 $\mathbf{a} + \mathbf{b}, k\mathbf{a} \in \mathbb{V}$ ，则称 \mathbb{V} 对于向量的加法和数乘两种运算封闭， \mathbb{V} 为数域 \mathbb{K} 上的 n 维向量空间或者线性空间。

下面介绍几个向量空间的例子。

例 2.2.1. 数域 \mathbb{K} 上的 n 维向量，按照如下定义的加法和数乘运算，构成数域 \mathbb{K} 上的向量空间。

考虑向量空间 $\mathbb{V} = \mathbb{K}^n$ ，任意两个向量 $\mathbf{a}, \mathbf{b} \in \mathbb{V}$, $\lambda \in \mathbb{K}$ 满足：

1. 加法

$$\mathbf{a} + \mathbf{b} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} + \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} a_1 + b_1 \\ \vdots \\ a_n + b_n \end{pmatrix} \in \mathbb{V}$$

2. 数乘

$$\lambda \mathbf{a} = \begin{pmatrix} \lambda a_1 \\ \vdots \\ \lambda a_n \end{pmatrix} \in \mathbb{V}$$

如果数域 \mathbb{K} 为实数域 \mathbb{R} 或复数域 \mathbb{C} ，此时所有 n 元实数组或复数组构成的向量空间 \mathbb{R}^n 和 \mathbb{C}^n ，分别称为 n 维实向量空间 \mathbb{R}^n 或 n 维复向量空间 \mathbb{C}^n 。注意到，向量空间一定包含零元素。

例 2.2.2. 数域 \mathbb{K} 上的 $m \times n$ 矩阵，按照如下定义的加法和数乘运算，构成数域 \mathbb{K} 上的向量空间。

考虑矩阵空间 $\mathbb{V} = \mathbb{K}^{m \times n}$ ，任意的两个矩阵 $\mathbf{A}, \mathbf{B} \in \mathbb{V}$, $\lambda \in \mathbb{K}$ 满足：

1. 加法

$$\mathbf{A} + \mathbf{B} = \begin{pmatrix} a_{11} + b_{11} & \dots & a_{1n} + b_{1n} \\ \vdots & & \vdots \\ a_{m1} + b_{m1} & \dots & a_{mn} + b_{mn} \end{pmatrix} \in \mathbb{V}$$

2. 数乘

$$\lambda \mathbf{A} = \begin{pmatrix} \lambda a_{11} & \dots & \lambda a_{1n} \\ \vdots & & \vdots \\ \lambda a_{m1} & \dots & \lambda a_{mn} \end{pmatrix} \in \mathbb{V}$$

下面这个例子强调区分定义中 λ 属于的数域与向量所在的集合 \mathbb{V} 。

例 2.2.3. 复数域 \mathbb{C} :

- 令 λ 所在的数域 $\mathbb{K} = \mathbb{C}$, 定义加法为复数加法、数乘为复数乘法, 根据复数的加法和乘法, 我们可以知道 **复数域 \mathbb{C} 是自身上的向量空间**。
- 令 λ 所在的数域 $\mathbb{K} = \mathbb{R}$, 定义加法为实部与实部相加, 虚部与虚部相加, 而数乘则是将实数分别乘至实部和虚部 (不需要引入复数的乘法)。容易知道, **复数域 \mathbb{C} 是实数域 \mathbb{R} 上的向量空间**。

我们通过下面这个例子更广义的理解向量空间中向量的含义。

例 2.2.4. 数域 \mathbb{R} 上的次数小于 n 的一元多项式, 即

$$\mathbb{P}_n = \{p : p(x) = a_{n-1}x^{n-1} + \dots + a_1x + a_0, \text{ 其中 } a_0, a_1, \dots, a_{n-1} \in \mathbb{R}\}$$

构成 \mathbb{R} 上的向量空间。这是因为对于 $\forall p_1, p_2 \in \mathbb{P}_n$ 及任意数 $k \in \mathbb{K}$, 有 $p_1 + p_2, kp_1 \in \mathbb{P}_n$ 。

2.2.2 向量子空间

如果一个“大”集合是一个线性空间或者向量空间, 如果我们能找到一个它所包含的“小”集合仍然是一个向量空间, 我们说这个“小”空间是“大”空间的子空间。用严格的数学语言描述就是,

定义 2.2.2. 设 \mathbb{X} 是 \mathbb{K} 上的 n 维线性空间, \mathbb{Y} 是 \mathbb{X} 的子集且满足: 若 $\mathbf{x}, \mathbf{y} \in \mathbb{Y}$, 则 $\mathbf{x} + \mathbf{y} \in \mathbb{Y}$; 若 $a \in \mathbb{K}, \mathbf{x} \in \mathbb{Y}$, 则 $a\mathbf{x} \in \mathbb{Y}$, 则称 \mathbb{Y} 是 \mathbb{X} 的线性子空间, 简称子空间。

子空间也一定包含零元素。

例 2.2.5. 非空的线性空间一定包含以下两个子空间: 自身和 $\{\mathbf{0}\}$ 。我们把只含零向量的子集称为零子空间。

零子空间和线性空间本身统称为平凡子空间, 其它子空间叫做非平凡子空间。

通过图 2.5 中几个例子体会子空间与子集的区别。

例 2.2.6. 图 2.5 中只有 D 是 \mathbb{R}^2 的子空间。在 A 和 C 中不能保证封闭性。 B 则不包括 $\mathbf{0}$ 。

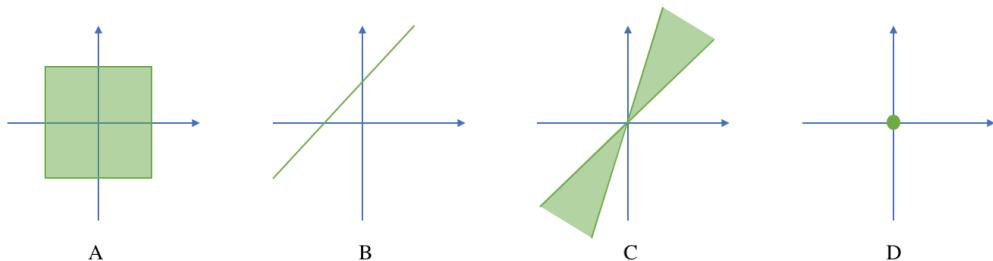


图 2.5: \mathbb{R}^2 中的一些子集

例 2.2.7. $\mathbb{V} = \{x | a^T x = 0, x \in \mathbb{R}^n\}$ 是 n 维空间的子空间。

若 $x_1 \in \mathbb{V}, x_2 \in \mathbb{V}$, 有 $a^T x_1 = 0, a^T x_2 = 0$, 则 $a^T(x_1 + x_2) = 0$ 。有 $x_1 + x_2 \in \mathbb{V}$ 。

对任意 $c \in \mathbb{R}, x \in \mathbb{V}$, 有 $a^T(cx) = ca^T x = 0$, 有 $cx \in \mathbb{V}$ 。

例 2.2.8. 设 $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{k \times n}, b \in \mathbb{R}^m, b \neq 0$,

- 方程组 $Ax = \mathbf{0}$ 的解空间 $\mathbb{V} = \{x | Ax = \mathbf{0}, x \in \mathbb{R}^n\}$ 是 \mathbb{R}^n 中的子空间,
- 方程组 $Ax = b$ 的解空间 $\mathbb{V} = \{x | Ax = b, x \in \mathbb{R}^n\}$, 不是子空间,
- 方程组 $Ax = \mathbf{0}$ 的解空间和 $Bx = \mathbf{0}$ 的解空间的交集是 \mathbb{R}^n 中的子空间。

子空间的并集仍是子空间么? 答案显然是否定的。

2.2.3 子空间的交、和、直和

接下来, 我们考虑几种运算, 子空间在经过这些运算后仍然是子空间。

定理 2.2.1. 设 \mathbb{Y}_1 与 \mathbb{Y}_2 都是数域 \mathbb{K} 上的线性空间 \mathbb{X} 的子空间。若用 $\mathbb{Y}_1 \cap \mathbb{Y}_2$ 表示 \mathbb{Y}_1 与 \mathbb{Y}_2 中的公共元素集合, 则 $\mathbb{Y}_1 \cap \mathbb{Y}_2$ 也是 \mathbb{X} 的子空间, 且称 $\mathbb{Y}_1 \cap \mathbb{Y}_2$ 为 \mathbb{Y}_1 与 \mathbb{Y}_2 的交。

由集合的交的定义可以看出, 子空间的交适合下列运算规律:

$$\mathbb{Y}_1 \cap \mathbb{Y}_2 = \mathbb{Y}_2 \cap \mathbb{Y}_1 \text{(交换律)}$$

$$(\mathbb{Y}_1 \cap \mathbb{Y}_2) \cap \mathbb{Y}_3 = \mathbb{Y}_1 \cap (\mathbb{Y}_2 \cap \mathbb{Y}_3) \text{(结合律)}$$

由结合律, 我们可以定义多个子空间的交:

$$\mathbb{Y}_1 \cap \mathbb{Y}_2 \cap \cdots \cap \mathbb{Y}_s = \bigcap_{i=1}^s \mathbb{Y}_i,$$

它也是子空间。

定义 2.2.3. 给定向量空间 \mathbb{X} 的两个子空间 $\mathbb{Y}_1, \mathbb{Y}_2$, 若用 $\mathbb{Y}_1 + \mathbb{Y}_2$ 表示全体形如 $\mathbf{y}_1 + \mathbf{y}_2 (\mathbf{y}_1 \in \mathbb{Y}_1, \mathbf{y}_2 \in \mathbb{Y}_2)$ 的向量组成的集合, 则 $\mathbb{Y}_1 + \mathbb{Y}_2$ 也是一个子空间, $\mathbb{Y}_1 + \mathbb{Y}_2$ 称为和。

可以看出, 子空间的和适合下列运算规律:

$$\mathbb{Y}_1 + \mathbb{Y}_2 = \mathbb{Y}_2 + \mathbb{Y}_1 \text{(交换律)}$$

$$(\mathbb{Y}_1 + \mathbb{Y}_2) + \mathbb{Y}_3 = \mathbb{Y}_1 + (\mathbb{Y}_2 + \mathbb{Y}_3) \text{(结合律)}$$

由结合律, 我们可以定义多个子空间的和

$$\mathbb{Y}_1 + \mathbb{Y}_2 + \cdots + \mathbb{Y}_s = \sum_{i=1}^s \mathbb{Y}_i$$

它是由所有表示成

$$\mathbf{a}_1 + \mathbf{a}_2 + \cdots + \mathbf{a}_s, \mathbf{a}_i \in \mathbb{Y}_i (i = 1, 2, \dots, s)$$

的向量组成的子空间。

我们着重区分一下子空间的“和”和集合的“并”的区别。

例 2.2.9. 设集合 $\mathbb{A} = \{(x, y) | y = 0, x \in \mathbb{R}\}, \mathbb{B} = \{(x, y) | x = 0, y \in \mathbb{R}\}$, 它们都是 \mathbb{R}^2 的子空间。

$\mathbb{A} \cup \mathbb{B} = \{(x, y) | xy = 0\}$, 而 $\mathbb{A} + \mathbb{B} = \mathbb{R}^2$ 。 \mathbb{A} 和 \mathbb{B} 的并集是所有 x -轴和 y -轴上的点, 而 \mathbb{A} 和 \mathbb{B} 的和是整个 xoy 平面。

定义 2.2.4. 如果 \mathbb{Y} 中的每个向量 \mathbf{x} 可唯一地表成 $\mathbf{x} = \mathbf{y}_1 + \mathbf{y}_2 (\mathbf{y}_1 \in \mathbb{Y}_1, \mathbf{y}_2 \in \mathbb{Y}_2)$ 的形式, 则称 \mathbb{Y} 为 \mathbb{Y}_1 与 \mathbb{Y}_2 的直和。记作 $\mathbb{Y} = \mathbb{Y}_1 + \mathbb{Y}_2$ 或 $\mathbb{Y}_1 \oplus \mathbb{Y}_2$

定理 2.2.2. 和 $\mathbb{Y}_1 + \mathbb{Y}_2$ 为直和的必要充分条件是: 由 $\mathbf{y}_1 + \mathbf{y}_2 = \mathbf{0} (\mathbf{y}_1 \in \mathbb{Y}_1, \mathbf{y}_2 \in \mathbb{Y}_2)$ 可推出 $\mathbf{y}_1 = \mathbf{y}_2 = \mathbf{0}$ 。

定理 2.2.3. 和 $\mathbb{Y}_1 + \mathbb{Y}_2$ 为直和的必要充分条件是: 由 $\mathbb{Y}_1 \cap \mathbb{Y}_2 = \{\mathbf{0}\}$ 。

推论 2.2.1. 和 $\mathbb{V}_1 + \mathbb{V}_2$ 为直和的充分必要条件是

$$\mathbb{V}_1 \cap \mathbb{V}_2 = \{\mathbf{0}\}$$

子空间的直和的概念可以推广到多个子空间的情形。

定义 2.2.5. 设 $\mathbb{V}_1, \mathbb{V}_2, \dots, \mathbb{V}_s$ 都是线性空间 \mathbb{V} 的子空间。如果和 $\mathbb{V}_1 + \mathbb{V}_2 + \cdots + \mathbb{V}_s$ 中每个向量 \mathbf{a} 的分解式

$$\mathbf{a} = \mathbf{a}_1 + \mathbf{a}_2 + \cdots + \mathbf{a}_s, \quad \mathbf{a}_i \in \mathbb{V}_i (i = 1, 2, \dots, s)$$

是唯一的, 这个和就称为直和。记为 $\mathbb{V}_1 \oplus \mathbb{V}_2 \oplus \cdots \oplus \mathbb{V}_s$ 。

和两个子空间的直和一样, 我们有

定理 2.2.4. $\mathbb{V}_1, \mathbb{V}_2, \dots, \mathbb{V}_s$ 是 \mathbb{V} 的一些子空间, 下面这些条件是等价的:

- 1) $\mathbb{W} = \sum \mathbb{V}_i$ 是直和;
- 2) 零向量的表示方法唯一;
- 3) $\mathbb{V}_i \cap \sum_{j \neq i} \mathbb{V}_j = \{\mathbf{0}\}, \quad (i = 1, 2, \dots, s)$.

子空间的直和反映了不同子空间的元素有某种“无关性”。

2.2.4 线性无关性

向量之间除了运算关系外还存在着各种关系, 其中最主要的关系是向量组的线性相关与线性无关。下面我们讨论这两个关系。

定义 2.2.6. 设向量 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_s$ 是数域 \mathbb{K} 上的 n 维向量组, k_1, k_2, \dots, k_s 是数域 \mathbb{K} 上的一组数, 那么表达式

$$k_1\mathbf{a}_1 + k_2\mathbf{a}_2 + \cdots + k_s\mathbf{a}_s.$$

称为向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_s$ 的一个线性组合, 而 k_1, k_2, \dots, k_s 称为组合系数。若向量 \mathbf{b} 是向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_s$ 的一个线性组合, 即

$$\mathbf{b} = k_1\mathbf{a}_1 + k_2\mathbf{a}_2 + \cdots + k_s\mathbf{a}_s,$$

则称 \mathbf{b} 可以由向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_s$ 线性表出。

例 2.2.10. 零向量 $\mathbf{0}$ 是任意向量组的线性组合, 这是因为 $\mathbf{0} = \sum_{i=1}^k 0\mathbf{b}_i$ 总是正确的。事实上我们更为关心 k_1, k_2, \dots, k_s 不全为零时的情况。

例如, 设向量组 $\mathbf{a}_1 = (2, -1, 3, 1)$, $\mathbf{a}_2 = (4, -2, 5, 4)$, $\mathbf{a}_3 = (2, -1, 4, -1)$, 则有 $\mathbf{a}_3 = 3\mathbf{a}_1 - \mathbf{a}_2$, 这表示 \mathbf{a}_3 可以由 $\mathbf{a}_1, \mathbf{a}_2$ 线性表出。

定义 2.2.7. 设 $\mathbf{a}_i \in \mathbb{K}^n (i = 1, 2, \dots, r)$ 。若在 \mathbb{K} 中存在 r 个不全为零的数 $\lambda_i (i = 1, 2, \dots, r)$, 使 $\sum_{i=1}^r \lambda_i \mathbf{a}_i = \mathbf{0}$, 则称向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r$ 线性相关。反之, 如果向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r$ 不线性相关, 即只有 $\lambda_1, \lambda_2, \dots, \lambda_r$ 全为零时, 才能使得 $\sum_{i=1}^r \lambda_i \mathbf{a}_i = \mathbf{0}$, 则称向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r$ 线性无关。

定义 2.2.8. 向量组的一部分组称为一个极大线性无关组, 如果这个部分组本身线性无关, 但从原向量组的其余向量中任取一个添加进去后, 所得的部分组都线性相关。

例 2.2.11. 一个地理范例可能有助于理解线性无关的概念。



图 2.6: 一个线性相关和线性无关的例子

如果你在上海，问一个人前往宣城的路线，他可能会说“你可以首先向西北方向行驶 180 公里到常州，然后再向西南方向 244.8 公里前往宣城。”通过这两句话我们足以知道宣城的位置。这个人可能会补充“那里距离这里大约是向西 282.9 公里。”虽然最后的这句陈述是正确的，但我们可以从先前的信息自己推断出来。

在这个例子中，“西北方向 180 公里”向量（红色）和“西南方向 244.8 公里”向量（黑色）是线性无关的。这意味着西南向量不能用西北向量来描述，反之亦然。然而，第三个“向西 282.9 公里”向量（蓝色）是其他两个向量的线性组合，它使得该组向量组线性相关。

例 2.2.12. 在向量组

$$\mathbf{a}_1 = (2, -1, 3, 1) \quad \mathbf{a}_2 = (4, -2, 5, 4) \quad \mathbf{a}_3 = (2, -1, 2, 3)$$

中，由 $\mathbf{a}_1, \mathbf{a}_2$ 组成的部分组就是一个极大线性无关组。首先， $\mathbf{a}_1, \mathbf{a}_2$ 线性无关，因为由

$$\begin{aligned} k_1\mathbf{a}_1 + k_2\mathbf{a}_2 &= k_1(2, -1, 3, 1) + k_2(4, -2, 5, 4) \\ &= (2k_1 + 4k_2, -k_1 - 2k_2, 3k_1 + 5k_2, k_1 + 4k_2) = (0, 0, 0, 0), \end{aligned}$$

就有 $k_1 = k_2 = 0$ ，同时， $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ 线性相关 ($\mathbf{a}_2 = \mathbf{a}_1 + \mathbf{a}_3$)。不难看出， $\mathbf{a}_2, \mathbf{a}_3$ 也是一个极大线性无关组。

如果两组向量组，它们能够线性表出的东西是相同的，那么利用这两组向量组对空间中的向量的表示能力是一样的，一个向量组能线性表示的，另一个向量组也能。一个向量组不能线性表示的，另一个向量组也不能。我们对这种同样的表示能力给出一个数学定义。

定义 2.2.9. 设 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_s$ 和 $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_t$ 是数域 \mathbb{K} 上的两个向量组，如果向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_s$ 中每一个向量 \mathbf{a}_i ($i = 1, 2, \dots, s$) 都可以用向量组 $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_t$ 线性表出，那么称向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_s$ 可以用向量组 $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_t$ 线性表出。如果两个向量组可以互相线性表出，则称为它们等价。

例 2.2.13. 例如, 设

$$\mathbf{a}_1 = (1, 0), \mathbf{a}_2 = (0, 1);$$

$$\mathbf{b}_1 = (1, 1), \mathbf{b}_2 = (-1, 1),$$

则向量组 $\mathbf{a}_1, \mathbf{a}_2$ 与向量组 $\mathbf{b}_1, \mathbf{b}_2$ 是等价的。

$$\begin{aligned}\mathbf{a}_1 &= \frac{1}{2}\mathbf{b}_1 - \frac{1}{2}\mathbf{b}_2, & \mathbf{a}_2 &= \frac{1}{2}\mathbf{b}_1 + \frac{1}{2}\mathbf{b}_2 \\ \mathbf{b}_1 &= \mathbf{a}_1 + \mathbf{a}_2, & \mathbf{b}_2 &= -\mathbf{a}_1 + \mathbf{a}_2\end{aligned}$$

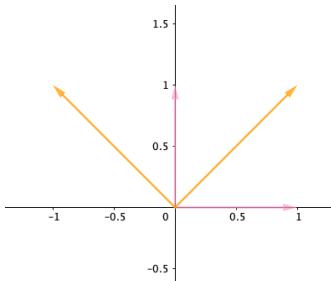


图 2.7: 向量组等价

2.2.5 生成集、基底与坐标

在例 2.2.13 中, 对于 2 维向量空间中的任何一个向量, 我们既可以用 $\mathbf{a}_1, \mathbf{a}_2$ 的线性组合表示, 也可以用 $\mathbf{b}_1, \mathbf{b}_2$ 的线性组合表示。换言之, 能用 $\mathbf{a}_1, \mathbf{a}_2$ 的线性组合表示, 和用 $\mathbf{b}_1, \mathbf{b}_2$ 的线性组合表示的向量所形成的向量空间是相同的, 都是 \mathbb{R}^2 。我们引入生成集的概念。

定义 2.2.10. 设 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r$ 是 \mathbb{V} 的一组向量, 则这组向量所有可能的线性组合 $\sum_{k=1}^r \lambda_k \mathbf{a}_k$ 所成的集合是 \mathbb{V} 的一个子空间, 称为由 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r$ 张成的子空间, 记作 $L(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r)$ 或 $\text{span}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r)$ 。 $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r\}$ 叫做 \mathbb{V} 的一个生成集。

定理 2.2.5. 两个向量组张成相同的子空间的充分必要条件是: 这两个向量组等价。

生成集可以张成的线性子空间。在这个线性子空间的每一个向量能被这个向量组(生成集)线性表出。

对于 3 维几何空间中的向量, 线性无关的向量最多是 3 个, 而任意 4 个向量都是线性相关的, 也就是存在某个向量可以用其他 3 个向量线性表出, 在这 3 个向量的生成集中。对于 n 元数组所构成的向量空间, 有 n 个线性无关的向量, 而任意 $n+1$ 个向量都是线性相关的, 也就是存在某个向量可以用其他 n 个向量线性表出, 在这 n 个向量的生成集中。现在我们要找出能够张成一个线性子空间的最小生成集, 以及这个最小生成集中, 应该有多少个向量。

定义 2.2.11. 如果在向量空间 \mathbb{V} 中有 n 个线性无关的向量 a_1, a_2, \dots, a_n , 且 \mathbb{V} 中任一向量都可以用它们线性表出, 则称 \mathbb{V} 为 \mathbb{K} 上的 n 维线性空间, n 称为 \mathbb{V} 的维数, 记作 $\dim(\mathbb{V}) = n$ 。而 a_1, a_2, \dots, a_n 就是 \mathbb{V} 的一组基。

例 2.2.14. 复数域 \mathbb{C} 在 \mathbb{C} 上和 \mathbb{R} 上是两个不同的向量空间。

- 因为在 \mathbb{C} 上它是一维的, 数 1 就是一组基;
- 而在 \mathbb{R} 上它是二维的, 数 1 与 i 就是一组基。

这个例子告诉我们, 维数是和所考虑的数域有关的。

定理 2.2.6. 令 \mathbb{V} 是一向量空间, $\mathbb{B} \subseteq \mathbb{V}, \mathbb{B} \neq \emptyset$ 下列命题等价:

- \mathbb{B} 是 \mathbb{V} 的一个基
- \mathbb{B} 是最小生成集
- \mathbb{B} 是 \mathbb{V} 中的极大线性无关组
- \mathbb{V} 中每一个向量能被 \mathbb{B} 线性表出

定义 2.2.12. 如果一组基中的每一个向量长度均为 1, 我们称其为标准基。

在后面的课程中, 我们将会严格说明向量的长度。

例 2.2.15. 在 \mathbb{R}^3 中, 常用基 $\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$ 就是一组标准基。

例 2.2.16. 对于一个由向量 x_1, x_2, x_3 张成的向量空间 $\mathbb{U} \subseteq \mathbb{R}^4$

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

我们关心 x_1, x_2, x_3 是否是 \mathbb{U} 的一组基。为此, 我们需要确认 x_1, x_2, x_3 是否线性无关。因此, 我们需要解 $\sum_{i=1}^3 \lambda_i x_i = \mathbf{0}$

这是一个关于下面这个矩阵的一个线性方程组, 并且我们对这个矩阵作行初等变换可将其化成阶梯型

$$(x_1, x_2, x_3) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

从而我们可以发现 $\mathbf{x}_1, \mathbf{x}_2$ 是线性无关的, $\lambda_1\mathbf{x}_1 + \lambda_2\mathbf{x}_2 = \mathbf{0}$ 只有 $\lambda_1 = \lambda_2 = 0$ 时成立。因此 $\{\mathbf{x}_1, \mathbf{x}_2\}$ 是 \mathbb{U} 的一组基。

这个例子说明, \mathbb{U} 是 \mathbb{R}^4 中的 2 维向量空间。如果我们添加向量 $\mathbf{e}_3 = (0, 0, 1, 0)^T$, $\mathbf{e}_4 = (0, 0, 0, 1)^T$, 那么因为 $\mathbf{e}_1 = \mathbf{x}_1 - \mathbf{e}_3$, $\mathbf{e}_2 = \mathbf{x}_2 - \mathbf{e}_3 - \mathbf{e}_4$, 则 $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4$ 可以由 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{e}_3, \mathbf{e}_4$ 线性表出, 也就是 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{e}_3, \mathbf{e}_4$ 生成的子空间为 \mathbb{R}^4 。

定理 2.2.7. 设 $\mathbb{Y} = L(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m)$ 是 n 维空间 \mathbb{X} 的一个 m 维子空间, 则向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ 可扩张为 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m, \mathbf{a}_{m+1}, \dots, \mathbf{a}_n$ 使 $\mathbb{X} = L(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m, \mathbf{a}_{m+1}, \dots, \mathbf{a}_n)$ 。

注意: 其中 $L(\mathbf{a}_{m+1}, \dots, \mathbf{a}_n)$ 也是 \mathbb{Y} 的一个子空间。

$$L(\mathbf{a}_{m+1}, \dots, \mathbf{a}_n) \oplus L(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m) = L(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m, \mathbf{a}_{m+1}, \dots, \mathbf{a}_n).$$

我们给出两个子空间和的维数与各自维数的关系。

定理 2.2.8. 维数公式: $\dim(\mathbb{Y}_1 + \mathbb{Y}_2) = \dim \mathbb{Y}_1 + \dim \mathbb{Y}_2 - \dim(\mathbb{Y}_1 \cap \mathbb{Y}_2)$

对于直和: $\dim(\mathbb{Y}_1 \oplus \mathbb{Y}_2) = \dim \mathbb{Y}_1 + \dim \mathbb{Y}_2$

定义 2.2.13. 如果一个向量空间 \mathbb{V} 中任一向量都能被 n 个线性无关的向量线性表出时, \mathbb{V} 称为有限维线性空间, 否则, 称为无限维线性空间。

按照这个定义, 不难看出, 几何空间中向量所成的向量空间是三维的; n 元数组所构成的空间是 n 维的; 由所有实系数多项式所成的实线性空间是无限维的, 但是次数小于 n 的实多项式空间是 n 维的, 因为对于任意的 n , 都有 n 个线性无关的向量 $1, x, \dots, x^{n-1}$ 可以线性表示出所有次数小于 n 的多项式。

无限维空间是一个专门研究的对象, 它与有限维空间有比较大的差别。但是, 上面提到的线性表出, 线性相关, 线性无关等性质, 只要不涉及维数和基, 就对无限维空间成立。在本课程中, 我们主要讨论有限维空间。

在解析几何中我们看到, 为了研究向量的性质, 引入坐标是一个重要的步骤。对于有限维向量空间, 坐标同样是一个有力的工具。

定义 2.2.14. 在 n 维向量空间 \mathbb{V} 中, n 个线性无关的向量 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 称为 \mathbb{V} 的一组基。设 \mathbf{a} 是 \mathbb{V} 中任一向量, 于是 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n, \mathbf{a}$ 线性相关, 因此 \mathbf{a} 可以被基 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 线性表出:

$$\mathbf{a} = a_1\varepsilon_1 + a_2\varepsilon_2 + \dots + a_n\varepsilon_n,$$

其中系数 a_1, a_2, \dots, a_n 是被向量 \mathbf{a} 和基 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 唯一确定的, 这组数就称为 \mathbf{a} 在基 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 下的坐标, 记为 $(a_1, a_2, \dots, a_n)^T$ 。

下面我们来看几个例子。

例 2.2.17. 在向量空间 P_n 中,

$$1, x, x^2, \dots, x^{n-1}$$

是 n 个线性无关的向量, 而且每一个次数小于 n 的数域 \mathbb{K} 上的多项式都可以被它们线性表出, 所以 P_n 是 n 维的, 而 $1, x, x^2, \dots, x^{n-1}$ 就是它的一组基。在这组基下, 多项式 $f(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1}$ 的坐标就是它的系数 $(a_0, a_1, \dots, a_{n-1})^T$ 。

如果在 \mathbb{V} 中取另外一组基

$$\varepsilon_1' = 1, \varepsilon_2' = (x - a), \dots, \varepsilon_n' = (x - a)^{n-1}.$$

那么按泰勒展开公式

$$f(x) = f(a) + f'(a)(x - a) + \dots + \frac{f^{(n-1)}(a)}{(n-1)!}(x - a)^{n-1}.$$

因此, $f(x)$ 在基 $\varepsilon_1', \varepsilon_2', \dots, \varepsilon_n'$ 下的坐标是

$$\left(f(a), f'(a), \dots, \frac{f^{(n-1)}(a)}{(n-1)!} \right)^T.$$

例 2.2.18. 在 n 维向量空间 \mathbb{V} 中, 显然

$$\begin{cases} \varepsilon_1 = (1, 0, \dots, 0), \\ \varepsilon_2 = (0, 1, \dots, 0), \\ \dots \\ \varepsilon_n = (0, 0, \dots, 1) \end{cases}$$

是一组基。对每一个向量 $\mathbf{a} = (a_1, a_2, \dots, a_n)$, 都有

$$\mathbf{a} = a_1\varepsilon_1 + a_2\varepsilon_2 + \dots + a_n\varepsilon_n.$$

所以 $(a_1, a_2, \dots, a_n)^T$ 就是向量 \mathbf{a} 在这组基下的坐标。

不难证明,

$$\begin{cases} \varepsilon_1' = (1, 1, \dots, 1), \\ \varepsilon_2' = (0, 1, \dots, 1), \\ \dots \\ \varepsilon_n' = (0, 0, \dots, 1) \end{cases}$$

是 \mathbb{V} 中 n 个线性无关的向量。在基 $\varepsilon_1', \varepsilon_2', \dots, \varepsilon_n'$ 下, 对于向量 $\mathbf{a} = (a_1, a_2, \dots, a_n)$, 有

$$\mathbf{a} = a_1\varepsilon_1' + (a_2 - a_1)\varepsilon_2' + \dots + (a_n - a_{n-1})\varepsilon_n'.$$

因此, \mathbf{a} 在基 $\varepsilon_1', \varepsilon_2', \dots, \varepsilon_n'$ 下的坐标为

$$(a_1, a_2 - a_1, \dots, a_n - a_{n-1})^T.$$

2.2.6 秩

接下来我们将介绍矩阵的秩。有了极大线性无关组、维数等概念的铺垫，我们很容易理解秩的概念。

定义 2.2.15. 向量组 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r$ 的极大线性无关组中所含向量的个数称为这个向量组的秩，记作 $\text{rank}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r\}$ 。

此外我们规定：由零向量组成的向量组的秩为零。

定义 2.2.16. 矩阵 A 的行(列)向量组的秩称为 A 的行秩(列秩)，其中矩阵的行秩和列秩相等，它们都称为矩阵 A 的秩，记作 $\text{rank}(A)$ 。

定理 2.2.9. $\dim L(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r) = \text{rank}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r\}$

例 2.2.19. 设矩阵

$$A = \begin{pmatrix} 1 & 1 & 3 & 1 \\ 0 & 2 & -1 & 4 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

A 的行向量组为

$$\mathbf{a}_1 = (1, 1, 3, 1) \quad \mathbf{a}_2 = (0, 2, -1, 4)$$

$$\mathbf{a}_3 = (0, 0, 0, 5) \quad \mathbf{a}_4 = (0, 0, 0, 0).$$

可以证明， $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ 是向量组 $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$ 的一个极大线性无关组。因此，向量组 $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$ 的秩为 3，换句话说，矩阵 A 的行秩为 3。 A 的列向量组为

$$\mathbf{b}_1 = (1, 0, 0, 0), \mathbf{b}_2 = (1, 2, 0, 0),$$

$$\mathbf{b}_3 = (3, -1, 0, 0), \mathbf{b}_4 = (1, 4, 5, 0).$$

用同样的方法可以证明， $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_4$ 线性无关，且 $\mathbf{b}_3 = \frac{7}{2}\mathbf{b}_1 - \frac{1}{2}\mathbf{b}_2$ ，所以 $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_4$ 是向量组 $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4$ 的一个极大线性无关组，于是向量组 $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4$ 的秩为 3，换句话说，矩阵 A 的列秩为 3。

定理 2.2.10. 对于 $m \times n$ 的矩阵 A ，假设其秩为 r ，则存在秩同样为 r 两个矩阵： $F_{m \times r}$ （列满秩）和 $G_{r \times n}$ （行满秩），使得 $A = FG$ ，把这种分解称其为矩阵 A 的满秩分解。

证明：由于矩阵 A 的秩为 r ，可以通过一系列初等行变换 P 和初等列变换 Q 使得

$$PAQ = \begin{pmatrix} I_r & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{pmatrix}$$

由于初等行列变换都是可逆的，

$$\mathbf{A} = \mathbf{P}^{-1} \begin{pmatrix} \mathbf{I}_r & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{pmatrix} \mathbf{Q}^{-1}$$

取 \mathbf{P}^{-1} 的前 r 列作为 \mathbf{F} , \mathbf{Q}^{-1} 的前 r 行作为 \mathbf{G} , 即为满足条件 $\mathbf{A} = \mathbf{FG}$ 的一组矩阵。这样的矩阵分解并不是唯一的。可以对 \mathbf{F} 右乘可逆矩阵 \mathbf{X} , \mathbf{G} 左乘 \mathbf{X}^{-1} , 则 $\mathbf{F}_1 = \mathbf{FX}$, $\mathbf{G}_1 = \mathbf{X}^{-1}\mathbf{G}$ 也是满足条件的一组矩阵。

$\mathbf{A} = \mathbf{FG}$, 也就是说

$$\mathbf{A} = \mathbf{f}_1\mathbf{g}_1 + \mathbf{f}_2\mathbf{g}_2 + \cdots + \mathbf{f}_r\mathbf{g}_r$$

其中 \mathbf{f}_i 是 \mathbf{F} 的第 i 列, \mathbf{g}_i 是 \mathbf{G} 的第 i 行。 $\mathbf{f}_i\mathbf{g}_i$ 都是秩为 1 的矩阵, 因此, 这种分解也称为秩-1 分解。也就是说, 对于秩为 r 的矩阵, 可以写成 r 个秩-1 矩阵和的形式。显然, k 个秩-1 矩阵和的矩阵其秩最多为 k 。也就是说, 如果想把秩为 r 的矩阵写成若干个秩-1 矩阵和的形式, 至少需要 r 个。□

例 2.2.20. 一般在推荐系统中, 数据往往使用“用户——物品”矩阵来表示的。用户对其接触过的物品进行评分, 评分表示了用户对于物品的喜爱程度, 分数越高, 表示用户越喜欢这个物品。而这个矩阵往往是稀疏的, 空白项是用户还未接触到的物品, 推荐系统的任务则是选择其中的部分物品推荐给用户。这就需要对矩阵中的空白项进行补全。

	物品1	物品2	物品3	物品4	物品5	物品6	物品7	物品8	物品9	物品10
用户1	3					5			2	
用户2			3		5			2		
用户3		1		2			5			
用户4			3					3		5
用户5	5				2					

图 2.8: 用户-物品表

矩阵补全问题可以转化为寻找与观测到数据集合 \mathbb{E} 中所有项匹配的最低秩评分矩阵。形式化如下

$$\begin{aligned} \min_{\mathbf{X}} \quad & \text{rank}(\mathbf{X}) \\ \text{s.t.} \quad & \mathbf{X}_{ij} = \mathbf{M}_{ij} \quad \forall i, j \in \mathbb{E} \end{aligned}$$

其中 \mathbb{E} 为可以被观察到评分的(用户, 物品)指标集, \mathbf{M} 为观察评分矩阵, \mathbf{M}_{ij} 为观测到的用户 i 对物品 j 的评分, \mathbf{X} 为预测评分矩阵, \mathbf{X}_{ij} 为预测的用户 i 对物品 j 的评分。

或者转化为限定在秩为 r 的条件下, 求矩阵使得观测到的评分与预测的评分矩阵对应项最接近:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \sum_{ij} (\mathbf{X}_{ij} - \mathbf{M}_{ij})^2 \quad \forall i, j \in \mathbb{E} \\ \text{s.t.} \quad & \text{rank}(\mathbf{X}) = r \end{aligned}$$

利用秩-1 分解，将 \mathbf{X} 看作 $\sum_{i=k}^r \mathbf{f}_k \mathbf{g}_k$ ，其中 \mathbf{f} 是列向量， \mathbf{g} 是行向量。 $\mathbf{X}_{ij} = \sum_k \mathbf{f}_k[i] \mathbf{g}_k[j]$ ，优化问题可以进一步写作：

$$\min_{\mathbf{f}_k, \mathbf{g}_k, 1 \leq k \leq r} \quad \sum_{ij} \left(\sum_{i=k}^r \mathbf{f}_k[i] \mathbf{g}_k[j] - M_{ij} \right)^2 \quad \forall i, j \in \mathbb{E}$$

2.2.7 仿射空间

仿射子空间和子空间密切相关，可以看作子空间的推广。我们常常假定数据分布在一个仿射子空间上，也就是一个子空间加上一个偏移量。

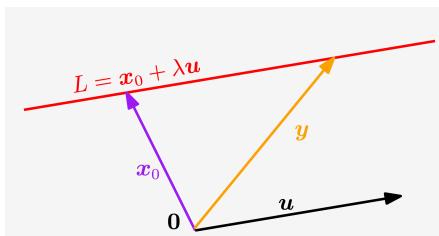


图 2.9: 仿射子空间

定义 2.2.17. 令 \mathbb{V} 是一线性空间， $\mathbf{x}_0 \in \mathbb{V}$ 且 $\mathbb{U} \subseteq \mathbb{V}$ 是一线性子空间，则子集

$$\mathbb{L} = \mathbf{x}_0 + \mathbb{U} := \{\mathbf{x}_0 + \mathbf{u} | \mathbf{u} \in \mathbb{U}\} \subseteq \mathbb{V}$$

是一仿射子空间。我们定义线性子空间的维数为仿射子空间的维数。

注意，如果 $\mathbf{x}_0 \notin \mathbb{U}$ ，则仿射子空间 \mathbb{L} 不是一个线性子空间。若 \mathbb{U} 有一基底 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ ，则 \mathbb{L} 中的每一个元素 \mathbf{x} 均可写成 $\mathbf{x}_0 + k_1 \mathbf{a}_1 + k_2 \mathbf{a}_2 + \dots + k_m \mathbf{a}_m$ 。通过定义是容易得到这一结论。

例 2.2.21. \mathbb{R}^3 中常见的仿射子空间：

1. 零维仿射子空间：单点集 $\{\mathbf{x}_0\}$ ；
2. 一维仿射子空间：直线 $\{\mathbf{x}_0 + k\mathbf{u}\}$ ；
3. 二维仿射子空间：平面 $\{\mathbf{x}_0 + k_1 \mathbf{u}_1 + k_2 \mathbf{u}_2\}$ ；
4. \mathbb{R}^3 本身；

例 2.2.22. 我们已经知道线性方程组 $A\mathbf{x} = \mathbf{b}, \mathbf{b} \neq \mathbf{0}$ 的解空间不是一个线性空间，但是它的解空间是一个仿射空间。

证明。设 $A\mathbf{x} = \mathbf{0}$ 的解空间为 \mathbb{V} ，它是一个子空间；设 \mathbf{x}_0 是 $A\mathbf{x} = \mathbf{b}$ 的一个特解。

$\forall \mathbf{x} \in \mathbf{x}_0 + \mathbb{V}$ ， \mathbf{x} 必可以写成 $\mathbf{x} = \mathbf{x}_0 + \mathbf{x}_1$ ，其中 $\mathbf{x}_1 \in \mathbb{V}$ 。显然，

$$A\mathbf{x} = A(\mathbf{x}_0 + \mathbf{x}_1) = A\mathbf{x}_0 + A\mathbf{x}_1 = \mathbf{0} + \mathbf{b} = \mathbf{b}.$$

说明 $x_0 + \mathbb{V} \subseteq \{x | Ax = b\}$

反之, $\forall x$ 满足 $Ax = b$, 则 $Ax - Ax_0 = A(x - x_0) = \mathbf{0}$, 则

$$x - x_0 \in \mathbb{V}, \quad x \in x_0 + \mathbb{V}.$$

说明 $\{x | Ax = b\} \subseteq x_0 + \mathbb{V}$ 。

综上, 线性方程组 $Ax = b, b \neq \mathbf{0}$ 的解空间为 $x_0 + \mathbb{V}$, 这是一个仿射空间。 \square

注记 7. 本节开头的例子说明, 类似像鸢尾花数据集合、图像集合或文档集合中可能每一个元素对应一个高维特征向量, 假设所有元素落在一个高维向量空间, 但是这些元素通常只有几个维度的内蕴特征, 要找到这个特征, 就要想办法把它们“拉回”低维的内蕴特征向量空间, 这些低维向量空间通常作为高维特征空间的子空间, 这种操作在机器学习和数据分析领域通常把它称为降维或特征选择。那么怎么拉回呢? 这涉及到映射和投影。我们在下一节以及 3.2 节会来介绍这些内容。

注记 8. 另外, 应该注意到, 我们这里定义的向量空间是数学上“纯粹”的向量空间, 也就是满足加法和数乘运算的封闭性, 并且子空间也继承了这一性质。但是在数据分析和机器学习领域, 很多时候, 我们说的向量空间并不是数学上“纯粹”的向量空间, 而是附加了额外数学结构的向量空间, 比如距离结构或更一般的度量结构。为什么要这么做呢? 这大致有两种考虑: 一是很多实际问题中数据对象虽然构成了集合, 但是这些对象对加法或数乘可能是没有意义的, 因此我们只是假设它们构成向量空间或者通过进一步的处理使它们构成向量空间; 二是在数据科学领域, 我们往往对数据对象的相似性更感兴趣, 因此需要在数据集合或假设的向量空间上引入相似性度量, 这种度量可以通过比如距离结构来定义。例如我们在 2.1 节介绍了用文本表示向量的例子, 事实上, 向量空间模型是处理文本最基本的模型。在向量空间模型中, 把对文本内容的处理简化为向量空间中的向量运算, 并且它以空间上的向量相似度表达语义的相似度。把文本表示为文档空间的向量, 就可以通过计算向量之间的相似性来度量文档间的相似性。文本处理中最常用的相似性度量方式是余弦距离。我们将在 3.1 节来讨论, 如何在向量空间的基础上, 引入范数、内积、角度和正交性等概念, 建立满足数据分析所需的特殊的代数结构, 并形成特殊的向量空间或线性空间。

2.3 线性映射与线性变换

前面我们讨论了向量空间内向量的有关简单运算: 向量加法、向量与标量的乘法, 但尚未涉及两个向量空间之间的转换关系。然而, 在自然科学、社会科学和数学的一些分支中, 不同向量空间内向量之间的线性变换起着重要的作用。因此, 为了研究两个向量空间之间的关系, 有必要考虑能够实现从一个向量空间到另一个向量空间转换的函数。在我们的日常生活中, 也经常遇到这种转换。当我们欲将一幅图像变换为另一幅图像时, 通常会移动它的位置, 或者旋转

它。例如，函数 $\mathbf{T}(x, y) = (\alpha x, \beta y)$ 就能够将图像的 x 坐标和 y 坐标改变尺度。根据 α 和 β 大于 1 还是小于 1，图像就能够被放大或者缩小。

事实上，从数据处理的角度来看，我们主要面对两个空间，也即原始数据的输入空间和最后结果的输出空间。当然，在输入空间和输出空间之间还可能有一些中间的隐空间，这在深度学习领域非常常见。因此，从非概率的角度来看，数据分析和机器学习的基本问题之一就是寻找输入空间的和输出空间之间一个“好”的映射关系或函数关系。这个映射关系或函数关系通常称为数据模型，是数据分析或机器学习系统最重要的组成部分。数据模型与各种具体的数据分析或机器学习任务，如分类、回归和降维等关联，就会形成所谓的分类模型、回归模型和降维方法等。在这些模型中，如果从映射或函数关系是线性或非线性的角度看，又会被分为所谓的线性模型和非线性模型，比如，机器学习中常见的线性回归就是最基本的线性模型，而深度学习模型通常都是由线性映射和非线性映射复合而成的非线性模型。

下面我们来从映射的定义出发，来介绍线性映射和线性变换，并讨论在机器学习数据模型中的应用和联系。

2.3.1 映射

映射的定义

定义 2.3.1. 设 \mathbb{V}, \mathbb{W} 是两个非空集合，如果存在一个法则 f ，使得对 \mathbb{V} 中每个元素 v ，按法则 f ，在 \mathbb{W} 中有唯一确定的元素 w 与之对应，则称 f 为从 \mathbb{V} 到 \mathbb{W} 的映射，记作

$$f : \mathbb{V} \rightarrow \mathbb{W},$$

其中 w 称为元素 v （在映射 f 下）的像，并记作 $f(v)$ ，即

$$w = f(v),$$

而元素 v 称为元素 w （在映射 f 下）的一个原像；集合 \mathbb{V} 称为映射 f 的定义域，记作 D_f ，即 $D_f = \mathbb{V}$ ； \mathbb{V} 中所有元素的像所组成的集合称为映射 f 的值域，记作 R_f 或 $f(\mathbb{V})$ ，即

$$R_f = f(V) = \{f(v) | v \in \mathbb{V}\}.$$

从上述映射的定义中，需要注意的是：

(1) 构成一个映射必须具备以下三个要素：集合 \mathbb{V} ，即定义域 $D_f = \mathbb{V}$ ；集合 \mathbb{W} ，即值域的范围： $R_f \subset \mathbb{W}$ ；对应法则 f ，使对每个 $v \in \mathbb{V}$ ，有唯一确定的 $w = f(v)$ 与之对应。

(2) 对每个 $v \in \mathbb{V}$ ，元素 v 的像 w 是唯一的；而对每个 $w \in R_f$ ，元素 w 的原像不一定是唯一的；映射 f 的值域 R_f 是 \mathbb{W} 的一个子集，即 $R_f \subset \mathbb{W}$ ，不一定 $R_f = \mathbb{W}$ 。

图像中的分类问题，可以看作是学习从分类到标签的映射。在 MNIST 数据集中，每张图片为 28×28 的灰度图像，可以看作是 28×28 维的向量，我们希望找到一个映射，将其映射到分类标签数字上。这里数字图像数据集可以看作集合 \mathbb{V} ，每一张图片就是 \mathbb{V} 中的元素 v 。分类标签集合 $\{0, 1, 2, \dots, 9\}$ 就是 \mathbb{W} ，每个标签数字是 w 。

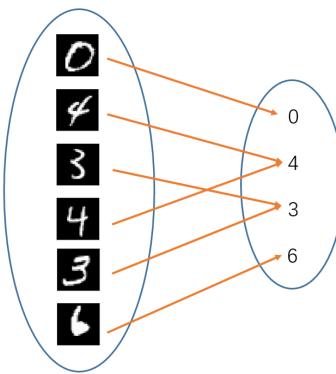


图 2.10: MNIST 数字分类问题是从数字图片数据集到数字集合的映射

定义 2.3.2. 设 \mathbb{V}, \mathbb{W} 是数域 \mathbb{K} 上的两个有限维的向量空间, $\varphi : \mathbb{V} \rightarrow \mathbb{W}$, 如果 φ 满足

1. $\forall x, y \in \mathbb{V} : \varphi(x) = \varphi(y) \implies x = y$, 则 φ 称为单射;
2. $\varphi(\mathbb{V}) = \mathbb{W}$, 则 φ 称为满射;
3. 即满足单射又满足满射, 则 φ 称为双射。

例 2.3.1. 下列映射中

1. $\Phi_1 : \mathbb{R}^n \rightarrow \mathbb{R}^n, \Phi_1(x) = 2x$ 既是单射又是满射, 所以是双射。
2. $\Phi_2 : \mathbb{R}^n \rightarrow \mathbb{R}, \Phi_2(x) = \max(x_i)$ 只是满射, 不是单射。
3. $\Phi_3 : \mathbb{R}^n \rightarrow \mathbb{R}, \Phi_3(x) = \max(|x_i|)$ 既不是单射也不是满射。

映射又称为算子, 根据集合 \mathbb{V}, \mathbb{W} 的不同情形, 在不同的数学分支中, 映射又有不同的惯用名称。例如, 从非空集合 \mathbb{V} 到数集 \mathbb{W} 的映射又称为 \mathbb{V} 上的泛函, 从非空集 \mathbb{V} 到它自身的映射又称为 \mathbb{V} 上的变换, 从实数集(或其子集) \mathbb{V} 到实数集 \mathbb{W} 的映射通常称为定义在 \mathbb{V} 上的函数。注: 如果不加说明, 本书考虑的集合均为有限维的向量空间。

定义 2.3.3. 设有两个映射

$$g : \mathbb{U} \rightarrow \mathbb{V}_1, f : \mathbb{V}_2 \rightarrow \mathbb{W},$$

其中 $\mathbb{V}_1 \subset \mathbb{V}_2$ 。则由映射 g 和 f 可以定出一个从 \mathbb{U} 到 \mathbb{W} 的对应法则, 它将每个 $u \in \mathbb{U}$ 映成 $f[g(u)] \in \mathbb{W}$ 。显然, 这个对应法则确定了一个从 \mathbb{U} 到 \mathbb{W} 的映射, 这个映射称为映射 g 和 f 构成的复合映射, 记作 $f \circ g$, 即

$$f \circ g : \mathbb{U} \rightarrow \mathbb{W},$$

$$(f \circ g)(v) = f[g(v)], v \in \mathbb{V}.$$

由复合映射的定义可知, 映射 g 和 f 构成复合映射的条件是: g 的值域 R_g 必须包含在 f 的定义域内, 即 $R_g \subset D_f$ 。否则, 不能构成复合映射, 由此可以知道, 映射 g 和 f 的复合是有顺序的, $f \circ g$ 有意义并不表示 $g \circ f$ 也有意义。即使 $f \circ g$ 与 $g \circ f$ 都有意义, 复合映射 $f \circ g$ 与 $g \circ f$ 也未必相同。

2.3.2 线性映射: 线性模型的观点

线性映射的定义

定义 2.3.4. 设 \mathbb{V}, \mathbb{W} 是数域 \mathbb{K} 上的两个有限维的向量空间, φ 是 \mathbb{V} 到 \mathbb{W} 的一个映射 ($\varphi : \mathbb{V} \rightarrow \mathbb{W}$)。如果对任何向量 $\mathbf{x}, \mathbf{y} \in \mathbb{V}$ 及任意的 $\alpha, \beta \in \mathbb{K}$, 有

$$\varphi(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha\varphi(\mathbf{x}) + \beta\varphi(\mathbf{y}),$$

则称 φ 为 \mathbb{V} 到 \mathbb{W} 的线性映射。

在上述定义中, $\varphi(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha\varphi(\mathbf{x}) + \beta\varphi(\mathbf{y})$, 即线性是叠加性和齐次性的合称。更一般地, 有

$$\varphi(c_1\mathbf{u}_1 + \cdots + c_p\mathbf{u}_p) = c_1\varphi(\mathbf{u}_1) + \cdots + c_p\varphi(\mathbf{u}_p)$$

例 2.3.2. 考虑映射 $\epsilon : \mathbb{V} \rightarrow \mathbb{V}$, $\epsilon(\mathbf{x}) = \mathbf{x}$, 我们称这种映射为恒等映射

$$\epsilon(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha\mathbf{x} + \beta\mathbf{y} = \alpha\epsilon(\mathbf{x}) + \beta\epsilon(\mathbf{y})$$

恒等映射是线性映射。

例 2.3.3. 考察映射 $\mathcal{T} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$

$$\mathcal{T}_1(\mathbf{x}) = \begin{bmatrix} x_1 + x_2 \\ x_1^2 - x_2^2 \end{bmatrix}, \text{ 其中, } \mathbf{x} = [x_1, x_2, x_3]^T$$

$$\mathcal{T}_2(\mathbf{x}) = \begin{bmatrix} x_1 - x_2 \\ x_2 + x_3 \end{bmatrix}, \text{ 其中, } \mathbf{x} = [x_1, x_2, x_3]^T$$

容易看出, 映射 $\mathcal{T}_1 : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ 不满足线性关系式, 故不是线性映射; 而映射 $\mathcal{T}_2 : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ 满足线性关系式, 为线性映射。

例 2.3.4. 考虑映射 $\mathcal{T}_{\mathbf{Q}}(A) = \mathbf{Q}^{-1}\mathbf{A}\mathbf{Q}$, 其中 \mathbf{Q} 是可逆矩阵。

$$\begin{aligned} \mathcal{T}_{\mathbf{Q}}(\alpha\mathbf{A} + \beta\mathbf{B}) &= \mathbf{Q}^{-1}(\alpha\mathbf{A} + \beta\mathbf{B})\mathbf{Q} \\ &= \alpha\mathbf{Q}^{-1}\mathbf{A}\mathbf{Q} + \beta\mathbf{Q}^{-1}\mathbf{B}\mathbf{Q} \\ &= \alpha\mathcal{T}_{\mathbf{Q}}(\mathbf{A}) + \beta\mathcal{T}_{\mathbf{Q}}(\mathbf{B}) \end{aligned}$$

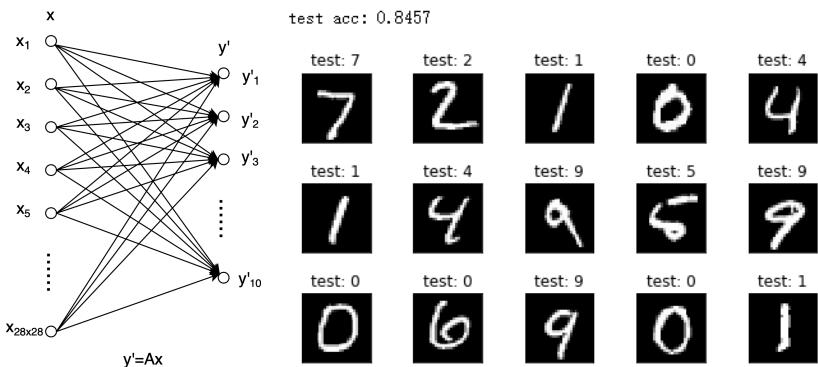


图 2.11: 线性映射分类准确率

在 MNIST 数字识别的例子中，我们把标签不再看作一个数字，如果标签为 i ，那么我们把它看作只有第 i 个分量为 1，其余分量为 0 的 10 维向量 \mathbf{y} ，所有标签向量在 10 维向量空间 \mathbb{V} 中。我们把图像数据集看作 28×28 维向量空间 \mathbb{W} ，想要找到一个映射，将 28×28 维向量空间映射到 10 维向量空间中去。

假设我们使用线性映射来完成这件事，设一张图片向量为 \mathbf{x} ，其标签向量为 \mathbf{y} ，通过我们选择的线性映射

$$f : \mathbb{W} \rightarrow \mathbb{V}$$

$$\mathbf{y}' = \mathbf{A}\mathbf{x}$$

其中 \mathbf{y}' 也是 \mathbb{V} 中的向量，希望 $\mathbf{y}' \approx \mathbf{y}$ 。选择合适的损失函数，通过优化得到 \mathbf{A} 。

同态、同构、自同态、自同构

考虑两个向量空间之间的一些特殊映射。

定义 2.3.5. 设 \mathbb{V}, \mathbb{W} 是数域 \mathbb{K} 上的任意两个集合，如果 φ 满足

- (1) $\varphi : \mathbb{V} \rightarrow \mathbb{W}$ 是线性映射，则 \mathbb{V}, \mathbb{W} 同态(Homomorphism)， φ 称为同态映射；
- (2) $\varphi : \mathbb{V} \rightarrow \mathbb{W}$ 是线性映射且是双射，则 \mathbb{V}, \mathbb{W} 同构(Isomorphism)， φ 称为同构映射；
- (3) $\varphi : \mathbb{V} \rightarrow \mathbb{V}$ 是线性映射，则 \mathbb{V} 自同态(Endomorphism)， φ 称为自同态映射；
- (4) $\varphi : \mathbb{V} \rightarrow \mathbb{V}$ 是线性映射且是双射，则 \mathbb{V} 自同构(Automorphism)， φ 称为自同构映射。

定义 2.3.6. 设 \mathbb{V}, \mathbb{W} 是数域 \mathbb{K} 上的两个有限维的向量空间，如果 $\varphi : \mathbb{V} \rightarrow \mathbb{W}$ 是一个双射，则可以定义它的逆映射，记作 $\varphi^{-1} : \mathbb{W} \rightarrow \mathbb{V}$ ，对于 $\forall \mathbf{x} \in \mathbb{V}$ 和 $\forall \mathbf{y} \in \mathbb{W}$ 使得

$$\varphi^{-1}(\varphi(\mathbf{x})) = \mathcal{E}(\mathbf{x}) = \mathbf{x}, \varphi(\varphi^{-1}(\mathbf{y})) = \mathcal{E}(\mathbf{y}) = \mathbf{y}$$

例 2.3.5. 根据逆映射的定义,

- 在例 2.3.5 中, 只有 1 例有逆映射;
- 恒等映射的逆映射是其本身;
- 在例 2.3.4 中定义了映射 $T_Q(A) = Q^{-1}AQ$ 的逆映射为 $T_{Q^{-1}}$ 。

例 2.3.6. 映射 $\varphi: \mathbb{R}^2 \rightarrow \mathbb{C}$, $\varphi(x) = x_1 + ix_2$ 是同态映射, 因为

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

定理 2.3.1. 考虑向量空间 $\mathbb{V}, \mathbb{W}, \mathbb{X}$, 则有

- (1) 对于线性映射 $\varphi: \mathbb{V} \rightarrow \mathbb{W}$ 和 $\phi: \mathbb{W} \rightarrow \mathbb{X}$, 则 $\phi(\varphi)$ 也是一个线性映射;
- (2) 对于双射 $\varphi: \mathbb{V} \rightarrow \mathbb{W}$ 和 $\phi: \mathbb{W} \rightarrow \mathbb{X}$, 则 $\phi(\varphi)$ 也是一个双射;
- (3) 如果 $\varphi: \mathbb{V} \rightarrow \mathbb{W}$ 是同构映射, 则 $\varphi^{-1}: \mathbb{W} \rightarrow \mathbb{V}$ 也是一个同构映射;
- (4) 如果 $\varphi: \mathbb{V} \rightarrow \mathbb{W}, \phi: \mathbb{W} \rightarrow \mathbb{X}$ 是线性映射, 且 $\lambda \in \mathbb{R}$, 则 $\varphi + \phi$ 和 $\lambda\varphi$ 也是线性映射。

深度学习中, 常常利用映射的复合构建更为强大、准确率更高的分类器, 比如在 MNIST 数字识别的例子中, 先用 f_1 将图像映射成 50 维的向量, 再用 f_2 将 50 维的向量映射为 10 维的向量, 但是如果我们利用线性映射的复合构造分类器, 即

$$\mathbf{h}_1 = f_1(\mathbf{x}) = \mathbf{A}_1\mathbf{x}$$

$$\mathbf{y}' = f_2(\mathbf{h}_1) = \mathbf{A}_2\mathbf{h}_1$$

设 $\mathbf{A} = \mathbf{A}_2\mathbf{A}_1$,

$$\mathbf{y}' = f_2(f_1(\mathbf{x})) = \mathbf{A}_2\mathbf{A}_1\mathbf{x} = \mathbf{Ax}$$

得到的仍然是一个线性映射, 并不能提高分类的准确性。

定理 2.3.2. 设 \mathbb{V}, \mathbb{W} 是数域 \mathbb{K} 上的两个有限维的向量空间, \mathbb{V}, \mathbb{W} 同构, 当且仅当 $\dim(\mathbb{V}) = \dim(\mathbb{W})$ 。

定理 2.3.2 表明了两个维数相同的向量空间之间存在着一个满足双射的线性映射, 从这个观点看, 同构的向量空间是可以不加区别的, 维数是有限维向量空间的唯一本质特征。

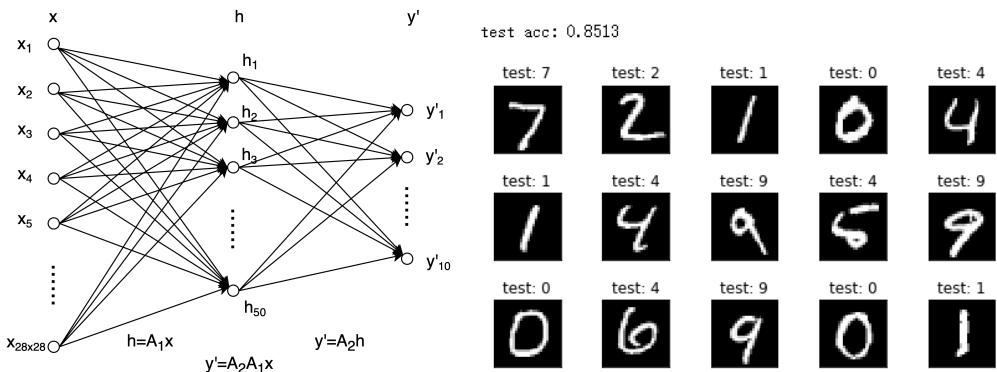


图 2.12: 双层线性网络准确率

2.3.3 线性映射的矩阵表示

由定理2.3.2, 我们知道任何 n 维向量空间都同构于 \mathbb{R}^n 。

定义 2.3.7. 我们现在考虑一个 n 维向量空间 \mathbb{V} 的基底 $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ 。在下面, 基向量的顺序很重要。因此, 我们把 $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ 称这个 n 元组为 \mathbb{V} 的有序基。

回顾坐标的概念, 给定一个向量空间 \mathbb{V} 和 \mathbb{V} 的一个有序基底 $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n)$, 任何的 $\mathbf{x} \in \mathbb{V}$, 我们得到 \mathbf{x} 关于 \mathbf{B} 的唯一表示 (线性组合)

$$\mathbf{x} = \alpha_1 \mathbf{b}_1 + \dots + \alpha_n \mathbf{b}_n$$

然后 $\alpha_1, \dots, \alpha_n$ 是 \mathbf{x} 在 \mathbf{B} 下的坐标。下面的向量

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

是 \mathbf{x} 关于 \mathbf{B} 的坐标表示, 即 $\mathbf{x} = \mathbf{B}\boldsymbol{\alpha}$ 。

例 2.3.7. 考虑一个几何矢量 $x \in \mathbb{R}^2$, 坐标 $[2, 3]^T$, 可以用标准基 $e_1, e_2 \in \mathbb{R}^2$ 来表示。这意味着, 我们可以写 $x = 2e_1 + 3e_2$ 。然而, 我们不必选择标准基来表示这个向量, 如果我们使用基向量 $b_1 = [1, -1]^T, b_2 = [1, 1]^T$, 我们将获得坐标 $[-\frac{1}{2}, \frac{5}{2}]^T$ 来表示同一矢量。

现在, 我们准备在矩阵和线性映射之间建立有限维向量空间之间的联系。

定义 2.3.8. 考虑向量空间 \mathbb{V}, \mathbb{W} 的有序基 $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ 和 $\mathbf{C} = (\mathbf{c}_1, \dots, \mathbf{c}_m)$ 。然后考虑一个线性映射 $\Phi : \mathbb{V} \rightarrow \mathbb{W}$, 对于 $j \in \{1, \dots, n\}$

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

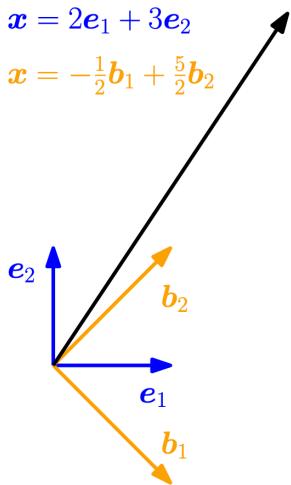


图 2.13: 不同基下的坐标

$\Phi(b_j)$ 是关于 C 的唯一表示。那么，我们称这个 $m \times n$ 的矩阵为 A_Φ ，它的元素为：

$$A_\Phi(i, j) = a_{ij}$$

是 Φ 的变换矩阵。 $\Phi(b_j)$ 在 \mathbb{W} 有序基 C 下的坐标是 A_Φ 的第 j 列。

记 $\Phi(\mathbf{B}) = (\Phi(\mathbf{b}_1), \Phi(\mathbf{b}_2), \dots, \Phi(\mathbf{b}_n))$ ，则

$$\Phi(\mathbf{B}) = CA_\Phi.$$

设向量空间 \mathbb{V}, \mathbb{W} 的有序基分别为 \mathbf{B}, \mathbf{C} ，线性映射 $\Phi : \mathbb{V} \rightarrow \mathbb{W}$ 的变换矩阵 A_Φ ，如果 $\mathbf{x} \in \mathbb{V}$ 关于 \mathbf{B} 的坐标是 $\hat{\mathbf{x}}$ ， $\mathbf{y} = \Phi(\mathbf{x}) \in \mathbb{W}$ 关于 \mathbf{C} 的坐标是 $\hat{\mathbf{y}}$ ：

$$\Phi(\mathbf{x}) = \Phi(\mathbf{B}\hat{\mathbf{x}}) = \Phi(\mathbf{B})\hat{\mathbf{x}} = CA_\Phi\hat{\mathbf{x}} = C(A_\Phi\hat{\mathbf{x}})$$

$A_\Phi\hat{\mathbf{x}}$ 就是 $\Phi(\mathbf{x})$ 关于 \mathbf{C} 的坐标，由此得到坐标的映射关系：

$$\hat{\mathbf{y}} = A_\Phi\hat{\mathbf{x}}$$

这意味着这个变换矩阵可以用来计算在两个空间各自基下坐标的映射关系。

例 2.3.8. 考虑同态 $\Phi : \mathbb{V} \rightarrow \mathbb{W}$ ， \mathbb{V} 有序基 $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_3)$ ， \mathbb{W} 的有序基 $\mathbf{C} = (\mathbf{c}_1, \dots, \mathbf{c}_4)$ ，有：

$$\Phi(\mathbf{b}_1) = \mathbf{c}_1 - \mathbf{c}_2 + 3\mathbf{c}_3 - \mathbf{c}_4$$

$$\Phi(\mathbf{b}_2) = 2\mathbf{c}_1 + \mathbf{c}_2 + 7\mathbf{c}_3 + 2\mathbf{c}_4$$

$$\Phi(\mathbf{b}_3) = 3\mathbf{c}_2 + \mathbf{c}_3 + 4\mathbf{c}_4$$

其变换矩阵为：

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

其中 $\mathbf{a}_j, j = 1, 2, 3$, 是 $\Phi(b_j)$ 关于 \mathbb{W} 的有序基 \mathbf{C} 的坐标。

考虑向量空间 $\mathbb{V}, \mathbb{W}, \mathbb{X}$, 我们知道线性映射的复合仍是线性映射

$$\Phi : \mathbb{V} \rightarrow \mathbb{W}$$

$$\Psi : \mathbb{W} \rightarrow \mathbb{X}$$

$$\Psi \circ \Phi : \mathbb{V} \rightarrow \mathbb{X}$$

记 $\mathbf{A}_\Phi, \mathbf{A}_\Psi$ 是对应的变换矩阵, 则 $\mathbf{A}_{\Psi \circ \Phi} = \mathbf{A}_\Psi \mathbf{A}_\Phi$ 。

例 2.3.9. 令

$$\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n, \Phi(\mathbf{x}) = 3\mathbf{x}, \Psi : \mathbb{R}^n \rightarrow \mathbb{R}, \Psi(\mathbf{x}) = \sum_{i=1}^n x_i$$

则

$$\mathbf{A}_\Phi = 3\mathbf{I}$$

$$\mathbf{A}_\Psi = (1, 1, \dots, 1)$$

$$\text{而 } \Psi \circ \Phi(\mathbf{x}) = \sum_{i=1}^n 3x_i$$

$$\mathbf{A}_{\Psi \circ \Phi} = (3, 3, \dots, 3) = (1, 1, \dots, 1) 3\mathbf{I} = \mathbf{A}_\Psi \mathbf{A}_\Phi$$

接下来, 我们将研究当我们改变 \mathbb{V} 和 \mathbb{W} 的基底时, 一个线性映射 $\Phi : \mathbb{V} \rightarrow \mathbb{W}$ 的变换矩阵如何变化的。

考虑 \mathbb{V} 的两个有序基底:

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

和 \mathbb{W} 的两个有序基底

$$\mathbf{C} = (\mathbf{c}_1, \dots, \mathbf{c}_n), \tilde{\mathbf{C}} = (\tilde{\mathbf{c}}_1, \dots, \tilde{\mathbf{c}}_n)$$

$\mathbf{A}_\Phi \in \mathbb{R}^{m \times n}$ 是线性映射 $\Phi : \mathbb{V} \rightarrow \mathbb{W}$ 的变换矩阵。其中 \mathbb{V} 的基底是 B , \mathbb{W} 的基底是 C 。

$\tilde{\mathbf{A}}_\Phi \in \mathbb{R}^{m \times n}$ 是线性映射 $\Phi : \mathbb{V} \rightarrow \mathbb{W}$ 的变换矩阵。其中 \mathbb{V} 的基底是 $\tilde{\mathbf{B}}$, \mathbb{W} 的基底是 $\tilde{\mathbf{C}}$ 。

例 2.3.10. 考虑基为 $\mathbf{e}_1, \mathbf{e}_2$ 的 \mathbb{R}^2 上的变换矩阵 $\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$, 如果我们将新的基底定义为

$$\mathbf{B} = \left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right)$$

我们可以得到一个对角的变换矩阵

$$\tilde{A} = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}$$

这便简化了 A 。

定理 2.3.3. (基变换) 对于一个线性映射 $\Phi : \mathbb{V} \rightarrow \mathbb{W}$, 设 \mathbb{V} 的两个有序基底:

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

\mathbb{W} 有两个有序基底

$$\mathbf{C} = (\mathbf{c}_1, \dots, \mathbf{c}_n), \tilde{\mathbf{C}} = (\tilde{\mathbf{c}}_1, \dots, \tilde{\mathbf{c}}_n).$$

设 Φ 在基 \mathbf{B} 与 \mathbf{C} 下的变换矩阵为 A_Φ , 在基 $\tilde{\mathbf{B}}$ 与 $\tilde{\mathbf{C}}$ 下的变换矩阵为 \tilde{A}_Φ 。则有

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

其中 $S \in \mathbb{R}^{n \times n}$ 是 \mathbb{V} 中恒等映射的变换矩阵 (从 \mathbf{B} 到 $\tilde{\mathbf{B}}$), $T \in \mathbb{R}^{m \times m}$ 是 \mathbb{W} 中恒等映射的变换矩阵 (从 \mathbf{C} 到 $\tilde{\mathbf{C}}$)。

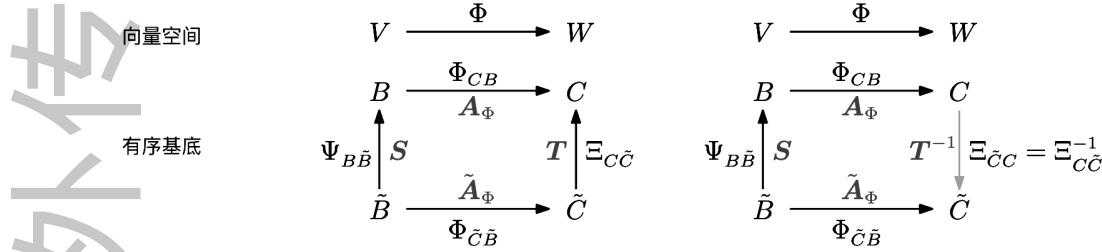


图 2.14: 不同基底下的坐标关系

当我们分别在 \mathbb{V} 中把基底从 B 变换为 \tilde{B} 以及在 \mathbb{W} 中把基底从 C 变换为 \tilde{C} , 我们可以通过一些步骤来得到相应的变换矩阵 \tilde{A}_Φ 。

- 首先, 我们写出联系新基底 \tilde{B} 下坐标和旧基底 B 下坐标的线性映射 $\Psi_{B\tilde{B}} : \mathbb{V} \rightarrow \mathbb{V}$ 所对应的矩阵表示。
- 然后我们再使用 Φ_{CB} 的变换矩阵 A_Φ 将坐标映射到以 C 为基底的 \mathbb{W} 中。
- 最后我们再使用线性映射 $\Xi_{\tilde{C}C} : \mathbb{W} \rightarrow \mathbb{W}$ 把坐标从用基底 C 表示到用基底 \tilde{C} 表示。

因此我们可以将线性映射 $\Phi_{\tilde{C}\tilde{B}}$ 表示为:

$$\Phi_{\tilde{C}\tilde{B}} = \Xi_{\tilde{C}C} \circ \Phi_{CB} \circ \Psi_{B\tilde{B}} = \Xi_{CC}^{-1} \circ \Phi_{CB} \circ \Psi_{B\tilde{B}}$$

定义 2.3.9. 如果对于两个矩阵 $A, B \in \mathbb{R}^{m \times n}$, 存在可逆矩阵 $S \in \mathbb{R}^{n \times n}$, $T \in \mathbb{R}^{m \times m}$ 使得, $A = T^{-1}BS$ 成立, 则称 A, B 等价

定义 2.3.10. 如果对于两个矩阵 $A, B \in \mathbb{R}^{n \times n}$, 存在可逆矩阵 $S \in \mathbb{R}^{n \times n}$ 使得, $A = S^{-1}BS$ 成立。则称 A, B 相似。

所以两个相似的矩阵必定等价, 反之则不然。

例 2.3.11. 考虑一个线性映射 $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^4$, 其在标准基下的变换矩阵为

$$\begin{pmatrix} 1 & 2 & 0 \\ -1 & 1 & 3 \\ 3 & 7 & 1 \\ -1 & 2 & 4 \end{pmatrix}$$

我们寻找一个新的基下的 Φ 的变换矩阵。令新的基分别为

$$\tilde{B} = \left(\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \right), \tilde{B} = \left(\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right).$$

所以

$$S = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, T = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

因此我们可以得到

$$\tilde{A}_\Phi = T^{-1}A_\Phi S = \frac{1}{2} \begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 0 \\ -1 & 1 & 3 \\ 3 & 7 & 1 \\ -1 & 2 & 4 \end{pmatrix} \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} -4 & -4 & -2 \\ 6 & 0 & 0 \\ 4 & 8 & 4 \\ 1 & 6 & 3 \end{pmatrix}$$

线性映射的像与核两个重要的线性子空间。

定义 2.3.11. 对于 $\Phi : \mathbb{V} \rightarrow \mathbb{W}$ 我们定义核空间(零空间):

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

像空间(值域):

$$Im(\Phi) := \Phi(\mathbb{V}) = \{\mathbf{w} \in \mathbb{W} | \exists \mathbf{v} \in \mathbb{V} : \Phi(\mathbf{v}) = \mathbf{w}\}$$

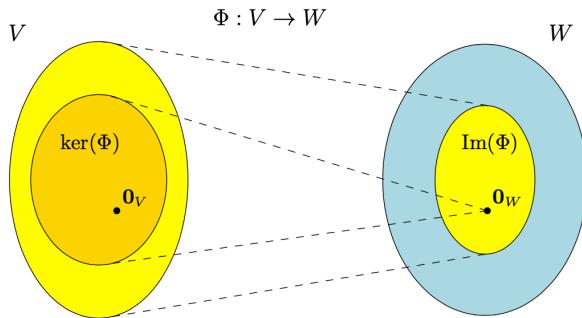


图 2.15: 核空间与像空间

接下来给出一些关于像空间与核空间的结论。

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

- 总有 $\Phi(\mathbf{0}_{\mathbb{V}}) = \mathbf{0}_{\mathbb{W}}$, 因此 $\mathbf{0}_{\mathbb{V}} \in \ker(\Phi)$;
也就是说零空间永远非空。
- $\text{Im}(\Phi) \subseteq \mathbb{W}$ 是 \mathbb{W} 的子空间;
- $\ker(\Phi) \subseteq \mathbb{V}$ 是 \mathbb{V} 的子空间;
- Φ 是单射当且仅当 $\ker(\Phi) = \mathbf{0}$;
- $\text{rank}(\mathbf{A}) = \dim(\text{Im}(\Phi))$;
- Φ 的核空间是方程 $A\mathbf{x} = \mathbf{0}$ 的解空间。

定义 2.3.12. \mathbf{A} 的列向量张成空间叫做列空间。

考虑 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 和线性映射 $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m, \mathbf{x} \rightarrow A\mathbf{x}$ 。

对于 $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n]$, 我们可以得到

$$\text{Im}(\Phi) = \{A\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\} = \left\{\sum_{i=1}^n x_i \mathbf{a}_i\right\} = L(\mathbf{a}_1, \dots, \mathbf{a}_n) \subseteq \mathbb{R}^m$$

所以 Φ 的像空间是可以由 \mathbf{A} 的列向量张成的。

例 2.3.12. 考虑映射

$$\Phi : \mathbb{R}^4 \rightarrow \mathbb{R}^2, \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & -1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} x_1 + 2x_2 - x_3 \\ x_1 + x_4 \end{pmatrix}$$

是线性映射。

Φ 的像空间就是变换矩阵的列空间。

$$Im(\Phi) = L\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right)$$

为了得到零空间我们需要解 $\mathbf{Ax} = \mathbf{0}$ 。

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

最终我们可以给出

$$ker(\Phi) = L\left(\begin{pmatrix} 0 \\ 1/2 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 1/2 \\ 0 \\ 1 \end{pmatrix}\right)$$

2.3.4 线性变换

线性变换的定义

在这一节中，我们主要讨论向量空间 \mathbb{V} 到自身的映射，称为 \mathbb{V} 的一个变换。下面如果不作声明，所考虑的都是数域 \mathbb{K} 上的向量空间。

定义 2.3.13. 设 \mathbb{V} 是数域 \mathbb{K} 上的向量空间，如果对任何向量 $\mathbf{x}, \mathbf{y} \in \mathbb{V}$ 及任意的 $\alpha, \beta \in \mathbb{K}$ ，有

$$\mathcal{A}(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha\mathcal{A}(\mathbf{x}) + \beta\mathcal{A}(\mathbf{y}),$$

则称 \mathcal{A} 为 \mathbb{V} 上的线性变换， $\mathcal{A}(\mathbf{x})$ 和 $\mathcal{A}(\mathbf{y})$ 代表元素 \mathbf{x} 和 \mathbf{y} 在变换 \mathcal{A} 下的像。

例 2.3.13. 下列线性映射是线性变换：

- 恒等映射 $\epsilon : \mathbb{V} \rightarrow \mathbb{V}$, $\epsilon(\mathbf{x}) = \mathbf{x}$ 。
- 例 2.3.4 中的线性映射 $T_Q(\mathbf{A}) = Q^{-1}\mathbf{A}Q$, 其中 Q 是可逆矩阵。我们称其为矩阵的相似变换。

设 \mathbb{V} 是数域 \mathbb{K} 上的 n 维向量空间, $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 是 \mathbb{V} 的一组基，现在我们来建立线性变换与矩阵之间的关系。

空间 \mathbb{V} 中任一向量 ξ 可以被基 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 线性表出，即有

$$\mathbf{a} = a_1\varepsilon_1 + a_2\varepsilon_2 + \cdots + a_n\varepsilon_n$$

其中系数是唯一确定的，它们就是 ξ 在这组基下的坐标。由于线性变换保持线性关系不变，因而在 ξ 的像 $\mathcal{A}\xi$ 与基的像 $\mathcal{A}\varepsilon_1, \mathcal{A}\varepsilon_2, \dots, \mathcal{A}\varepsilon_n$ 之间也存在：

$$\begin{aligned} \mathcal{A}\xi &= \mathcal{A}(x_1\varepsilon_1 + x_2\varepsilon_2 + \cdots + x_n\varepsilon_n) \\ &= x_1\mathcal{A}(\varepsilon_1) + x_2\mathcal{A}(\varepsilon_2) + \cdots + x_n\mathcal{A}(\varepsilon_n). \end{aligned} \tag{2.3}$$

上式表明，如果我们就知道了基 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 的像，那么向量空间中任意一个向量 ξ 的像也就知道了，或者说

定理 2.3.4. 设 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 是向量空间 \mathbb{V} 的一组基, 如果线性变换 A 与 B 在这组基上的作用相同, 即

$$\mathcal{A}\varepsilon_i = \mathcal{B}\varepsilon_i, \quad i = 1, 2, \dots, n,$$

那么 $A = B$.

证明. \mathcal{A} 与 \mathcal{B} 相等的意义是它们对每个向量的作用相同。因此，我们就是要证明对任一向量 a ，等式 $\mathcal{A}\xi = \mathcal{B}\xi$ 成立，由(2.3)得，

$$\begin{aligned}\mathcal{A}\xi &= x_1\mathcal{A}(\varepsilon_1) + x_2\mathcal{A}(\varepsilon_2) + \cdots + x_n\mathcal{A}(\varepsilon_n) \\ &= x_1\mathcal{B}(\varepsilon_1) + x_2\mathcal{B}(\varepsilon_2) + \cdots + x_n\mathcal{B}(\varepsilon_n) = \mathcal{B}\xi.\end{aligned}$$

□

定理2.3.4指出，一个线性变换完全被它在一组基上的作用所决定，然而，基向量的像却完全可以是任意的，也就是说

定理 2.3.5. 设 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 是向量空间 \mathbb{V} 的一组基, 对于任意一组向量 a_1, a_2, \dots, a_n , 一定存在一个线性变换 A 使得

$$A\varepsilon_i \equiv g_i, \quad i = 1, 2, \dots, n. \quad (2.4)$$

证明 我们来作出所要的线性变换，设

$$\xi = \sum_{i=1}^n x_i \varepsilon_i$$

是向量空间 V 的任意一个向量，我们定义 V 的变换 A 为

$$\mathcal{A}\xi = \sum_{i=1}^n x_i a_i$$

下面来证明变换 A 是线性的。

在 V 中任取两个向量,

$$\mathbf{b} = \sum_{i=1}^n b_i \varepsilon_i, \mathbf{c} = \sum_{i=1}^n c_i \varepsilon_i.$$

于是

$$\mathbf{b} + \mathbf{c} = \sum_{i=1}^n (b_i + c_i) \varepsilon_i,$$

$$k\mathbf{b} = \sum_{i=1}^n kb_i \varepsilon_i, k \in \mathbb{K}$$

按照所定义的 \mathcal{A} 的表达式, 有

$$\begin{aligned}\mathcal{A}(\mathbf{b} + \mathbf{c}) &= \sum_{i=1}^n (b_i + c_i) \mathbf{a}_i, \\ &= \sum_{i=1}^n b_i \mathbf{a}_i + \sum_{i=1}^n c_i \mathbf{a}_i = \mathcal{A}\mathbf{b} + \mathcal{A}\mathbf{c}, \\ \mathcal{A}(k\mathbf{b}) &= \sum_{i=1}^n kb_i \mathbf{a}_i = k \sum_{i=1}^n b_i \mathbf{a}_i = k\mathcal{A}\mathbf{b}.\end{aligned}$$

因此, \mathcal{A} 是线性变换。再来证 \mathcal{A} 满足 (2.4)。因为

$$\varepsilon_i = 0\varepsilon_1 + \cdots + 0\varepsilon_{i-1} + 1\varepsilon_i + 0\varepsilon_{i+1} + \cdots + 0\varepsilon_n, i = 1, 2, \dots, n,$$

所以

$$\mathcal{A}\varepsilon_i = 0\mathbf{a}_1 + \cdots + 0\mathbf{a}_{i-1} + 1\mathbf{a}_i + 0\mathbf{a}_{i+1} + \cdots + 0\mathbf{a}_n = \mathbf{a}_i, i = 1, 2, \dots, n.$$

□

结合以上两点, 则有

定理 2.3.6. 设 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 是向量空间 \mathbb{V} 的一组基, $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ 是 \mathbb{V} 中任意 n 个向量, 存在唯一的线性变换 \mathcal{A} 使得

$$\mathcal{A}\varepsilon_i = \mathbf{a}_i, \quad i = 1, 2, \dots, n.$$

线性变换与矩阵

有了以上的讨论, 就可以建立线性变换与矩阵之间的关系。

定义 2.3.14. 设 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 是数域 \mathbb{K} 上 n 维向量空间 \mathbb{V} 的一组基, \mathcal{A} 是 \mathbb{V} 的一个线性变换, 基向量的像可以被基线性表出:

$$\left\{ \begin{array}{l} \mathcal{A}\varepsilon_1 = a_{11}\varepsilon_1 + a_{21}\varepsilon_2 + \cdots + a_{n1}\varepsilon_n, \\ \mathcal{A}\varepsilon_2 = a_{12}\varepsilon_1 + a_{22}\varepsilon_2 + \cdots + a_{n2}\varepsilon_n, \\ \cdots \cdots \\ \mathcal{A}\varepsilon_n = a_{1n}\varepsilon_1 + a_{2n}\varepsilon_2 + \cdots + a_{nn}\varepsilon_n. \end{array} \right.$$

用矩阵来表示就是

$$\begin{aligned}\mathcal{A}(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n) &= (\mathcal{A}\varepsilon_1, \mathcal{A}\varepsilon_2, \dots, \mathcal{A}\varepsilon_n), \\ &= (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)\mathbf{A}\end{aligned}$$

其中

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}.$$

矩阵 A 称为 \mathcal{A} 在基 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 下的矩阵。

由线性变换的矩阵可以直接计算一个向量的像。

定理 2.3.7. 设线性变换 \mathcal{A} 在基 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 下的矩阵是 A , 且向量 ξ 在基 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 下的坐标为 (x_1, x_2, \dots, x_n) , 则 $\mathcal{A}\xi$ 在基 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 下的坐标 (y_1, y_2, \dots, y_n) 可以按照公式

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = A \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

计算。

证明. 由假设

$$\xi = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

于是

$$\begin{aligned} \mathcal{A}\xi &= (\mathcal{A}\varepsilon_1, \mathcal{A}\varepsilon_2, \dots, \mathcal{A}\varepsilon_n) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \\ &= (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n) A \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}. \end{aligned}$$

另一方面, 由假设

$$\mathcal{A}\xi = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n) \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

由于 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 线性无关, 所以

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = A \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}.$$

草稿清外切

□

伸缩与旋转

我们先来看一个例子：

例 2.3.14. 考虑线性变换 A 在数域 \mathbb{R}^2 上的三组矩阵

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

它们对原数据的改变如图所示。

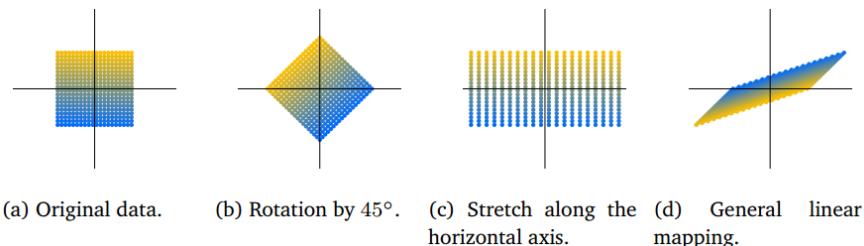


图 2.16: 线性变换对原数据的影响

如果使用矩阵 A_1 , 将会对原数据进行 45 度旋转, 而使用矩阵 A_2 将会对横坐标进行 2 倍拉伸, 矩阵 A_3 融合了旋转和拉伸操作。

2.3.5 仿射映射

与线性空间之间的映射相似, 我们可以定义两个仿射空间的映射。

定义 2.3.15. 设两个线性空间 \mathbb{V}, \mathbb{W} 与一个线性映射 $\Phi : \mathbb{V} \rightarrow \mathbb{W}$ 则映射 $\phi : \mathbb{V} \rightarrow \mathbb{W}$ 且 $\mathbf{a} \in \mathbb{W}$

$$\phi(\mathbf{x}) = \mathbf{a} + \Phi(\mathbf{x})$$

是一个仿射映射, 又称仿射变换。

例 2.3.15. 函数 $f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$ 就是一个仿射函数。设仿射空间

$$\mathbb{V} = \mathbf{c} + L(\mathbf{d})$$

则 f 将这个仿射空间的函数映射到了仿射空间

$$\mathbb{W} = \mathbf{A}\mathbf{c} + \mathbf{b} + L(\mathbf{Ad})$$

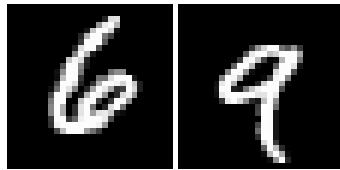


图 2.17: MNIST 数据集中的数字图片

例 2.3.16. 考虑如何将图 2.17 旋转一个角度。

以原图像的点 $s = (s_1, s_2)$ 为旋转中心, 经过旋转后, 该旋转中心在旋转后的目标图像位置为 $d = (d_1, d_2)$ 。逆时针旋转角度为 δ 。则一个像素的原位置 $x = (x_1, x_2)$ 与旋转后的目标位置 $y = (y_1, y_2)$ 的关系为:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos \delta & -\sin \delta \\ \sin \delta & \cos \delta \end{pmatrix} \begin{pmatrix} x_1 - s_1 \\ x_2 - s_2 \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

通过原像素位置计算目标操作位置进行旋转操作的方式称为前向变换。

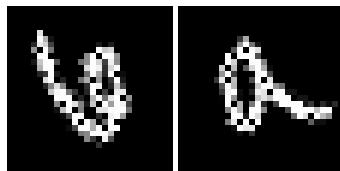


图 2.18: 旋转后的结果

测试一下我们用前向变换方式得到的逆时针旋转 60° 的图像。以原图像的点 $s = (s_1, s_2)$ 为旋转中心, 经过旋转后, 该旋转中心在旋转后的目标图像位置为 $d = (d_1, d_2)$ 。旋转角度为 δ 。一个像素的原位置 $x = (x_1, x_2)$ 与旋转后的目标位置 $y = (y_1, y_2)$ 的关系为:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos \delta & -\sin \delta \\ \sin \delta & \cos \delta \end{pmatrix} \begin{pmatrix} x_1 - s_1 \\ x_2 - s_2 \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

利用矩阵的逆, 得到变化后的目标像素 $y = (y_1, y_2)$ 的原坐标 $x = (x_1, x_2)$ 为:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \cos \delta & \sin \delta \\ -\sin \delta & \cos \delta \end{pmatrix} \begin{pmatrix} y_1 - d_1 \\ y_2 - d_2 \end{pmatrix} + \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$$

因此, 只需要对目标图像进行遍历, 依次填充原图像相应坐标点的像素, 即可得到旋转后的图像。这种方式称为反向变换。

在 MNIST 数字分类例子中, 即使我们将线性映射改为仿射映射, 重新构造模型并使用优化算法求解, 最终得到的模型的准确率也几乎不能提升。同样, 由于多个仿射函数的复合函数仍然是仿射函数, 也不能提高模型的预测能力。

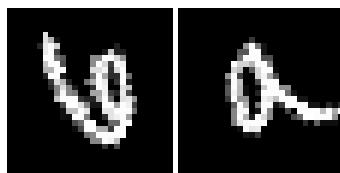


图 2.19: 旋转后的结果

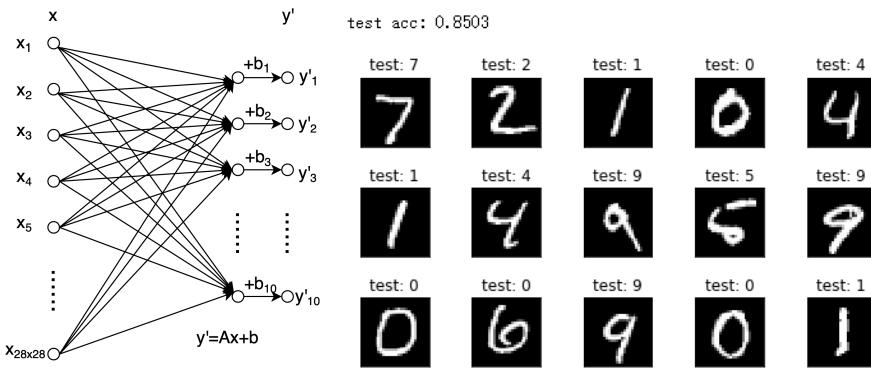


图 2.20: 仿射映射模型分类准确率

注记 9. 本节我们主要讨论了线性映射、线性变换及其矩阵表示和仿射映射等，它们分别是机器学习领域线性模型的基础，在机器学习领域通常把由仿射映射或线性映射与一些简单函数复合得到的映射统称为线性模型。通常线性模型用于数据量较少的任务建模，其泛化能力是有限的。另外，由本节神经网络的例子我们可以看出，仅仅利用单层神经网络中线性映射的复合，最后对数据处理效果的提升几乎没有，粗略的说，这可能是由于线性映射并不能捕捉到数据中的非线性特征，因此需要引入非线性映射或非线性函数，对于深度神经网络来说，也就是激活函数，将起到非常重要的作用。除此之外，数据分析和机器学习领域的任务建模还会遭遇很多其他的非线性函数或非线性模型，比如基于行列式或二次型的任务建模等。关于这类非线性函数模型我们将在 3.4 节做详细介绍。

下一节我们将首先从映射或函数的角度来讨论矩阵的一些基本特征，包括行列式、迹和二次型等。

2.4 矩阵的基本特征

在进一步讨论数据分析和机器学习建模用的非线性函数之前，本节我们先讨论行列式，迹和特征值及其相关的二次型和特征向量等，我们把这些线性代数中反映矩阵特征的一些量称为

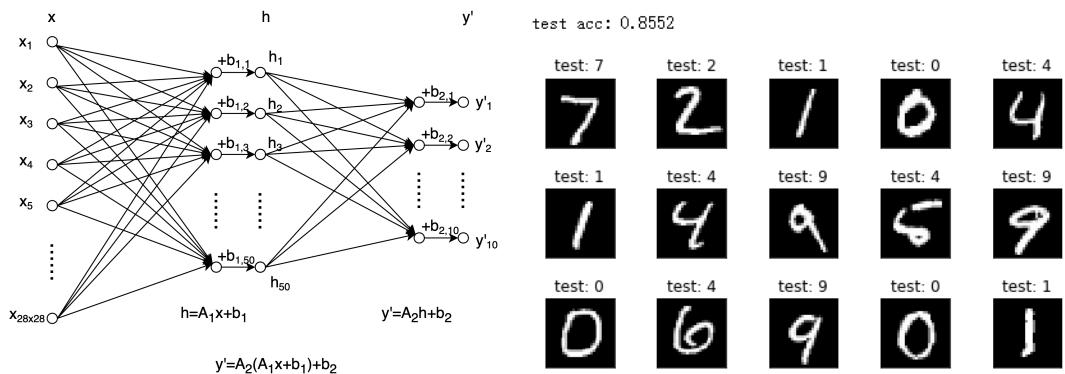


图 2.21: 仿射映射复合模型分类准确率

矩阵的基本特征。从数学的角度看，它们反映了矩阵的数值性状和几何性状。从数据科学的角度看，数据矩阵的特征值和特征向量常常反映数据内部特征关系。接下来，我们首先来看看行列式的概念。

2.4.1 行列式

行列式是关于矩阵的一个函数，将 n 维矩阵向量空间中一个 $n \times n$ 的矩阵 A 映射到一个标量，记作 $\det(A)$ 或 $|A|$ 。无论是在线性代数、多项式理论，还是在微积分学中（比如说换元积分法中），行列式作为基本的数学工具，都有着重要的应用。

行列式概念最早出现在解线性方程组的过程中。解方程是代数中一个非常基本的问题。对于二元线性方程组

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1, \\ a_{21}x_1 + a_{22}x_2 = b_2, \end{cases}$$

当 $a_{11}a_{22} - a_{12}a_{21} \neq 0$ 时，此方程组有唯一解，即

$$x_1 = \frac{b_1a_{22} - a_{12}b_2}{a_{11}a_{22} - a_{12}a_{21}}, x_2 = \frac{a_{11}b_2 - a_{12}b_1}{a_{11}a_{22} - a_{12}a_{21}}.$$

我们称 $a_{11}a_{22} - a_{12}a_{21}$ 为二阶行列式，用符号表示为

$$a_{11}a_{22} - a_{12}a_{21} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}.$$

于是上述解可以用二阶行列式叙述为：当二阶行列式

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \neq 0$$

时，该方程组有唯一解。

我们将这个结果推广到 n 元线性方程组，首先就要给出 n 级行列式的定义及性质，这就是本节的主要内容。

定义 2.4.1. n 个不同的元素排成一列，叫做这 n 个元素的全排列。一个排列中，如果一个大元素在小元素前，则称这两个数构成一个逆序。一个排列中存在的所有逆序的数目称为排列的逆序数。排列 j_1, j_2, \dots, j_n 的逆序数记为 $\tau(j_1, j_2, \dots, j_n)$ 。如果逆序数为奇数，称这个排列为奇排列；如果逆序数为偶数，称这个排列为偶排列。

例 2.4.1. • $\tau(3, 2, 1, 4) = 3$, $3, 2, 1, 4$ 为奇排列

- $\tau(1, 3, 2, 4) = 1$, $1, 3, 2, 4$ 为奇排列
- $\tau(3, 1, 2, 4) = 2$, $3, 1, 2, 4$ 为偶排列

定义 2.4.2. $\det(A)$ 叫做矩阵 A 的行列式，是从 $\mathcal{R}^{n \times n}$ 映射到 \mathcal{R} 的一个函数，其中 $A \in \mathcal{R}^{n \times n}$ 。

$$\det(A) = \sum_{j_1, j_2, \dots, j_n} (-1)^{r(j_1, j_2, \dots, j_n)} a_{1j_1} a_{2j_2} \cdots a_{nj_n}$$

其中 $r(j_1, j_2, \dots, j_n)$ 表示排列 (j_1, j_2, \dots, j_n) 的逆序数，即满足 $1 \leq i_1 \leq i_2 \leq n$ 且 $j_{i_1} > j_{i_2}$ 的有序数对 (i_1, i_2) 的个数。

行列式的性质

行列式具有如下性质：

- (1) 行列互换（转置），行列式的值不变；
- (2) 行列式中一行的公因子可以提出去；
- (3) 如果行列式中有一行的元素全为零，则这个行列式的值等于零；
- (4) 如果行列式中有某一行是两组数的和，则这个行列式等于这两个行列式的和，这两个行列式的一行分别是第一组数与第二组数，其余各行与原行列式的相应各行相同；
- (5) 对换行列式中的两行，行列式反号；
- (6) 如果行列式中有两行相同或成比例，则这个行列式的值等于零；
- (7) 把行列式的某一行的倍数加到另一行上去，行列式的值不变。

行列式的计算

(1) 利用定义计算

(2) 化行列式为上（下）三角形行列式

定义 2.4.3. 主对角线（从左上角到右下角的对角线）下（上）方的元素全为零的行列式称为下三角形行列式（上三角形行列式）。主对角线以外的元素全为零的行列式称为对角形行列式。

定理 2.4.1. 上(下)三角形行列式的值等于主对角线上元素的乘积, 即

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & a_{nn} \end{vmatrix} = a_{11}a_{22}\cdots a_{nn},$$

$$\begin{vmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} = a_{11}a_{22}\cdots a_{nn}.$$

特别地, 对角形行列式的值等于它的主对角线上的元素的乘积。

计算行列式的基本方法常利用行列式的性质, 把行列式化为上三角形的行列式, 再根据定理2.4.1计算。

(3) 行列式按一行(列)展开

定义 2.4.4. 在 n 阶行列式中, 划去元素 a_{ij} 所在的第 i 行与第 j 列, 剩下的元素按原来次序组成的 $n-1$ 阶行列式称为元素 a_{ij} 的余子式, 记作 M_{ij} 。令 $A_{ij}=(-1)^{i+j}M_{ij}$, 称为 A_{ij} 为元素 a_{ij} 的代数余子式。

定理 2.4.2. 设 $D=|a_{ij}|$, 以 A_{ij} 表示元素 a_{ij} 的代数余子式, 则下列公式成立 $k=i$:

$$\sum_{s=1}^n a_{ks}A_{ij} = \begin{cases} D & \text{当 } k=i \\ 0 & \text{当 } k \neq i \end{cases}. \quad (2.5)$$

$$\sum_{s=1}^n a_{sl}A_{sj} = \begin{cases} D & \text{当 } l=j \\ 0 & \text{当 } l \neq j \end{cases}. \quad (2.6)$$

公式(2.5)和(2.6)表明: 行列式等于它的任意一行(列)的元素与此元素的代数余子式的乘积之和; 行列式中任意一行(列)的元素与另外一行(列)的相应元素的代数余子式的乘积之和等于零。计算行列式的另一种基本方法是利用行列式的性质, 使其一行(列)变成只有少数几个非零元素, 然后再按这一行(列)展开。

例 2.4.2. 行列式

$$\begin{vmatrix} 5 & 3 & -1 & 2 & 0 \\ 1 & 7 & 2 & 5 & 2 \\ 0 & -2 & 3 & 1 & 0 \\ 0 & -4 & -1 & 4 & 0 \\ 0 & 2 & 3 & 5 & 0 \end{vmatrix} = (-1)^{(2+5)}2 \begin{vmatrix} 5 & 3 & -1 & 2 \\ 0 & -2 & 3 & 1 \\ 0 & -4 & -1 & 4 \\ 0 & 2 & 3 & 5 \end{vmatrix}$$

$$\begin{aligned}
 &= -2 \times 5 \begin{vmatrix} -2 & 3 & 1 \\ -4 & -1 & 4 \\ 2 & 3 & 5 \end{vmatrix} = -10 \begin{vmatrix} -2 & 3 & 1 \\ 0 & -7 & 2 \\ 0 & 6 & 6 \end{vmatrix} \\
 &= (-10) \times (-2) \begin{vmatrix} -7 & 2 \\ 6 & 6 \end{vmatrix} = 20 \times (-42 - 12) = -1080
 \end{aligned}$$

有了行列式的计算方法，我们可以给出另外一个解线性方程组的方法。

定理 2.4.3. 设线性方程组为

$$Ax = b$$

，我们记 b 为常数列， $|A|_j$ 为用常数列 b 代替 A 中的第 j 列，其余列不变所得矩阵的行列式。

则若 $|A| \neq 0$ ，则线性方程组有唯一解，且

$$x_1 = \frac{|A|_1}{|A|}, x_2 = \frac{|A|_2}{|A|}, \dots, x_n = \frac{|A|_n}{|A|}$$

这一结论，我们称为克莱姆法则。

行列式的几何意义

概括来说，行列式有两种解释，第一种解释为行列式是行列式中的行或列向量所构成的超平行多面体的有向面积或有向体积；另一种解释为矩阵 A 的行列式 $\det(A)$ 就是线性变换 A 下图形面积或体积的伸缩因子。

例如，一个 2×2 矩阵 $A = \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix}$ 的行列式是平面直角坐标系 xoy 平面上以行向量 $a = (a_1, a_2)$, $b = (b_1, b_2)$ 为邻边的平行四边形的有向面积：若这个平行四边形是由向量 a 沿逆时针方向转到 b 而得到的，面积取正值；若这个平行四边形是由向量 a 沿顺时针方向转到 b 而得到的，面积取负值，如图2.22所示。

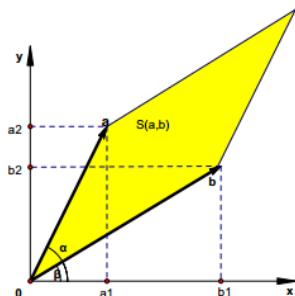


图 2.22: 二阶行列式的几何意义

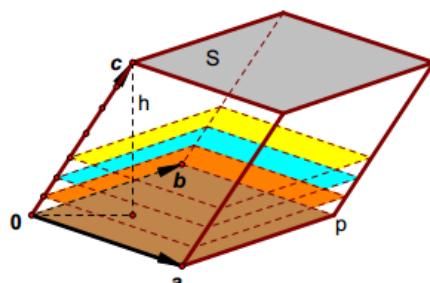


图 2.23: 三阶行列式的几何意义

类似地，三阶行列式的值就是它的三个向量在 $Oxyz$ 空间上张成的平行六面体的有向体积。例如图2.23，我们给定起点相同的三个向量 $\mathbf{a}, \mathbf{b}, \mathbf{c}$ 并以其作为平行六面体的三条边，则可以确定一个平行六面体。设图中

$$\mathbf{a} = \begin{pmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{pmatrix}, \mathbf{c} = \begin{pmatrix} 0 \\ \frac{1}{4} \\ 1 \end{pmatrix}$$

那么这个平行六面体的体积为

$$\det([\mathbf{a}, \mathbf{b}, \mathbf{c}]) = \begin{vmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{1}{4} \\ 0 & 0 & 1 \end{vmatrix} = 1$$

我们可以很容易的验证这个结论。因为这个平行六面体的底面 \mathbf{a}, \mathbf{b} 张成的平行四边形面积为 1，相应的高 h 为 1。

而关于伸缩因子的几何解释，假设 \mathbf{A} 是一个行向量（或列向量）为 \mathbf{a}, \mathbf{b} 的 2×2 的矩阵。那么，这里的线性变换 \mathbf{A} 是指将 \mathbb{R}^2 中的单位正方形变成 \mathbb{R}^2 中以 \mathbf{a}, \mathbf{b} 为邻边的平行四边形；如果原图像是一个圆，那么线性变换 \mathbf{A} 则将之变成一个椭圆。

同样地，在 3 维的情形下（如2.23）， \mathbf{A} 将 \mathbb{R}^3 中的一个单位立方体映射成 \mathbb{R}^3 中由 \mathbf{A} 的行向量确定的平行六面体；如果原图形是一个球，则线性变换 \mathbf{A} 将之变成一个椭球。

一般地，一个 $n \times n$ 矩阵 \mathbf{A} 将 \mathbb{R}^n 中单位 n 立方体变成 \mathbb{R}^n 中 \mathbf{A} 行向量确定的 n 维平行体。对非单位正方形（立方体或超立方体）以同样的方式变换，即伸缩因子为像域的容积/原域的容积。而 $n \times n$ 矩阵 \mathbf{A} 的行列式 $\det(\mathbf{A})$ 就是这个伸缩因子。

下面这个特殊的矩阵反映了矩阵的行列式与矩阵的逆之间的关系。

定义 2.4.5. 矩阵

$$\mathbf{A}^* = \begin{pmatrix} A_{11} & A_{21} & \cdots & A_{n1} \\ A_{12} & A_{22} & \cdots & A_{n2} \\ \vdots & \vdots & & \vdots \\ A_{1n} & A_{2n} & \cdots & A_{nn} \end{pmatrix}$$

称为 \mathbf{A} 的伴随矩阵。

注意：伴随矩阵 \mathbf{A}^* 的第 i 行第 j 列元素是矩阵的第 j 行第 i 列元素的代数余子式。

由定理2.4.3和伴随矩阵的定义：

定理 2.4.4.

$$\mathbf{AA}^* = \mathbf{A}^*\mathbf{A} = \begin{pmatrix} |\mathbf{A}| & 0 & \cdots & 0 \\ 0 & |\mathbf{A}| & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & |\mathbf{A}|\end{pmatrix} = |\mathbf{A}|\mathbf{I}$$

如果 A 可逆，则

$$A^* = |A|A^{-1}, \quad A^{-1} = \frac{1}{|A|}A^*$$

2.4.2 迹运算

迹也是关于矩阵的一个函数，将 n 维矩阵向量空间中一个 $n \times n$ 的矩阵 A 映射到一个标量，记作 $\text{Tr}(A)$ 。

定义 2.4.6. 矩阵 $A = (a_{ij})$ 对角元素的和

$$\text{Tr}(A) = \sum_i a_{ii}$$

称为矩阵 A 的迹。

定理 2.4.5. 矩阵的迹运算有以下这些性质：

- (1) $\text{Tr}(A) = \text{Tr}(A^T) \quad A \in \mathbb{R}^{n \times n}$
- (2) $\text{Tr}(A + B) = \text{Tr}(A) + \text{Tr}(B) \quad A, B \in \mathbb{R}^{n \times n}$
- (3) $\text{Tr}(AB) = \text{Tr}(BA) \quad A \in \mathbb{R}^{n \times m}, B \in \mathbb{R}^{m \times n}$
- (4) $\text{Tr}(ABC) = \text{Tr}(CAB) = \text{Tr}(BCA) \quad A \in \mathbb{R}^{n_1 \times n_2}, B \in \mathbb{R}^{n_2 \times n_3}, C \in \mathbb{R}^{n_3 \times n_1}$
- (5) $\text{Tr}(G^{-1}AG) = \text{Tr}(A) \quad A, G \in \mathbb{R}^{n \times n}, G \text{ 可逆}$

证明. (1)

$$\text{Tr}(A) = \sum_i^n a_{ii} = \text{Tr}(A^T)$$

(2)

$$\text{Tr}(A + B) = \sum_i^n (a_{ii} + b_{ii}) = \sum_i^n a_{ii} + \sum_i^n b_{ii} = \text{Tr}(A) + \text{Tr}(B)$$

(3)

$$\text{Tr}(AB) = \sum_{i=0}^n \sum_{j=0}^m a_{ij}b_{ji} = \sum_{j=0}^m \sum_{i=0}^n b_{ji}a_{ij} = \text{Tr}(BA)$$

(4)

$$\text{Tr}(ABC) = \text{Tr}((AB)C) = \text{Tr}(CAB) = \text{Tr}(BCA)$$

(5)

$$\text{Tr}(G^{-1}AG) = \text{Tr}(AGG^{-1}) = \text{Tr}(A)$$

□

对于多个矩阵的连乘积，只要其运算结果是一个方阵，我们有更一般的结论：

性质 2.4.1. 迹的循环置换不变性, 即

$$\mathrm{Tr}(A_1 A_2 \cdots A_n) = \mathrm{Tr}(A_n A_1 \cdots A_{n-1}) = \cdots = \mathrm{Tr}(A_2 A_3 \cdots A_1) \quad (2.7)$$

例 2.4.3. 设矩阵 A 和 B 分别为

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, B = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}$$

则

$$AB = \begin{pmatrix} 6 & 6 \\ 15 & 15 \end{pmatrix}, BA = \begin{pmatrix} 5 & 7 & 9 \\ 5 & 7 & 9 \\ 5 & 7 & 9 \end{pmatrix}$$

则有 $\mathrm{Tr}(AB) = \mathrm{Tr}(BA) = 21$ 。

推论 2.4.1. 相似矩阵的迹是相等的, 因为

$$\mathrm{Tr}(Q^{-1}AQ) = \mathrm{Tr}(QQ^{-1}A) = \mathrm{Tr}(A)$$

2.4.3 对称矩阵与二次型

二次型及其矩阵表示

在解析几何中, 我们看到, 当坐标原点与中心重合时, 一个有心二次曲线的一般方程是

$$ax^2 + 2bxy + cy^2 = f. \quad (2.8)$$

为了便于研究这个二次曲线的几何性质, 我们可以选择适当的角度 θ , 作转轴 (反时针方向转轴)

$$\begin{cases} x = x' \cos \theta - y' \sin \theta, \\ y = x' \sin \theta + y' \cos \theta, \end{cases} \quad (2.9)$$

把方程(2.8)化成标准方程。在二次曲线的研究中也有类似的情况。

(2.8) 的左端是一个二次齐次多项式。从代数的观点看, 所谓化标准方程就是用变量的线性替换(2.9)化简一个二次齐次多项式, 使它只含有平方项。本节就是来介绍它的一些最基本的性质。

定义 2.4.7. 一个系数在数域 \mathbb{K} 上的 x_1, x_2, \dots, x_n 的二次齐次多项式

$$\begin{aligned} f(x_1, x_2, \dots, x_n) = & a_{11}x_1^2 + 2a_{12}x_1x_2 + \cdots + 2a_{1n}x_1x_n \\ & + a_{22}x_2^2 + 2a_{23}x_2x_3 + \cdots + 2a_{2n}x_2x_n \\ & + \cdots \cdots + a_{nn}x_n^2 \end{aligned} \quad (2.10)$$

称为数域 \mathbb{K} 上的 n 元二次型，简称二次型，当 \mathbb{K} 为 \mathbb{R} 或 \mathbb{C} 时，分别称为实二次型或复二次型。对称矩阵(*symmetric*)是转置和自己相等的矩阵，即

$$\mathbf{A} = \mathbf{A}^T$$

二次型(2.10)的系数排成的对称矩阵

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

称为所给二次型的矩阵，其中 $a_{ij} = a_{ji}, i, j = 1, 2, \dots, n$ ，若令 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ ，则所给的二次型可表示为：

$$f(x_1, x_2, \dots, x_n) = \mathbf{x}^T \mathbf{A} \mathbf{x}$$

二次型的矩阵的秩也称为二次型的秩。

与在几何中一样，在处理许多其它问题时也常常希望通过变量的线性替换来简化有关的二次型，因此我们引入

定义 2.4.8. 设 $x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_n$ 是两组文字，系数在数域 \mathbb{K} 中的一组关系式

$$x_i = \sum_{j=1}^n c_{ij} y_j \quad (i = 1, 2, \dots, n) \quad (2.11)$$

称为由 x_1, x_2, \dots, x_n 到 y_1, y_2, \dots, y_n 的一个线性替换，或简称线性替换。若 $\det(c_{ij}) \neq 0$ ，则称线性替换(2.10)为非退化的线性替换。

如果把方程(2.9)看作线性替换，那么它就是非退化的，因为

$$\begin{vmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{vmatrix} = 1 \neq 0.$$

不难看出，如果把(2.11)代入(2.10)，那么得到的 y_1, \dots, y_n 的多项式仍然是二次齐次的，换句话说，线性替换把二次型变成二次型。

我们知道，经过一个非退化的线性替换，二次型还是变成二次型。那么，替换后的二次型与原来的二次型之间有什么关系？也就是说，我们需要找出替换后的二次型的矩阵与原二次型的矩阵之间的关系。

设

$$f(x_1, x_2, \dots, x_n) = \mathbf{x}^T \mathbf{A} \mathbf{x}, \mathbf{A} = \mathbf{A}^T$$

是一个二次型，作非退化线性替换

$$\mathbf{x} = \mathbf{C} \mathbf{y},$$

我们得到一个 y_1, y_2, \dots, y_n 的二次型

$$\mathbf{y}^T \mathbf{B} \mathbf{y}.$$

则有

$$f(x_1, x_2, \dots, x_n) = \mathbf{x}^T \mathbf{A} \mathbf{x} = (\mathbf{C} \mathbf{y})^T \mathbf{A} (\mathbf{C} \mathbf{y}) = \mathbf{y}^T \mathbf{C}^T \mathbf{A} \mathbf{C} \mathbf{y} = \mathbf{y}^T (\mathbf{C}^T \mathbf{A} \mathbf{C}) \mathbf{y} = \mathbf{y}^T \mathbf{B} \mathbf{y}.$$

即有

$$\mathbf{B} = \mathbf{C}^T \mathbf{A} \mathbf{C}.$$

这就是前后两个二次型的矩阵的关系，与之对应，我们引入

定义 2.4.9. 设 \mathbf{A}, \mathbf{B} 都是 \mathbb{K} 上的 $n \times n$ 矩阵，若存在 \mathbb{K} 上的可逆的 $n \times n$ 矩阵 \mathbf{C} ，使得 $\mathbf{B} = \mathbf{C}^T \mathbf{A} \mathbf{C}$ ，则称 \mathbf{A} 与 \mathbf{B} 是合同矩阵，记作 $\mathbf{A} \simeq \mathbf{B}$ 。

合同是矩阵之间的一个关系，不难看出，合同关系具有

(1) 反身性： $\mathbf{A} = \mathbf{I}^T \mathbf{A} \mathbf{I}$ ；

(2) 对称性：由 $\mathbf{B} = \mathbf{C}^T \mathbf{A} \mathbf{C}$ 即得 $\mathbf{A} = (\mathbf{C}^{-1})^T \mathbf{B} \mathbf{C}^{-1}$ ；

(3) 传递性：由 $\mathbf{A}_1 = \mathbf{C}_1^T \mathbf{A} \mathbf{C}_1$ 和 $\mathbf{A}_2 = \mathbf{C}_2^T \mathbf{A}_1 \mathbf{C}_2$ 即得

$$\mathbf{A}_2 = (\mathbf{C}_1 \mathbf{C}_2)^T (\mathbf{C}_1 \mathbf{C}_2).$$

因此我们有经过一非退化的线性替换，二次型仍变成二次型，且新二次型的矩阵与原二次型的矩阵是合同的。

标准型

定理 2.4.6. 数域 \mathbb{K} 上任意一个二次型都可经过非退化的线性替换化为平方和

$$d_1 x_1^2 + d_2 x_2^2 + \cdots + d_n x_n^2$$

的形式，它称为所给二次型的标准形。

定理2.4.6也可等价地叙述为如下的定理2.4.7。

定理 2.4.7. 数域 \mathbb{K} 上任意一个对称矩阵都合同于一个对角矩阵，即对于任意一个对称矩阵 \mathbf{A} 都可以找到一可逆矩阵 \mathbf{C} ，使得

$$\mathbf{C}^T \mathbf{A} \mathbf{C}$$

成对角矩阵。

用初等变换法可以将二次型化为标准形。

设二次型 $f = f(x_1, x_2, \dots, x_n)$ 的矩阵为 \mathbf{A} ，作初等变换

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{I} \end{pmatrix} \xrightarrow{\begin{array}{l} \text{对 } \mathbf{A} \text{ 作成对的初等行、列变换} \\ \text{对 } \mathbf{I} \text{ 只作其中的初等列变换} \end{array}} \begin{pmatrix} \mathbf{D} \\ \mathbf{C} \end{pmatrix}$$

其中 \mathbf{D} 是对角矩阵 $\mathbf{D} = [d_1, d_2, \dots, d_n]$ ， \mathbf{C} 是非退化的线性替换矩阵，此时， $f = d_1 y_1^2 + d_2 y_2^2 + \cdots + d_n y_n^2$ 。

例 2.4.4. 用初等变换法化二次型 $f(x_1, x_2, x_3) = x_1x_2 + x_1x_3 - 3x_2x_3$ 为标准形。

解. $f(x_1, x_2, x_3)$ 的矩阵为

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 0 & -3/2 \\ 1/2 & -3/2 & 0 \end{pmatrix}, \\ \begin{pmatrix} \mathbf{A} \\ \mathbf{I} \end{pmatrix} &= \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 0 & -3/2 \\ 1/2 & -3/2 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1/4 & 0 \\ 0 & 0 & 3 \\ 1 & -1/2 & 3 \\ 1 & 1/2 & -1 \\ 0 & 0 & 1 \end{pmatrix}. \\ \mathbf{D} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1/4 & 0 \\ 0 & 0 & 3 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 1 & -1/2 & 3 \\ 1 & 1/2 & -1 \\ 0 & 0 & 1 \end{pmatrix}, \end{aligned}$$

线性替换为

$$\begin{cases} x_1 = y_1 - \frac{1}{2}y_2 + 3y_3, \\ x_2 = y_1 + \frac{1}{2}y_2 - y_3, \\ x_3 = y_3 \end{cases}$$

由此得 $f(x_1, x_2, x_3) = y_1^2 - \frac{1}{4}y_2^2 + 3y_3^2$

二次型的惯性指数

定理 2.4.8. 在二次型的标准形中, 系数不为零的平方项的个数是唯一确定的, 与所作的非退化的线性替换无关。

定义 2.4.10. 设 $f(x_1, x_2, \dots, x_n)$ 是一实二次型, 其矩阵的秩为 r , 且标准形为

$$d_1y_1^2 + d_2y_2^2 + \cdots + d_py_p^2 - d_{p+1}y_{p+1}^2 - \cdots - d_r y_r^2 \quad (2.12)$$

其中 $d_i > 0 (i = 1, 2, \dots, r)$, 若再作一线性替换

$$\begin{aligned} y_i &= \frac{1}{\sqrt{d_i}}z_i (i = 1, 2, \dots, r), \\ y_i &= z_j (j = r+1, r+2, \dots, n), \end{aligned}$$

则 (2.12) 式就变成

$$z_1^2 + z_2^2 + \cdots + z_p^2 - z_{p+1}^2 - \cdots - z_r^2 \quad (2.13)$$

(2.13) 式称为实二次型 $f(x_1, x_2, \dots, x_n)$ 的规范形。

若 $f(x_1, x_2, \dots, x_n)$ 是一复二次型，其矩阵的秩为 r ，则其规范形为

$$z_1^2 + z_2^2 + \dots + z_r^2 (r \text{ 为二次型的秩})$$

定理 2.4.9. 任一复(实)系数的二次型，经过一适当的非退化的线性替换总可以化为规范形，且规范形是唯一的。

定理 2.4.9 换个说法就是，任意复数的对称矩阵合同与一个形式为

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

的对角矩阵，从而有，两个复数对称矩阵合同的充分必要条件是它们的秩相等。而实系数的二次型，其规范形完全被 r, p 这两个数所决定。

定义 2.4.11. 在实二次型 $f(x_1, x_2, \dots, x_n)$ 的规范形中，正与负平方项的个数 p 与 $r - p$ 分别称为 $f(x_1, x_2, \dots, x_n)$ 的正惯性指数与负惯性指数，正、负惯性指数之差 $p - (r - p) = 2p - r$ 称为 $f(x_1, x_2, \dots, x_n)$ 符号差。

正(负)定二次型

定义 2.4.12. 设 $f(x_1, x_2, \dots, x_n) = \mathbf{x}^T A \mathbf{x}$ 为 n 元实二次型，若对任一组不全为零的实数 c_1, c_2, \dots, c_n 都有

(1) $f(c_1, c_2, \dots, c_n) > 0 (< 0)$ ，则称 $f(x_1, x_2, \dots, x_n)$ 为正定二次型(负定二次型)，此时称 A 为正定矩阵(负定矩阵)。

(2) $f(c_1, c_2, \dots, c_n) \geq 0 (\leq 0)$ ，则称 $f(x_1, x_2, \dots, x_n)$ 为正半定二次型(负半定二次型)，此时称 A 为正半定矩阵(负半定矩阵)。正定二次型(负定二次型)必是正半定二次型(负半定二次型)。

(3) $f(x_1, x_2, \dots, x_n)$ 既不是正半定的，又不是负半定的，则称 $f(x_1, x_2, \dots, x_n)$ 为不定二次型。

定义 2.4.13. 设 $A = (a_{ij})_{n \times n}$ ，则称 $k (k \leq n)$ 阶子式

$$P_k = \begin{vmatrix} a_{i_1 i_1} & a_{i_1 i_2} & \cdots & a_{i_1 i_k} \\ a_{i_2 i_1} & a_{i_2 i_2} & \cdots & a_{i_2 i_k} \\ \cdots & \cdots & \cdots & \cdots \\ a_{i_k i_1} & a_{i_k i_2} & \cdots & a_{i_k i_k} \end{vmatrix}$$

称 A 的 k 阶主子式，其中 $1 \leq i_1 \leq i_2 < \dots < i_k \leq n$ ；而 $k(k \leq n)$ 阶子式

$$Q_k = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{vmatrix}$$

称为 A 的 k 阶顺序主子式。

定理 2.4.10. 对于实二次型 $f(x_1, \dots, x_n) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ ，其中 \mathbf{A} 是实对称的，那么下列条件等价：

- (1) $f(x_1, \dots, x_n)$ 是正（负）定的；
- (2) 它的正（负）惯性指数与 \mathbf{A} 的秩相等；
- (3) 它的规范形为

$$z_1^2 + z_2^2 + \cdots + z_n^2 \quad (-z_1^2 - z_2^2 - \cdots - z_n^2)$$

(4) A 的所有顺序主子式全大于零 (A 的奇数阶顺序主子式全小于零，偶数阶顺序主子式全大于零)。

例 2.4.5. 判别二次型

$$f(x_1, x_2, x_3) = 5x_1^2 + x_2^2 + 5x_3^2 + 4x_1x_2 - 8x_1x_3 - 4x_2x_3$$

是否正定。

解 $f(x_1, x_2, x_3)$ 的矩阵为

$$\begin{pmatrix} 5 & 2 & -4 \\ 2 & 1 & -2 \\ -4 & -2 & 5 \end{pmatrix}$$

它的顺序主子式

$$5 > 0, \begin{vmatrix} 5 & 2 \\ 2 & 1 \end{vmatrix} > 0, \begin{vmatrix} 5 & 2 & -4 \\ 2 & 1 & -2 \\ -4 & -2 & 5 \end{vmatrix} > 0,$$

因此， $f(x_1, x_2, x_3)$ 正定。

2.4.4 特征值与特征向量

定义 2.4.14. 对于一个 $n \times n$ 矩阵 A ，如果存在数 λ 和向量 \mathbf{x} 使得

$$A\mathbf{x}_0 = \lambda_0 \mathbf{x}_0 \tag{2.14}$$

则称 λ_0 是矩阵 A 的一个特征值，而 \mathbf{x}_0 是 λ_0 对应的矩阵 A 的一个特征向量。

我们容易知道矩阵 A 的特征值就是变元 λ 的 n 次多项式 $\det((\lambda I - A))$ 的 n 个根，所以特征值也称特征根。而 λ_0 对应的特征向量 x_0 就是齐次线性方程组 $(\lambda_0 I - A)x = \mathbf{0}$ 的非零解向量。因此进一步引入术语如下。

定义 2.4.15. 矩阵 $\lambda I - A$ 称为 A 的特征矩阵。

多项式 $\Delta_A(\lambda) = \det(\lambda I - A)$ 称为 A 的特征多项式。

所有特征值的集合 $\lambda(A)$ 称为 A 的谱。

对于 $\lambda_0 \in \lambda(A)$ ，线性方程组 $(\lambda_0 I - A)x = \mathbf{0}$ 的非零解子空间称为 A 的属于特征根 λ_0 的特征子空间，记做 $E_{\lambda_0}(A)$ ，其中的非零向量就是特征向量。

这些概念统称为矩阵 A 的特征系。

在数据科学中，我们一般只讨论实矩阵的特征值问题。

应注意，实矩阵的特征值和特征向量不一定是实数和实向量，但实特征值一定对应于实特征向量（方程(2.14)的解），而一般的复特征值对应的特征向量一定不是实向量。此外，由于特征方程为实系数方程，若一个特征值不是实数，则其复共轭也一定是它的特征值。

对于一个实对称矩阵来说，它的 n 个特征值均为实数，并且存在 n 个正交的实特征向量。

例 2.4.6. 根据定义求矩阵

$$A = \begin{pmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{pmatrix}$$

的特征值和特征向量。

矩阵 A 的特征方程为

$$\det(\lambda I - A) = \begin{vmatrix} \lambda - 1 & -2 & -2 \\ -2 & \lambda - 1 & -2 \\ -2 & -2 & \lambda - 1 \end{vmatrix} = (\lambda + 1)^2(\lambda - 5) = 0,$$

故 A 的特征值为 $\lambda_1 = \lambda_2 = -1$ （二重特征值）， $\lambda_3 = 5$ 。

对 $\lambda_1 = \lambda_2 = -1$ ，由 $(\lambda I - A)x = \mathbf{0}$ ，得到方程

$$\begin{pmatrix} -2 & -2 & -2 \\ -2 & -2 & -2 \\ -2 & -2 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

它有无穷多个解。

- 设 $x_2 = 1, x_3 = 0$ ，求出解为 $x = [-1, 1, 0]^T$ ，记为 x_1 ，

- 设 $x_2 = 0, x_3 = 1$ ，求出解为 $x = [-1, 0, 1]^T$ ，记为 x_2 ，

- 则 x_1 和 x_2 是属于特征值 -1 的两个线性无关的特征向量，属于 -1 的全部特征向量为

$$k_1 x_1 + k_2 x_2, k_1, k_2 \in \mathbb{R}.$$

同理， $\lambda_3 = 5$ 的一个特征向量为 $x_3 = [1, 1, 1]^T$ ，属于 5 的全部特征向量为 $kx_3, k \in \mathbb{R}$ 。

特征值与特征向量的性质

下面概括地介绍有关矩阵特征值、特征向量的一些性质。

定理 2.4.11. 设 $\lambda_j (j = 1, 2, \dots, n)$ 为 n 阶矩阵 A 的特征值，则

$$(1) \sum_{j=1}^n \lambda_j = \sum_{j=1}^n a_{jj} = \text{Tr}(A);$$

$$(2) \prod_{j=1}^n \lambda_j = \det(A);$$

(3) 矩阵转置不改变特征值，即 $\lambda(A) = \lambda(A^T)$ ；

(4) 若矩阵 A 为对角阵或上（下）三角阵，则其对角线元素即为矩阵的特征值；

(5) 若矩阵 A 为分块对角阵，或分块上（下）三角阵，例如

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ & A_{22} & \cdots & A_{2m} \\ & & \ddots & \vdots \\ & & & A_{mm} \end{pmatrix},$$

其中每个对角块 A_{jj} 均为方阵，则矩阵 A 的特征值为各对角阵块矩阵特征值的合并，即 $\lambda(A) = \bigcup_{j=1}^m \lambda(A_{jj})$ 。

(6) 矩阵 cA (c 为常数) 的特征值为 $c\lambda_1, c\lambda_2, \dots, c\lambda_n$ 。

(7) 矩阵 $A + cI$ (c 为常数) 的特征值为 $\lambda_1 + c, \lambda_2 + c, \dots, \lambda_n + c$ 。

(8) 矩阵 A^k (k 为正整数) 的特征值为 $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$ 。

(9) 设 $p(t)$ 为一多项式函数，则矩阵 $p(A)$ 的特征值为 $p(\lambda_1), p(\lambda_2), \dots, p(\lambda_n)$ 。

(10) 若 A 为非奇异矩阵，则 $\lambda_j \neq 0 (j = 1, 2, \dots, n)$ ，且矩阵 A^{-1} 的特征值为 $\lambda_1^{-1}, \dots, \lambda_n^{-1}$ 。

从上述结论(2)也可以看出，非奇异矩阵特征值均不为 0，而 0 一定是奇异矩阵的特征值。

定理 2.4.12. 矩阵的相似变换 (*similarity transformation*) 不改变特征值。设矩阵 A 和 B 为相似矩阵，即存在非奇异矩阵 X 使得 $B = X^{-1}AX$ ，则

(1) 矩阵 A 和 B 的特征值相等，即 $\lambda(A) = \lambda(B)$ ；

(2) 若 y 为 B 的特征向量，则相应地， Xy 为 A 的特征向量。

通过相似变换并不总能把矩阵转化为对角阵，或者说矩阵 A 并不总是可对角化的 (*diagonalizable*)。为了说明矩阵 A 何时可以对角化，下面给出特征值的代数重数、几何重数和亏损矩阵的概念，以及几个定理。

定义 2.4.16. 设矩阵 $A \in \mathbb{R}^{n \times n}$ 有 m 个 ($m \leq n$) 不同的特征值为 $\tilde{\lambda}_1, \dots, \tilde{\lambda}_m$ ，若 $\tilde{\lambda}_j$ 是特征方程的 n_j 重根，则称 n_j 为 $\tilde{\lambda}_j$ 的代数重数 (*algebraic multiplicity*)，并称 $\tilde{\lambda}_j$ 对应的特征子空间 (\mathbb{C}^n 的子空间) 的维数为 $\tilde{\lambda}_j$ 的几何重数 (*geometric multiplicity*)。

定理 2.4.13. 设矩阵 $A \in \mathbb{R}^{n \times n}$ 的 m 个不同的特征值为 $\tilde{\lambda}_1, \dots, \tilde{\lambda}_m$, 特征值 $\tilde{\lambda}_j (j = 1, \dots, m)$ 的代数重数为 n_j , 几何重数为 k_j , 则

(1) $\sum_{j=1}^m n_j = n$, 且任一个特征值的几何重数不大于代数重数, 即 $\forall j, n_j \geq k_j$;

(2) 不同特征值的特征向量线性无关, 并且将所有特征子空间的 $\sum_{j=1}^m k_j$ 个基 (特征向量) 放在一起, 它们构成一组线性无关向量;

(3) 若每个特征值的代数重数等于几何重数, 则总共可得 n 个线性无关的特征向量, 它们使全空间 \mathbb{C}^n 的基。

下面给出一个简单的几何重数和代数重数相等的例子。

例 2.4.7. 设矩阵 A

$$A = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 5 & -2 \\ -2 & 4 & -1 \end{pmatrix}$$

的代数重数和几何重数

解.

$$\det(\lambda I - A) = -(\lambda - 1)^2(\lambda - 3) = 0,$$

故 A 的特征值为 $\lambda_1 = \lambda_2 = 1$ (二重特征值), $\lambda_3 = 3$ 。

对于 $\lambda_1 = \lambda_2 = 1$, 有

$$\lambda I - A = \begin{pmatrix} 0 & 0 & 0 \\ 2 & -4 & 2 \\ 2 & -4 & 2 \end{pmatrix},$$

由于 λ_1 是二重根, 则 λ_1 的代数重数为 2; 而由于 $\lambda I - A$ 的秩为 1, 则 λ_1 的几何重数为 $3 - 1 = 2$ 。

定义 2.4.17. 若矩阵 $A \in \mathbb{R}^{n \times n}$ 的某个代数重数为 k 的特征值对应的线性无关特征向量数目少于 k (即几何重数小于代数重数), 则称 A 为亏损阵 (*defective matrix*), 否则称其为非亏损阵 (*nondefective matrix*)。

如果一个矩阵是非亏损阵, 那么他就可以通过相似变换来对角化。关于亏损矩阵和非亏损矩阵的相关计算, 包括矩阵分解和特征值计算及其在数据科学中的应用, 我们在第 5 章会详细介绍。

注记 10. 本节我们讨论了矩阵的基本特征, 包括行列式、迹和特征值以及相关的二次型和特征向量等。从行列式的定义可以看出, 它是 n 维矩阵空间到实数集合关于矩阵 A 的一个非线性函数。由迹的定义我们可以看出, 它是关于矩阵 A 的一个线性函数。而对于二次型, 如果自变量是矩阵 A , 那么它也是一个关于矩阵 A 的线性函数; 如果把向量 x 看作自变量, 则它是一

个非线性函数。这几个函数在数据分析和机器学习任务建模中具有重要的应用，很多问题最后都归结为基于行列式、迹和二次型的模型。特别是二次型，当把向量 x 看作自变量时，它是一个二次函数，与优化模型中二次规划有着紧密的联系，我们将在第 11 章优化问题部分给予详细的介绍。

2.5 阅读材料

本章我们以文本的向量表示和图像的矩阵表示作为切入，围绕实现数据分析与机器学习具体任务所需的数据表示、数据建模，系统的回顾了线性代数中的一些基本概念，如向量和矩阵的定义及运算，向量空间，线性映射与线性变换，矩阵的基本特征，包括行列式、迹和特征值以及二次型和特征向量等，这些概念和理论构成了本章数学内容的逻辑主线，相应的数据表述部分构成了我们隐藏的数据主线的一部分。这些数学概念在本书的后续章节、数据科学、机器学习与人工智能领域都有重要应用。特别是向量空间和线性映射的引入，为我们建立更复杂的满足数据处理的代数结构和度量空间以及与非线性函数复合产生更复杂的机器学习模型奠定了基础。这些内容将在第三章给予详细介绍。关于线性代数部分数学内容更详细的介绍，可以参考国内外优秀的教科书：[Axler, 2015], [Boyd and Vandenberghe, 2018], [Strang, 1988] [Giuseppe and Laurent, 2014], [Stoer and Burlirsch, 2002], [Deisenroth, Faisal and Ong 2019] 以及 [张贤达, 2004] 等。关于机器学习和数据分析内容的更详细介绍可以参考 [Hastie, Tibshirani and Friedman, 2016], [Bishop, 2006], [Duda, Hart and Stork, 2012], [Goodfellow, Bengio and Courville, 2017], [Scholkopf and Smola, 2002], [Scholkopf, Smola and Muller, 1997] 以及 [周志华, 2016] 等。

习题

习题 2.1. 假设向量 β 可以经向量组 $\alpha_1, \alpha_2, \dots, \alpha_r$ 线性表出，证明：表示法是唯一的充分必要条件是 $\alpha_1, \alpha_2, \dots, \alpha_r$ 线性无关。

习题 2.2. 设 $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3$ ，求方程式 $A\mathbf{x} = 12\mathbf{x}$ 所有的解，其中：

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

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

习题 2.3. 求出下列非齐次线性方程 $Ax=b$ 中所有解的集合 S ，其中 A 和 b 定义如下：

(1)

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

(2)

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

习题 2.4. 设 $A = \begin{pmatrix} 3 & 1 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 1 & -1 \\ 2 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

计算 $AB, AB - BA$ **习题 2.5.** 计算:

$$(1) \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^n \quad (2) \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix}^n \quad (3) \begin{pmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{pmatrix}^n$$

习题 2.6. 求 A^{-1} , 设:

$$(1) A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (2) A = \begin{pmatrix} 2 & 2 & 3 \\ 1 & -1 & 0 \\ -1 & 2 & 1 \end{pmatrix}$$

习题 2.7. 证明 $\alpha_1, \alpha_2, \dots, \alpha_r$ (其中 $\alpha_1 \neq 0$) 线性相关的充分必要条件是至少有一 α_i ($1 < i \leq s$) 可被 $\alpha_1, \alpha_2, \dots, \alpha_{i-1}$ 线性表出。**习题 2.8.** 设

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

将向量 $y = \begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}$ 表示成 x_1, x_2, x_3 的线性组合。

习题 2.9. 判断下列向量是否线性无关。

(1)

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

(2)

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

习题 2.10. 把向量 β 表成向量 $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ 的线性组合:

- (1) $\beta = (1, 2, 1, 1)$, $\alpha_1 = (1, 1, 1, 1)$, $\alpha_2 = (1, 1, -1, -1)$, $\alpha_3 = (1, -1, 1, -1)$, $\alpha_4 = (1, -1, -1, 1)$;
- (2) $\beta = (0, 0, 0, 1)$, $\alpha_1 = (1, 1, 0, 1)$, $\alpha_2 = (2, 1, 3, 1)$, $\alpha_3 = (1, 1, 0, 1)$, $\alpha_4 = (0, 1, -1, -1)$;

习题 2.11. 设 $\alpha_1 = (1, -1, 2, 4)$, $\alpha_2 = (0, 3, 1, 2)$, $\alpha_3 = (3, 0, 7, 14)$, $\alpha_4 = (1, -1, 2, 0)$, $\alpha_5 = (2, 1, 5, 6)$.

(1) 证明: α_1, α_2 线性无关;

(2) 把 α_1, α_2 扩充成一极大线性无关组。

习题 2.12. 计算下列矩阵的秩:

$$(1) \begin{pmatrix} 0 & 1 & 1 & -1 & 2 \\ 0 & 2 & -2 & -2 & 0 \\ 0 & -1 & -1 & 1 & 1 \\ 1 & 1 & 0 & 1 & -1 \end{pmatrix}, \quad (2) \begin{pmatrix} 1 & -1 & 2 & 1 & 0 \\ 2 & -2 & 4 & -2 & 0 \\ 3 & 0 & 6 & -1 & 1 \\ 0 & 3 & 0 & 0 & 1 \end{pmatrix}, \quad (3) \begin{pmatrix} 14 & 12 & 6 & 8 & 2 \\ 6 & 104 & 21 & 9 & 17 \\ 7 & 6 & 3 & 4 & 1 \\ 35 & 30 & 15 & 20 & 5 \end{pmatrix}$$

习题 2.13. 判断下列映射是否是线性映射。

(1) $a, b \in \mathbb{R}$

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

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

其中 $L^1([a, b])$ 表示 $[a, b]$ 上的可积函数集。

(2)

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

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

其中 $k \geq 1, C^k$ 表示连续可微的 k 次的集合, C^0 表示连续函数集。

(3)

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

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

(4)

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

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

(5) $\theta \in [0, 2\pi]$.

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

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

习题 2.14. 已知 E 是一个向量空间, 令 f 和 g 是 E 上的自同态映射, 且 $f \circ g = \text{id}_E$ 。证明 $f = \ker(g \circ f)$ $\text{Im } g = \text{Im}(g \circ f)$ 和 $\ker(f) \cap \text{Im}(g) = \{\mathbf{0}_E\}$ 。

习题 2.15. 对于 $\Phi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ 的变换矩阵是

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

(1) 求 $\ker(\Phi), \text{Im}(\Phi)$ 。

(2) 确定关于基 B 的变换矩阵 \tilde{A}_Φ 。

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

习题 2.16. 已知 \mathbb{R}^3 标准基下向量 c_1, c_2, c_3

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

令 $C = (c_1, c_2, c_3)$ 。

(1) 证明 C 是 \mathbb{R}^3 的基。

(2) $C' = (c'_1, c'_2, c'_3)$ 是 \mathbb{R}^3 的标准基。计算从 C' 到 C 的过渡矩阵 P_2 。

习题 2.17. 考虑 \mathbb{R}^2 中的四个向量 b_1, b_2, b'_1, b'_2 。令 $B = (b_1, b_2)$ 并且 $B' = (b'_1, b'_2)$

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

求 B' 到 B 的过渡矩阵。

习题 2.18. 判断如下的两个矩阵的正定性:

$$\mathbf{A}_1 = \begin{pmatrix} 9 & 6 \\ 6 & 5 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} 9 & 6 \\ 6 & 3 \end{pmatrix}$$

习题 2.19. 证明 $\mathbf{x}^T \mathbf{A} \mathbf{x} = \text{Tr}(\mathbf{x}^T \mathbf{A} \mathbf{x})$ 和 $\mathbf{x}^T \mathbf{A} \mathbf{x} = \text{Tr}(\mathbf{A} \mathbf{x} \mathbf{x}^T)$

习题 2.20. t 取什么值时, 下列二次型是正定的:

$$(1) \quad x_1^2 + x_2^2 + 5x_3^2 + 2tx_1x_2 - 2x_1x_3 + 4x_2x_3$$

$$(2) \quad x_1^2 + 4x_2^2 + x_3^2 + 2tx_1x_2 + 10x_1x_3 + 6x_2x_3$$

$$\text{习题 2.21. 设 } \mathbf{A} = \begin{pmatrix} 1 & 4 & 2 \\ 0 & -3 & 4 \\ 0 & 4 & 3 \end{pmatrix} \text{ 求 } \mathbf{A}^k$$

习题 2.22. 证明: 如果 \mathbf{A} 可逆, 证明: \mathbf{AB} 与 \mathbf{BA} 相似

习题 2.23. 设一个线性映射 $f: R^n \rightarrow R^m$, 如何计算(唯一)矩阵 \mathbf{A} , 对每一个 $\mathbf{x} \in R^n$ 都使 $f(\mathbf{x}) = \mathbf{Ax}$ 成立, 可以自己确定 f 在适当向量处的值表示。

习题 2.24. 已知线性映射

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

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

(1) 计算 \mathbf{A}_Φ

(2) 计算 $\text{rank}(\mathbf{A}_\Phi)$

(3) 计算 Φ 的核与像。核的维数 $\dim(\ker(\Phi))$ 和像的维数 $\dim(\text{Im}(\Phi))$ 是多少?

习题 2.25. 证明: 在 \mathbb{R}^n 上, 当且仅当对称矩阵 \mathbf{A} 是正定矩阵时, 函数 $f(\mathbf{x}) = (\mathbf{x}^T \mathbf{A} \mathbf{x})^{\frac{1}{2}}$ 是一个向量范数。

习题 2.26. 令 $\mathbf{A} \in \mathbb{R}^{n \times n}$, $p(\lambda) \doteq \det(\lambda \mathbf{I}_n - \mathbf{A}) = \lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1\lambda + c_0$ 是 \mathbf{A} 的特征多项式。

(1) 假设 \mathbf{A} 是可对角化的。证明:

$$p(\mathbf{A}) = \mathbf{A}^n + c_{n-1}\mathbf{A}^{n-1} + \cdots + c_1\mathbf{A} + c_0\mathbf{I}_n = 0$$

(2) 证明: 在一般情况下 $p(\mathbf{A}) = 0$ 是成立的, 即对于不可对角方阵也是成立的。

习题 2.27. 斐波那契数列前两项为 1, 自第三项起为之前两项之和。 a_i 表示斐波那契数列的第 i 项。记向量

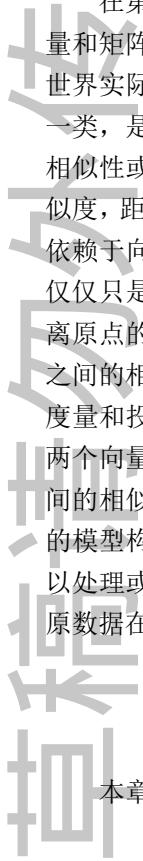
$$\alpha_i = \begin{bmatrix} a_i \\ a_{i+1} \end{bmatrix} \quad i = 1, 2, \dots$$

设 A 为 2×2 常量矩阵使得 $\alpha_{i+1} = A\alpha_i$:

- (1) 写出矩阵 A
- (2) 计算 A^n 并给出 a_n 的通项公式。

参考文献

- [1] Strang, G. 2006. Linear Algebra and Its Application, 4th. Brooks Cole.
- [2] Axler, S. 2015. Linear Algebra Done Right. third edn. Springer.
- [3] Boyd, S. and Vandenberghe, L. 2018. Introduction to Applied Linear Algebra. Cambridge University Press.
- [4] Giuseppe, C. and Laurent, E.G. 2014. Optimization Models. Cambridge University Press.
- [5] Stoer, J. and Burlirsch, R. 2002. Introduction to Numerical Analysis. Springer.
- [6] 张贤达. 矩阵分析与应用 [M]. 清华大学出版社, 2004.
- [7] Hastie, T., Tibshirani, R. and Friedman, J. 2016. The Elements of Statistical Learning. 2nd. Springer.
- [8] Deisenroth, M. P., Faisal, A. A., Ong, C. S. 2019. Mathematics for machine learning.
- [9] Bishop, C. M. 2006. Pattern recognition and machine learning. Springer.
- [10] Duda, R. O., Hart, P. E. and Stork, D. G. 2012. Pattern classification. John Wiley & Sons.
- [11] Goodfellow, I., Bengio, Y. and Courville, A. 深度学习 [M]. 人民邮电出版社, 2017.
- [12] 周志华. 机器学习 [M]. 清华大学出版社, 2016.
- [13] Scholkopf, B. and Smola, A. J. 2002. Learning with Kernels—Support Vector Machines, Regularization, Optimization, and Beyond. Adaptive Computation and Machine Learning. Cambridge, MA, USA: The MIT Press.
- [14] Scholkopf, B., Smola, A. J. and Muller, Klaus-Robert. 1997. Kernel principal component analysis. Pages 583-588 of: International Conference on Artificial Neural Networks. Springer.



第三章 度量与投影

在第 2 章我们已经从数据科学的角度对向量和矩阵基础做了简要的介绍，讨论了数据的向量和矩阵表示，它们统称为数据的低维结构表示。有了表示之后，为了获得数据所表示的现实世界实际对象的信息以及知识，比如说我们希望知道两段文本或两幅图像的类别信息，属于哪一类，是不是同一类，那么我们该怎么办呢？我们可以通过判断文本或者图像数据向量之间的相似性或者相关性来实现这一目标，最简单的方法就是用两向量之间的距离或者角度来表示相似度，距离越小或者角度越小，相似度越大。但是如果我们把文本或者图像放在向量空间中，只依赖于向量的加法和数乘运算似乎是不能实现这一目的。因为此时，表示文本或者图像的向量仅仅只是空间中的一个点，而且只能知道他们在空间中的位置，但是他们之间的远近关系以及离原点的距离，也就是说向量本身的长度、角度等几何特征并不清楚，也就无法刻画这些向量之间的相关性或相似性。为此我们需要在向量空间或者线性空间上引入向量之间新的数学结构：度量和投影，也就是范数和内积结构，用来刻画向量空间的几何特征，包括向量的长度，以及两个向量之间的距离和角度等。而内积和相应的范数以及距离或角度度量可以用来描述数据之间的相似性，这种相似性可以用于数据分析和机器学习中实现数据类别判断的分类和聚类方法的模型构建，比如支持向量机模型的构建等。另一方面，在数据科学中，常常将高维空间中难以处理或难以展示的数据投影到低维空间。我们自然需要思考，经过投影变化后得到的数据和原数据在某些关注的性质上有多大差异，为计算这种差异也需要我们利用度量。

本章的内容概览图如下：

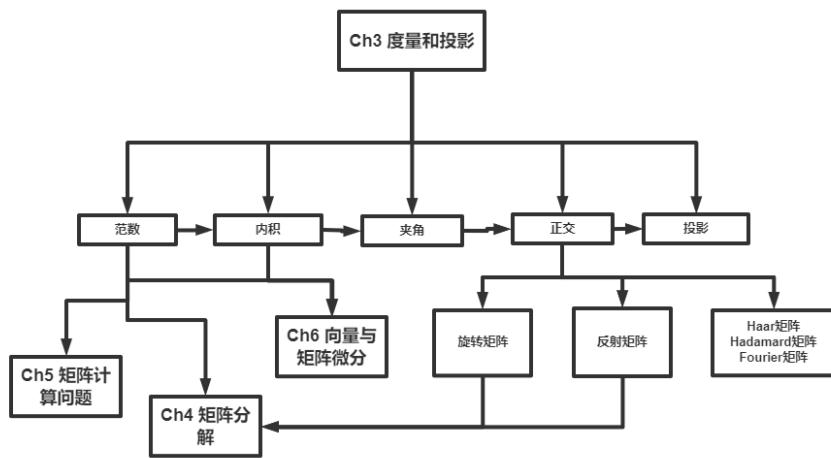


图 3.1: 本章导图

3.1 内积与范数：数据度量的观点

在许多实际的数据科学问题中，常需对同一线性空间中的向量（或矩阵）引入作为它们“大小”的一种度量，进而比较两个向量或矩阵的“接近”程度。引入这种体现其“大小”的量就是范数，它们在理论和实际应用中都占有重要的地位。

例如在第2.1.1节中，我们对纽约时报在2010年12月7日的四则新闻标题都进行了向量化的表示，一个自然的问题是“如何知道两则新闻标题表示的是相关信息？”通过对这四则新闻提要进行简单聚类来实现：

- (a) Suit Over Targeted Killing in Terror Case Is Dismissed ...
- (b) In Tax Deal With G.O.P, a Portent for the Next 2 Years ...
- (c) Obama Urges China to Check North Koreans ...
- (d) Top Test Scores From Shanghai Stun Educators ...

我们可以利用余弦相似度，一种度量向量之间相似性的工具来解答这个聚类问题。

$$\text{sim}_{\cos}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}| |\mathbf{y}|},$$

其中 \mathbf{x}, \mathbf{y} 是文本向量， $\text{sim}_{\cos}(\mathbf{x}, \mathbf{y})$ 表示 \mathbf{x}, \mathbf{y} 的余弦相似度。

在 MNIST 手写数字分类问题中。我们分别从每一类中取一些数据作为训练样本。当我们要对测试样本进行预测时时，根据最近邻算法，我们只需要去找到和这个数据最相似的训练数据所属的类别。我们可以把矩阵拉伸成向量，通过度量向量间的距离来度量矩阵间的差异，从而

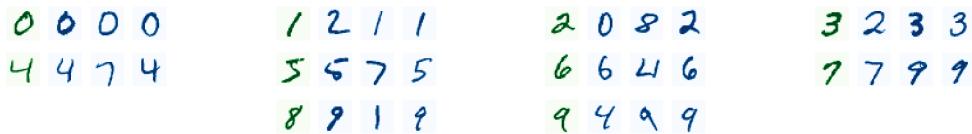


图 3.2: 对 MNIST 数据集进行分类 (绿色的为训练集, 蓝色的为测试集)

可以使用下面的公式计算图片间的差异:

$$d(A, B) = \sum_{jk} |A_{jk} - T_{jk}|$$

其中 d 是手写数字训练图片的表示矩阵 A 和测试图片的表示矩阵 T 之间的距离 (两个矩阵同等大小), j, k 取遍矩阵所有元素。距离越大, 则图片越不相似; 距离约小, 图片越相似。

从这两个例子中, 我们可以看到, 无论是分类还是聚类, 两个数据之间的相似性度量起着一个非常关键的作用。这就需要我们学习向量与向量之间的相似度量方法。下面我们会介绍向量的范数, 长度与距离。

3.1.1 向量范数、长度与距离

向量范数可以看做向量的模或者长度的推广。

例 3.1.1. 复数 $x = (a, b) = a + ib$ 的长度或者模指的是

$$\|x\| = \sqrt{a^2 + b^2}$$

显然复数 x 的模 $\|x\|$ 具有下列三条性质:

- (1) $\|x\| \geq 0$, 当且仅当 $x = 0$ 时等号成立;
- (2) $\|\lambda x\| = |\lambda| \|x\|$, ($\forall \lambda \in \mathbb{R}$);
- (3) $\|x + y\| \leq \|x\| + \|y\|$, ($x, y \in \mathbb{C}$).

例 3.1.2. n 维向量 $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ 的模或长度定义为

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

显然向量 x 的模 $\|x\|$ 也具有下列三条性质:

- (1) $\|x\| \geq 0$, 当且仅当 $x = 0$ 时等号成立;
- (2) $\|\lambda x\| = |\lambda| \|x\|$, ($\forall \lambda \in \mathbb{R}$);
- (3) $\|x + y\| \leq \|x\| + \|y\|$, ($x, y \in \mathbb{R}^n$).

向量的模又称为欧式长度, 代表了从原点 0 到点 x 的直线距离。

从这两个例子可以看出, 向量的模可以看做是向量到实数的一个映射函数, 满足非负性、齐次性和三角不等式成立。我们把它们进一步推广, 可得范数的定义。

定义 3.1.1. 设 \mathbb{V} 是数域上 \mathbb{K} 的 n 维线性空间, 函数

$$\|\cdot\| : \mathbb{V} \rightarrow \mathbb{R},$$

$$\mathbf{x} \mapsto \|\mathbf{x}\|,$$

它把向量 \mathbf{x} 映射为它的长度 $\|\mathbf{x}\| \in \mathbb{R}$, 并且使得对 $\forall \lambda \in \mathbb{R}$ 和 $\forall \mathbf{x}, \mathbf{y} \in \mathbb{V}$, 满足

(1) 非负性: $\|\mathbf{x}\| \geq 0$, $\|\mathbf{x}\| = 0$ 当且仅当 $\mathbf{x} = 0$;

(2) 齐次性: $\|\lambda \mathbf{x}\| = |\lambda| \|\mathbf{x}\|$;

(3) 三角不等式: $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$.

称 $\|\mathbf{x}\|$ 是向量 \mathbf{x} 的向量范数, 称定义了范数的线性空间 \mathbb{V} 为赋范线性空间。

例 3.1.3. 对任给的 $\mathbf{x} = (x_1, x_2, x_3)^T \in \mathbb{C}^3$, 试问如下实值函数是否构成向量范数?

1. $|x_1| + |2x_2 + x_3|$,
2. $|x_1| + |2x_2| - 5|x_3|$.

解. 我们只需要验证实值函数是否满足三条性质。

1. 非负性: $|x_1| + |2x_2 + x_3| \geq 0$;

齐次性: 令 $c \in \mathbb{C}$, $|cx_1| + |2cx_2 + cx_3| = |c|(|x_1| + |2x_2 + x_3|)$;

三角不等式: 令 $\mathbf{x} = (x_1, x_2, x_3)^T, \mathbf{y} = (y_1, y_2, y_3)^T \in \mathbb{C}^3$, 则 $|x_1 + y_1| + |2(x_2 + y_2) + (x_3 + y_3)| \leq |x_1| + |2x_2 + x_3| + |y_1| + |2y_2 + y_3|$.

所以 $|x_1| + |2x_2 + x_3|$ 是一个向量范数。

2. 取 $\mathbf{x} = (0, 0, 1)$ 则 $|0| + |2 \times 0| - 5|1| = -5 < 0$ 不满足非负性。

所以 $|x_1| + |2x_2| - 5|x_3|$ 不是一个向量范数。

3.1.2 l_p 范数

接下来我们看一类常用的范数, l_p 范数。

例 3.1.4. 对于任意 $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, 由

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, 1 \leq p < \infty$$

定义的 $\|\cdot\|_p$ 是 \mathbb{R}^n 上的向量范数, 称为 p 范数或 l_p 范数。

(1) 当 $p = 1$ 时, 得到 1 范数或 l_1 范数, 也称为 Manhattan 范数

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$$

(2) 当 $p = 2$ 时, 得到 2 范数或 l_2 范数, 也称为欧几里得范数

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$$

我们定义 ∞ 范数为 l_p 范数中， p 趋近于无穷的极限。

例 3.1.5. 对于任意 $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, 由

$$\|\mathbf{x}\|_\infty = \lim_{p \rightarrow \infty} \|\mathbf{x}\|_p,$$

也就是,

$$\|\mathbf{x}\|_\infty = \max_{i=1, \dots, n} |x_i|,$$

定义的 $\|\cdot\|_\infty$ 是 \mathbb{R}^n 上的向量范数, 称为 ∞ 范数或 l_∞ 范数。

证明. 验证 $\|\mathbf{x}\|_\infty \equiv \max_i |x_i|$ 是向量范数显然很容易。下证 $\max_i |x_i| = \lim_{p \rightarrow +\infty} \|\mathbf{x}\|_p$ 。令 $\|\mathbf{x}_j\| = \max_j |x_i|$, 则有

$$\begin{aligned} \|\mathbf{x}\|_\infty &= |\mathbf{x}_j| \leq \left(\sum_{i=1}^n |x_i|^p \right)^{(1/p)} = \|\mathbf{x}\|_p \\ &\leq (n|\mathbf{x}_j|^p)^{(1/p)} = n^{(1/p)} \|\mathbf{x}\|_\infty \end{aligned}$$

由极限的夹逼准则, 并注意到 $\lim_{p \rightarrow +\infty} n^{1/p} = 1$, 即得欲证结论。 \square

有的函数并不是范数, 但是也能反映向量间的相似性。

例 3.1.6. 当 $0 < p < 1$, 由

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}$$

定义的 $\|\cdot\|_p$ 不是 \mathbb{R}^n 上的向量范数。

证明. 考虑 $n = 2, p = \frac{1}{2}$. 取 $\boldsymbol{\alpha} = (1, 0)^T, \boldsymbol{\beta} = (0, 1)^T$, 则

$$\begin{aligned} \|\boldsymbol{\alpha}\|_{\frac{1}{2}} &= \|\boldsymbol{\beta}\|_{\frac{1}{2}} = 1, \|\boldsymbol{\alpha} + \boldsymbol{\beta}\|_{\frac{1}{2}} = 4 \\ \|\boldsymbol{\alpha} + \boldsymbol{\beta}\|_{\frac{1}{2}} &\geq \|\boldsymbol{\alpha}\|_{\frac{1}{2}} + \|\boldsymbol{\beta}\|_{\frac{1}{2}} \end{aligned}$$

\square

在数据科学中, 常通过向量中非零元素的数目判断向量的稀疏程度。

定义 3.1.2. 向量 \mathbf{x} 的基数函数定义为 \mathbf{x} 中非零元素的个数, 即

$$card(\mathbf{x}) = \sum_{i=1}^n \mathcal{I}(x_i \neq 0)$$

其中,

$$\mathcal{I}(x_i \neq 0) = \begin{cases} 1 & , x_i \neq 0 \\ 0 & , x_i = 0 \end{cases}$$

基数函数也被称为 l_0 范数，但是它并不满足范数定义的条件。

例 3.1.7. 求向量 $\mathbf{x} = (-1, 2, 4)^T$ 的 0, 1, 2, 和 ∞ -范数。

解.

$$\|\mathbf{x}\|_0 = 3$$

$$\|\mathbf{x}\|_1 = |-1| + 2 + 4 = 7$$

$$\|\mathbf{x}\|_2 = \sqrt{|-1|^2 + 2^2 + 4^2} = \sqrt{21}$$

$$\|\mathbf{x}\|_\infty = \max\{|-1|, 2, 4\} = 4$$

3.1.3 范数的几何意义与性质

定义 3.1.3. 对于 l_p 范数小于等于 1 的向量集合,

$$\mathcal{B}_p = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_p \leq 1\}$$

称为 l_p 的单位范数球。

例 3.1.8. 单位范数球的形状反映了不同范数的性质，对于不同的 p ，范数球有着不同的几何形状。图3.3分别表示了 $\mathcal{B}_2, \mathcal{B}_1, \mathcal{B}_\infty$ 在 \mathbb{R}^2 的范数球形状。

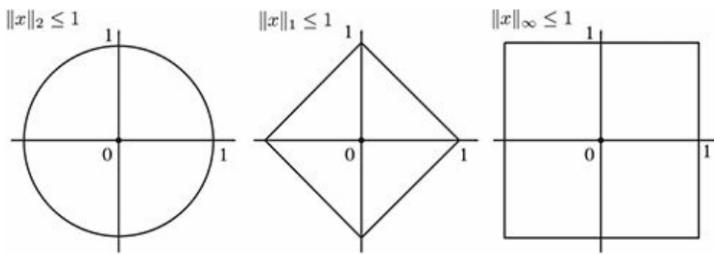


图 3.3: \mathbb{R}^2 上的范数球

范数的性质

定义 3.1.4. 设 $\{\mathbf{x}^{(k)}\}$ 为 \mathbb{R}^n 中一向量序列, $\mathbf{x}^* \in \mathbb{R}^n$, 其中

$$\mathbf{x}^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})^T, \mathbf{x}^* = (x_1^*, x_2^*, \dots, x_n^*)^T$$

如果 $\lim_{k \rightarrow \infty} x_i^{(k)} = x_i^* (i = 1, 2, \dots, n)$, 则称 $\mathbf{x}^{(k)}$ 收敛于向量 \mathbf{x}^* , 记作

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}^*$$

或者称 $\{\mathbf{x}^{(k)}\}$ 依坐标收敛于 \mathbf{x}^*

定理 3.1.1. (范数的连续性) 设非负函数 $N(\mathbf{x}) = \|\mathbf{x}\|$ 为 \mathbb{R}^n 上任一向量范数, 则 $N(\mathbf{x})$ 是 \mathbf{x} 分量 x_1, x_2, \dots, x_n 的连续函数。

证明. 设 $\mathbf{x} = \sum_{i=1}^n x_i \mathbf{e}_i, \mathbf{y} = \sum_{i=1}^n y_i \mathbf{e}_i$, 其中 $\mathbf{e}_i = (0, \dots, 1, 0, \dots, 0)^T$ (即第 i 个元素为 1)。只需证明当 $\mathbf{x} \rightarrow \mathbf{y}$ 时, $N(\mathbf{x}) \rightarrow N(\mathbf{y})$ 即可。事实上,

$$\begin{aligned} |N(\mathbf{x}) - N(\mathbf{y})| &= |\|\mathbf{x}\| - \|\mathbf{y}\|| \leq \|\mathbf{x} - \mathbf{y}\| = \left\| \sum_{i=1}^n (x_i - y_i) \mathbf{e}_i \right\| \\ &\leq \sum_{i=1}^n |x_i - y_i| \|\mathbf{e}_i\| \leq \|\mathbf{x} - \mathbf{y}\|_\infty \sum_{i=1}^n \|\mathbf{e}_i\| \end{aligned}$$

即

$$|N(\mathbf{x}) - N(\mathbf{y})| \leq c \|\mathbf{x} - \mathbf{y}\|_\infty \rightarrow 0 \quad (\text{当 } \mathbf{x} \rightarrow \mathbf{y} \text{ 时}),$$

其中

$$c = \sum_{i=1}^n \|\mathbf{e}_i\|.$$

□

例 3.1.9. 在 \mathbb{R}^n (或 \mathbb{C}^n) 上可以定义各种向量范数, 其数值大小一般不同, 但是在各种向量范数之间存在下述重要的关系

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_1 \leq n \|\mathbf{x}\|_\infty$$

$$\frac{1}{\sqrt{n}} \|\mathbf{x}\|_1 \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1$$

$$\frac{1}{\sqrt{n}} \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2$$

或者

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{n} \|\mathbf{x}\|_2 \leq n \|\mathbf{x}\|_\infty$$

定理 3.1.2. (范数的等价性) 设 $\|\mathbf{x}\|_s, \|\mathbf{x}\|_t$ 为 \mathbb{R}^n 上向量的任意两种范数, 则存在常数 $c_1, c_2 > 0$, 使得

$$c_1 \|\mathbf{x}\|_s \leq \|\mathbf{x}\|_t \leq c_2 \|\mathbf{x}\|_s, \quad \text{对一切 } \mathbf{x} \in \mathbb{R}^n.$$

证明. 只要就 $\|\mathbf{x}\|_s = \|\mathbf{x}\|_\infty$ 证明上式成立即可, 即证明存在常数 $c_1, c_2 > 0$, 使

$$c_1 \leq \frac{\|\mathbf{x}\|_t}{\|\mathbf{x}\|_\infty} \leq c_2, \quad \text{对一切 } \mathbf{x} \in \mathbb{R}^n \text{ 且 } \mathbf{x} \neq \mathbf{0}.$$

考虑泛函 $f(\mathbf{x}) = \|\mathbf{x}\|_t \geq 0, \mathbf{x} \in \mathbb{R}^n$.

记 $S = \{\mathbf{x} | \|\mathbf{x}\|_\infty = 1, \mathbf{x} \in \mathbb{R}^n\}$, 则 S 是一个有界闭集。由于 $f(\mathbf{x})$ 为 S 上的连续函数, 所以 $f(\mathbf{x})$ 于 S 上达到最大、最小值。设 $\mathbf{x} \in \mathbb{R}^n$ 且 $\mathbf{x} \neq 0$, 则 $\frac{\mathbf{x}}{\|\mathbf{x}\|_\infty} \in S$, 从而有

$$f(\mathbf{x}') = c_1 \leq f\left(\frac{\mathbf{x}}{\|\mathbf{x}\|_\infty}\right) \leq c_2 = f(\mathbf{x}''),$$

其中 $\mathbf{x}', \mathbf{x}'' \in S$ 。显然 $c_1, c_2 > 0$, 上式 $c_1 \leq \left\| \frac{\mathbf{x}}{\|\mathbf{x}\|_\infty} \right\| \leq c_2$, 即

$$c_1 \|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_t \leq c_2 \|\mathbf{x}\|_\infty, \quad \text{对一切 } \mathbf{x} \in \mathbb{R}^n.$$

□

注意, 定理3.1.2不能推广到无穷维空间。由定理3.1.2可得到结论: 如果在某一种范数意义下向量序列收敛, 则在任何一种范数意义下该向量序列亦收敛。

定理 3.1.3. (向量序列收敛定理) 设 $\{\mathbf{x}^{(k)}\}$ 为 \mathbb{R}^n 中一向量序列, $\mathbf{x}^* \in \mathbb{R}^n$ 则

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}^* \iff \lim_{k \rightarrow \infty} \left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\| = 0$$

其中 $\|\cdot\|$ 为向量的任一种范数。若 $\lim_{k \rightarrow \infty} \left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\| = 0$, 称向量序列 $\{\mathbf{x}^{(k)}\}$ 依范数收敛于 \mathbf{x}^* 。

证明. 显然, 对于 ∞ - 范数, 命题成立。即

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}^* \iff \left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\|_\infty \rightarrow 0 \text{ (当 } k \rightarrow \infty \text{ 时),}$$

而对于 \mathbb{R}^n 上任一种范数 $\|\cdot\|$, 由定理3.1.2, 存在常数 $c_1, c_2 > 0$, 使

$$c_1 \left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\|_\infty \leq \left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\|_t \leq c_2 \left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\|_\infty,$$

于是又有

$$\left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\|_\infty \rightarrow 0 \iff \left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\|_t \rightarrow 0 \text{ (当 } k \rightarrow \infty \text{ 时)}$$

□

这就说明向量列依坐标收敛等价于依范数收敛。

3.1.4 内积与夹角

内积引入了直观的几何概念, 例如向量的长度以及两个向量之间的角度或距离。引入内积的另外一个目的是确定向量是否彼此正交的。

内积的定义

定义 3.1.5. n 维实向量空间 \mathbb{R}^n 的标准内积(点积)是两个向量的对应元素乘积之和, 即

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

通常我们指内积都是指这种标准内积。下面给出一般性内积的定义。

定义 3.1.6. 设向量 $\mathbf{x}, \mathbf{y} \in \mathbb{V} \subset \mathbb{R}^n$, 假设有一个从 $\mathbb{V} \times \mathbb{V} \rightarrow \mathbb{R}$ 的函数 $\langle \mathbf{x}, \mathbf{y} \rangle$, 它满足

- (1) 非负性: 对于 $\forall \mathbf{x} \in \mathbb{V}$, 有 $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$, $\langle \mathbf{x}, \mathbf{x} \rangle = 0$ 当且仅当 $\mathbf{x} = 0$;
- (2) 对称性: $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$;
- (3) 齐次性: 对于 $\forall \lambda \in \mathbb{R}, \mathbf{x}, \mathbf{y} \in \mathbb{V}$, 有 $\langle \lambda \mathbf{x}, \mathbf{y} \rangle = \lambda \langle \mathbf{x}, \mathbf{y} \rangle$;
- (4) 线性性: 对于 $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{V}$, 有 $\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$ 。

称 $\langle \mathbf{x}, \mathbf{y} \rangle$ 是向量 \mathbf{x}, \mathbf{y} 的内积, 称定义了内积的线性空间 \mathbb{V} 为内积空间。若内积是点积时, 称定义了标准内积的线性空间为欧氏空间。

例 3.1.10. 考虑 $\mathbb{V} = \mathbb{R}^2$. 如果我们定义

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

则 $\langle \cdot, \cdot \rangle$ 是一个内积, 但不是点积。

对称、正定矩阵表示内积

我们可以通过正定矩阵来定义内积。

- 考虑一个定义了内积的 n 维线性空间 \mathbb{V} 以及其上的内积 $\langle \cdot, \cdot \rangle : \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{R}$ 和有序基底 $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ 。对任意的 $\mathbf{x}, \mathbf{y} \in \mathbb{V}$, 可以用基向量线性表出, 也即 $\mathbf{x} = \sum_{i=1}^n \psi_i \mathbf{b}_i \in \mathbb{V}$ 以及 $\mathbf{y} = \sum_{j=1}^n \lambda_j \mathbf{b}_j \in \mathbb{V}$ 。
- 由内积的线性性, 可得 \mathbf{x}, \mathbf{y} 的内积

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

其中 $A_{ij} := \langle \mathbf{b}_i, \mathbf{b}_j \rangle$, $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ 分别是 \mathbf{x}, \mathbf{y} 的坐标。

- 所以内积是被 \mathbf{A} 唯一决定了, 而内积的对称性决定了 \mathbf{A} 也是对称的。
- 进一步地, 由内积的非负性可得

$$\forall \mathbf{x} \in \mathbb{V} \setminus \{\mathbf{0}\} : \mathbf{x}^T \mathbf{A} \mathbf{x} > 0.$$

例 3.1.11. 考虑下列矩阵

$$\mathbf{A}_1 = \begin{pmatrix} 9 & 6 \\ 6 & 5 \end{pmatrix}, \mathbf{A}_2 = \begin{pmatrix} 9 & 6 \\ 6 & 3 \end{pmatrix}$$

则 \mathbf{A}_1 是对称正定矩阵。因为它是对称的且对于任意的 $\mathbf{x} \in \mathbb{V} \setminus \{\mathbf{0}\}$ 有

$$\mathbf{x}^T \mathbf{A}_1 \mathbf{x} = (x_1, x_2) \begin{pmatrix} 9 & 6 \\ 6 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 9x_1^2 + 12x_1x_2 + 5x_2^2 = (3x_1 + 2x_2)^2 + x_2^2 > 0$$

\mathbf{A}_2 对称的但不正定。因为 $\mathbf{x}^T \mathbf{A}_2 \mathbf{x} = 9x_1^2 + 12x_1x_2 + 3x_2^2 = (3x_1 + 2x_2)^2 - x_2^2$ 可以小于 0。(比如 $\mathbf{x} = (2, -3)^T$ 时)

如果 $A \in \mathbb{R}^{n \times n}$ 是对称、正定的，则

$$\langle \mathbf{x}, \mathbf{y} \rangle = \hat{\mathbf{x}}^T A \hat{\mathbf{y}}$$

定义了一个关于有序基底 B 的内积，其中 $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ 是 \mathbb{V} 中向量 \mathbf{x}, \mathbf{y} 关于 B 下的坐标。

定理 3.1.4. 对于一个实值有限维空间 \mathbb{V} 和 \mathbb{V} 下一个有序基底 B ，如果 $\langle \cdot, \cdot \rangle : \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{R}$ 是一个内积当且仅当存在一个对称、正定矩阵 $A \in \mathbb{R}^{n \times n}$ 满足

$$\langle \mathbf{x}, \mathbf{y} \rangle = \hat{\mathbf{x}}^T A \hat{\mathbf{y}}$$

如果矩阵 $A \in \mathbb{R}^{n \times n}$ 是对称正定矩阵，则

- A 的核（零空间）只包含 $\mathbf{0}$ 因为 $\mathbf{x}^T A \mathbf{x} > 0$ 对于任意 $\mathbf{x} \neq 0$ 成立，即如果 $\mathbf{x} \neq 0$ 则 $A\mathbf{x} \neq 0$ ；
- A 的对角元是正的，因为 $a_{ii} = \mathbf{e}_i^T A \mathbf{e}_i > 0$ ，其中 \mathbf{e}_i 是 \mathbb{R}^n 中的标准基。

内积定义范数

内积和范数有着紧密的联系，我们可以利用内积来定义一个向量的范数。

已知向量范数需要满足

- (1) 非负性 $\|\mathbf{x}\| \geq 0$;
- (2) 齐次性 $\|a\mathbf{x}\| = |a|\|\mathbf{x}\|$;
- (3) 三角不等式 $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$.

如果我们将 $\|\mathbf{x}\|$ 定义为 $\sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ ，容易验证 $\sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ 满足范数的三个要求。

从这个角度看，一个内积空间天然包含有一个赋范线性空间。

定义 3.1.7. 设 \mathbb{V} 是内积空间，则由

$$\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}, \quad \forall \mathbf{x} \in \mathbb{V}$$

定义的函数 $\|\cdot\|$ 是 \mathbb{V} 上的向量范数，称为由内积 $\langle \cdot, \cdot \rangle$ 导出的范数。

并不是每个范数都可以由内积导出，如 l_1 和 l_∞ 范数不能由内积导出。

标准内积与 2-范数之间存在联系：

$$\|\mathbf{x}\|_2^2 = \mathbf{x}^T \mathbf{x}$$

柯西施瓦兹不等式

定理 3.1.5. [柯西施瓦兹不等式] 若 $\|\cdot\|$ 是由 $(\mathbb{V}, \langle \cdot, \cdot \rangle)$ 导出的范数，那么

$$\langle \mathbf{x}, \mathbf{y} \rangle^2 \leq \|\mathbf{x}\|^2 \|\mathbf{y}\|^2.$$

证明. 当 $\mathbf{y} = \mathbf{0}$ 时, 不等式成立。

当 $\mathbf{y} \neq \mathbf{0}$ 时, 对任意 $\lambda \in \mathbb{R}$,

$$\begin{aligned} 0 &\leq \langle \mathbf{x} - \lambda \mathbf{y}, \mathbf{x} - \lambda \mathbf{y} \rangle \\ &= \langle \mathbf{x} - \lambda \mathbf{y}, \mathbf{x} \rangle - \lambda \langle \mathbf{x} - \lambda \mathbf{y}, \mathbf{y} \rangle \\ &= \langle \mathbf{x}, \mathbf{x} \rangle - \lambda \langle \mathbf{x}, \mathbf{y} \rangle - \lambda^2 \langle \mathbf{y}, \mathbf{y} \rangle + \langle \mathbf{y}, \mathbf{y} \rangle \\ &= \|\mathbf{x}\|^2 - 2\lambda \langle \mathbf{x}, \mathbf{y} \rangle + \lambda^2 \|\mathbf{y}\|^2 \end{aligned}$$

取 $\lambda = \langle \mathbf{x}, \mathbf{y} \rangle \|\mathbf{y}\|^{-2}$, 得

$$0 \leq \|\mathbf{x}\|^2 - \langle \mathbf{x}, \mathbf{y} \rangle^2 \|\mathbf{y}\|^{-2}$$

从而得到

$$\langle \mathbf{x}, \mathbf{y} \rangle^2 \leq \|\mathbf{x}\|^2 \|\mathbf{y}\|^2.$$

或者

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\| \|\mathbf{y}\|.$$

□

例 3.1.12. 令 $\mathbf{x} = (1, 1)^T \in \mathbb{R}^2$, 如果我们把点积作为内积, 则向量 \mathbf{x} 的长度为

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}} = \sqrt{1^2 + 1^2} = \sqrt{2}.$$

我们现在采用一个不同的内积

$$\langle \mathbf{x}, \mathbf{y} \rangle := \mathbf{x}^T \begin{pmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{pmatrix} \mathbf{y} = x_1 y_1 - \frac{1}{2}(x_1 y_2 + x_2 y_1) + x_2 y_2,$$

则向量长度为

$$\langle \mathbf{x}, \mathbf{x} \rangle = x_1^2 - x_1 x_2 + x_2^2 = 1 - 1 + 1 = 1 \implies \|\mathbf{x}\| = \sqrt{1} = 1.$$

所以相对于点积这个内积使得 \mathbf{x} 变短了。事实上, 在 x_1, x_2 同号的情况下, 上述内积会给出一个比点积更小的向量长度值; 如果异号则给出更大的值。

利用范数或者内积, 我们可以定义两个向量间的距离。

定义 3.1.8. 考虑一个赋范空间 $(\mathbb{V}, \|\cdot\|)$ 。我们称

$$d(\mathbf{x}, \mathbf{y}) := \|\mathbf{x} - \mathbf{y}\|$$

为 $\mathbf{x}, \mathbf{y} \in \mathbb{V}$ 的距离。

如果 \mathbb{V} 是一个内积空间 $(\mathbb{V}, \langle \cdot, \cdot \rangle)$ 。

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

如果我们用点积作为内积, 则上述距离称为欧几里得距离, 简称欧氏距离。

和向量长度类似，向量间的距离不必需要内积，使用范数就足够了。如果我们使用内积导出的范数，则距离会依赖于内积的选择。我们给度量一个数学上的定义。

定义 3.1.9. 考虑一个内积空间 $(\mathbb{V}, \langle \cdot, \cdot \rangle)$ ，我们称映射

$$d : \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{R}$$

$$(\mathbf{x}, \mathbf{y}) \mapsto d(\mathbf{x}, \mathbf{y})$$

为度量。

定义 3.1.10. 一个度量空间由一个有序对 (\mathbb{V}, d) 表示，其中 \mathbb{V} 是一种集合， d 是定义在 V 上的一种度量：

$$d : \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{R}$$

且对任意 $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{V}$ ，需满足

- 非负性：即 $d(\mathbf{x}, \mathbf{y}) \geq 0$ ，且 $d(\mathbf{x}, \mathbf{y}) = 0 \iff \mathbf{x} = \mathbf{y}$ ；
- 对称性：即 $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ ；
- 三角不等式： $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

所以赋范线性空间按由范数导出的距离构成一个特殊的度量空间。度量空间也称为距离空间。

夹角的定义

有了内积和范数，便可以定义两个向量之间的角度。例如，假设笛卡尔坐标系中有两个非零向量 \mathbf{x}, \mathbf{y} ，它们与原点 \mathbf{o} 构成一个三角形，如图3.4所示。令 θ 是 \mathbf{ox} 与 \mathbf{oy} 之间的夹角， $\mathbf{z} = \mathbf{x} - \mathbf{y}$ 。对三角形 yxx' 运用勾股定理，有

$$\begin{aligned}\|\mathbf{z}\|_2^2 &= (\|\mathbf{y}\|_2 \sin \theta)^2 + (\|\mathbf{x}\|_2 - \|\mathbf{y}\|_2 \cos \theta)^2 \\ &= \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 - 2 \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \cos \theta.\end{aligned}$$

由于

$$\|\mathbf{z}\|_2^2 = \|\mathbf{x} - \mathbf{y}\|_2^2 = (\mathbf{x} - \mathbf{y})^\top (\mathbf{x} - \mathbf{y}) = \mathbf{x}^\top \mathbf{x} + \mathbf{y}^\top \mathbf{y} - 2 \mathbf{x}^\top \mathbf{y},$$

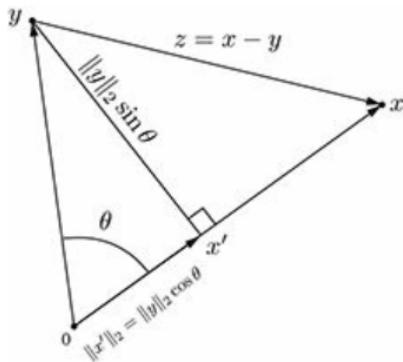
则有

$$\mathbf{x}^\top \mathbf{y} = \|\mathbf{x}\|_2 \|\mathbf{y}\| \cos \theta.$$

则向量 \mathbf{x}, \mathbf{y} 之间的夹角为

$$\cos \theta = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|} \tag{3.1}$$

当 $\mathbf{x}^\top \mathbf{y} = 0$ 时，向量 \mathbf{x}, \mathbf{y} 之间的角度为 90° ，称为正交。当 θ 为 0° 或者 180° 时， \mathbf{x}, \mathbf{y} 成一直线，即 $\mathbf{y} = k\mathbf{x}, k \in \mathbb{K}$ ，称为平行。

图 3.4: 向量 x, y 之间的夹角 θ

向量的正交

定义 3.1.11. 设向量 $x, y \in \mathbb{X}$, 如果 $\langle x, y \rangle = 0$, 则称 x, y 正交, 记作 $x \perp y$ 。特别地, 如果 $\|x\| = 1 = \|y\|$, 也即是单位向量时, 称 x, y 标准正交。

零向量与任何向量正交。

对于非零向量组 $\{x_1, x_2, \dots, x_d\}$, 如果对于 $\forall i \neq j$, 有 $\langle x_i, x_j \rangle = 0$, 则称向量组两两正交, 并且具有如下性质:

命题 3.1.1. 两两正交的向量组线性无关。

例 3.1.13. 考虑两个向量 $x = (1, 1)^T, y = (-1, 1)^T \in \mathbb{R}^2$ 。我们用两种不同的内积来确定它们之间的夹角 ω 。使用点积作为内积则可以得到 ω 为 90° , 所以 $x \perp y$ 。

而我们选择内积

$$\langle x, y \rangle = x^T \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} y$$

计算 x, y 之间的角度 ω 时,

$$\cos \omega = \frac{\langle x, y \rangle}{\|x\| \|y\|} = -\frac{1}{3} \implies \omega \approx 109.5^\circ$$

所以 x, y 不是正交的。

因此向量在一种内积下正交并不代表它们在其他内积下也正交。

定义 3.1.12. 方阵 $A \in \mathbb{R}^{n \times n}$ 是一个正交矩阵当且仅当它的列向量是标准正交的, 即

$$AA^T = I = A^T A,$$

因此 $A^{-1} = A^T$ 。

正交矩阵变换是特殊的，因为用正交矩阵 A 作用一个向量 x 时，向量 x 的长度不变。对于点积，我们得到

$$\|Ax\|^2 = (Ax)^T(Ax) = x^T A^T A x = x^T I x = x^T x = \|x\|^2.$$

并且两个向量 x, y 的夹角也不会在正交矩阵的作用下改变。同样用点积作为内积，则 Ax 和 Ay 的夹角为

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

这就是向量 x, y 之间的夹角。这就意味着正交矩阵 A 能够保持角度和长度不变。

3.1.5 数据科学中常用的相似性度量 I

聚类 (clustering) 和分类 (classification) 是数据分析的重要运算。所谓聚类，就是将一给定的大数据集聚为几个小的子数据集，并且每个子集 (目标类) 的数据都具有共同或者相似的特征。分类则是将一个或者多个未知类属的数据或特征向量划分到具有最接近特征的某个已知目标类别中。

在很多聚类与分类算法中一个很重要的数学工具就是相似性度量。本小节主要讨论非概率相关的相似性度量。

关于向量的相似性度量

在一个向量空间中，两点之间是否相似最直观的就是相近的相似。也就是说，我们可以将聚集在一起的点认为它们是相似的，而距离较远的点则相似度就低。

假设有 m 个样本，每个样本由 n 个属性的特征向量组成。样本集合可以用矩阵 X 表示

$$X = [x_{ij}]_{m \times n} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix}$$

矩阵的第 j 列表示第 j 个样本，第 i 行表示第 i 个属性，矩阵元素 x_{ij} 表示第 j 个样本的第 i 个属性值； $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$ 。

定义 3.1.13. 给定特征空间或样本集合 X , X 是由范数或内积导出的 m 维度量空间 \mathbb{R}^m 中点的集合，其中 $x_i, x_j \in X$, $x_i = (x_{1i}, x_{2i}, \dots, x_{mi})^T$, $x_j = (x_{1j}, x_{2j}, \dots, x_{mj})^T$, 样本 x_i 与样本 x_j 的闵可夫斯基距离，简称闵氏距离，定义为

$$d(x_i, x_j) = \sqrt[p]{\sum_{k=1}^m |x_{ki} - x_{kj}|^p} = \|x_i - x_j\|_p,$$

其中 $1 \leq p < \infty$ 。

- 当 $p = 2$ 时，对应欧氏距离
- 当 $p = 1$ 时，对应曼哈顿距离
- 当 $p \rightarrow \infty$ 时，对应切比雪夫距离

定义 3.1.14. 欧氏距离是指的是多维空间中各个点之间的直线距离，计算公式如下：

$$dist_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} = \|\mathbf{x} - \mathbf{y}\|_2$$

定义 3.1.15. 曼哈顿距离也称出租车距离，用以标明两个点在标准坐标系上的绝对轴距总和。计算公式如下：

$$dist(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n |x_i - y_i| = \|\mathbf{x} - \mathbf{y}\|_1$$

在 1-范数意义下的距离，我们称为曼哈顿距离。这是因为曼哈顿城的道路总是横着或者竖着，我们要计算从一点走到另外一点的距离就不能够使用两点之间的直线距离了。

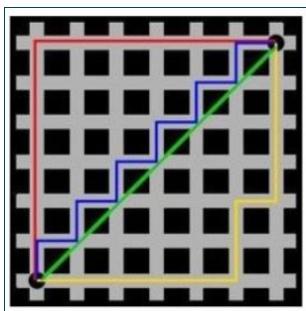


图 3.5: 曼哈顿距离

如图3.5所示，绿线代表欧氏距离，红线代表曼哈顿距离，蓝、黄线代表等价的曼哈顿距离。

定义 3.1.16. 切比雪夫距离是将二个点其各坐标数值差绝对值的最大值作为距离。

$$dist(\mathbf{x}, \mathbf{y}) = \max |x_i - y_i| = \|\mathbf{x} - \mathbf{y}\|_\infty$$

这个距离乍一看非常奇怪，实际上它类似于国际象棋中国王的走法，相当于国王从格子 (x_1, y_1) 走到格子 (x_2, y_2) 最少需要多少步。

	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1	1	1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

图 3.6: 国际象棋中的切比雪夫距离

 k -NN

下面我们将以闵式距离为例，用 k -NN 算法来展示不同相似性度量对于模型的影响。

例 3.1.14. k -近邻算法 (k -NN) 是机器学习中一种非常简单的算法。给定带类别的数据。当预测新的数据属于哪一类别时，我们只需比较距离这一数据最近的 k 个已知数据点中哪种类别是多数，则认为这个数据点就是该类别。

比如取 $k = 3$ ，图中的黑色点 (菱形) 即是所要预测的数据点，而蓝色 (圆点) 为正例，红色 (叉) 为负例。因为距离最近的三个点中，有两个是正例，一个为负例。故我们认为这个数据点为正例。

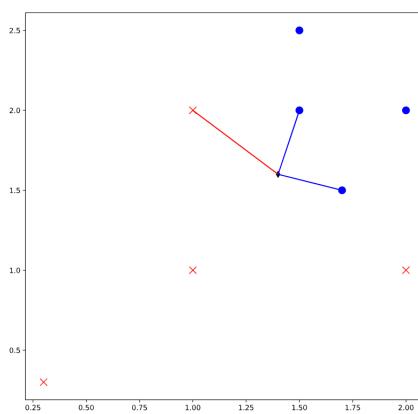


图 3.7: k -NN，待预测样本最接近的三个样本中，有两个是正类，一个是负类，因此，我们预测其为正类。

为了说明不同度量对模型的影响，给定训练集：

正例为：(1.5, 2), (1.7, 1.5), (2, 2), (1.5, 2.5)

负例为：(1, 2), (0.3, 0.3), (2, 1), (1, 1)

固定 $k = 3$, 然后将平面分成两部分, 一部分涂上红色表示某模型将此区域的点预测为负例, 另外一部分涂成蓝色表示正例。

哪种度量方式更好呢? 这取决于具体的问题以及给出的数据。如果我们确定给出的数据都是准确值没有任何误差。我们就有理由相信右边的模型比左边的模型更好。如果我们不能保证给出的数据都是准确值, 那左边的模型也有可能比右边的更好。

图3.8左图采用的距离度量方式是欧氏距离, 右图采用的是曼哈顿距离。

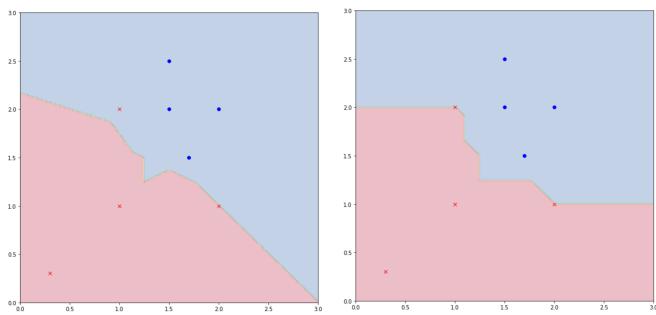


图 3.8: 左图采用欧氏距离, 右图采用曼哈顿距离

需要注意, 闵氏距离, 包括曼哈顿距离、欧氏距离和切比雪夫距离都存在明显的缺点。

举个例子: 二维样本(身高, 体重), 其中身高范围是 [150,190], 体重范围是 [50,60], 有三个样本: a(180,50), b(190,50), c(180,60)。那么 a 与 b 之间的闵氏距离(无论是曼哈顿距离、欧氏距离或切比雪夫距离)等于 a 与 c 之间的闵氏距离, 但是身高的 10cm 真的等价于体重的 10kg 么? 因此用闵氏距离来衡量这些样本间的相似度很有问题。

在学习了概率论之后, 我们将会给出解决这个问题的方案。

前面我们使用样本特征向量之间的闵氏距离作为相似性度量, 我们也可以考虑从特征向量之间的夹角来界定相似程度。

定义 3.1.17. 余弦相似度是通过计算两个样本特征向量 \mathbf{x}_i 和 \mathbf{x}_j 之间夹角的余弦值, 以此作为两个样本间相似度大小的衡量, 计算公式如下

$$\text{sim}_{\cos}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i||\mathbf{x}_j|} = \frac{\sum_{k=1}^m x_{ki}x_{kj}}{\left[\sum_{k=1}^m x_{ki}^2 \sum_{k=1}^m x_{kj}^2\right]^{\frac{1}{2}}}$$

显然因为夹角余弦取值范围为 [-1,1], 所以余弦相似度的取值范围也是 [-1,1]。

夹角余弦越大表示两个向量的夹角越小, 夹角余弦越小表示两向量的夹角越大。

当两个向量的方向重合时夹角余弦取最大值 1, 当两个向量的方向完全相反夹角余弦取最小值 -1。

例 3.1.15. 回顾例2.1.1中纽约时报的四则新闻提要，我们知道它们的分别可以用向量表示为：

$$\begin{aligned}\mathbf{a}' &= \left(\frac{1}{3}, \frac{2}{3}, 0, 0, 0, 0\right)^T, \\ \mathbf{b}' &= \left(\frac{1}{10}, 0, \frac{3}{10}, \frac{1}{5}, \frac{2}{5}, 0\right)^T, \\ \mathbf{c}' &= \left(0, 0, 0, \frac{1}{2}, 0, \frac{1}{2}\right)^T, \\ \mathbf{d}' &= \left(0, 0, 0, 0, 0, 1\right)^T.\end{aligned}$$

利用夹角的概念，经过计算，可得两两新闻提要之间的余弦相似度如表3.1所示：

表 3.1: 四则新闻标题两两之间的余弦夹角

$\cos \theta$	\mathbf{a}'	\mathbf{b}'	\mathbf{c}'	\mathbf{d}'
\mathbf{a}'	1	0.0816	0	0
\mathbf{b}'	0.0816	1	0.2582	0
\mathbf{c}'	0	0.2582	1	0.7071
\mathbf{d}'	0	0	0.7071	1

当两则新闻提要之间没有重复的单词出现，夹角余弦值为 0；当两则新闻提要是相同的，夹角余弦值为 1。

余弦相似度从夹角上区分差异，而对绝对的数值不敏感，因此没法衡量每个维度上数值的差异，我们通过下例进行说明：

例 3.1.16. 用户对内容评分，按 5 分制， X 和 Y 两个用户对两个内容的评分分别为 $(1, 2)$ 和 $(4, 5)$ 。

- X 和 Y 之间的余弦相似度 0.98，两者极为相似。但从评分上看 X 似乎不喜欢这两个内容，而 Y 则比较喜欢。
- 余弦相似度对数值的不敏感导致了结果的误差，需要修正这种不合理性就出现了调整余弦相似度，即所有维度上的数值都减去一个均值。
- 假设两个内容评分均值都是 3，那么调整后为 $(-2, -1)$ 和 $(1, 2)$ ，再用余弦相似度计算，得到 -0.8，相似度为负值并且差异不小，但显然更加符合现实。

除了这些，我们还有一个比较有用的相似性度量。

定义 3.1.18. 汉明距离表示两个（相同长度）字符串对应位置上的值不等的个数。

例 3.1.17. 例如：

- 1011101 与 1001001 之间的汉明距离是 2。
- 2143896 与 2233796 之间的汉明距离是 3。
- "toned" 与 "roses" 之间的汉明距离是 3。

这个距离常常用在字符串的处理上，但显然我们可以将其拓展应用到向量上。

3.1.6 矩阵的内积与范数

将向量的内积与范数加以推广，即可引出矩阵的内积与范数。

(广义) 矩阵范数

定义 3.1.19. 令 $m \times n$ 实矩阵 $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n]$ ，将这个矩阵“拉长”为 $mn \times 1$ 向量

$$\mathbf{a} = \text{vec}(\mathbf{A}) = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}$$

$\text{vec}(\mathbf{A})$ 称为矩阵 \mathbf{A} 的(列)向量化。

利用向量的内积和范数表达，即可以得到下面有关矩阵内积和范数的定义。

定义 3.1.20. 设矩阵 \mathbf{A} 和 \mathbf{B} 是 $m \times n$ 实矩阵，其矩阵内积为：

$$\langle \mathbf{A}, \mathbf{B} \rangle = \langle \text{vec}(\mathbf{A}), \text{vec}(\mathbf{B}) \rangle = \sum_{i=1}^n \mathbf{a}_i \mathbf{b}_i = \sum_{i=1}^n \langle \mathbf{a}_i, \mathbf{b}_i \rangle \quad (3.2)$$

或等价写作

$$\langle \mathbf{A}, \mathbf{B} \rangle = \text{vec}(\mathbf{A})^T \text{vec}(\mathbf{B}) = \text{Tr}(\mathbf{A}^T \mathbf{B}) \quad (3.3)$$

定义 3.1.21. 对于任意的 $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ ，与 $c \in \mathbb{R}$ 。如果函数 $\|\cdot\| : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ 满足条件

(1) $\|\mathbf{A}\| \geq 0 (\|\mathbf{A}\| = 0 \iff \mathbf{A} = \mathbf{0})$ (正定条件);

(2) $\|c\mathbf{A}\| = |c|\|\mathbf{A}\|$, c 为实数 (齐次条件);

(3) $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$ (三角不等式);

则称 $\|\cdot\|$ 是 $\mathbb{R}^{m \times n}$ 上的一个(广义)矩阵范数。

例 3.1.18. 对任意 $\mathbf{A} \in \mathbb{R}^{m \times n}$ ，由

$$\|\mathbf{A}\|_{m_1} := \sum_{i=1}^m \sum_{j=1}^n |a_{ij}|$$

定义的 $\|\cdot\|_{m_1}$ 是 $\mathbb{R}^{m \times n}$ 上的矩阵范数，称为 l_1 范数。

证明. 容易验证:

(1) $\|\mathbf{A}\|_{m_1} \geq 0$, 并且当 $\mathbf{A} = \mathbf{O}$ 即 $a_{ij} \equiv 0$ 时 $\|\mathbf{A}\|_{m_1} = 0$;

(2) $\|c\mathbf{A}\|_{m_1} = \sum_{i=1}^m \sum_{j=1}^n |ca_{ij}| = |c| \sum_{i=1}^m \sum_{j=1}^n |a_{ij}| = |c| \|\mathbf{A}\|_{m_1}$;

(3) $\|\mathbf{A} + \mathbf{B}\|_{m_1} = \sum_{i=1}^m \sum_{j=1}^n (|a_{ij} + b_{ij}|) \leq \sum_{i=1}^m \sum_{j=1}^n (|a_{ij}| + |b_{ij}|) = \|\mathbf{A}\|_{m_1} + \|\mathbf{B}\|_{m_1}$ 。

因此, 实函数 $\|\mathbf{A}\|_{m_1} = \sum_{i=1}^m \sum_{j=1}^n |a_{ij}|$ 是一种矩阵范数。

实际上, 这个范数就是 $\text{vec}(\mathbf{A})$ 的 l_1 范数。 \square

例 3.1.19. 对任意 $\mathbf{A} \in \mathbb{R}^{m \times n}$, 由

$$\|\mathbf{A}\|_{m_\infty} := \max_{1 \leq i \leq m, 1 \leq j \leq n} |a_{ij}|$$

定义的 $\|\cdot\|_{m_\infty}$ 是 $\mathbb{R}^{m \times n}$ 上的 (广义) 矩阵范数, 称为 l_∞ 范数。

实际上, 这个范数就是 $\text{vec}(\mathbf{A})$ 的 l_∞ 范数, 包括

例 3.1.20. 对任意 $\mathbf{A} \in \mathbb{R}^{m \times n}$, 由

$$\|\mathbf{A}\|_F := \left(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2 \right)^{\frac{1}{2}} = (\text{Tr}(\mathbf{A}^\top \mathbf{A}))^{\frac{1}{2}}$$

定义的 $\|\cdot\|_F$ 是 $\mathbb{R}^{m \times n}$ 上的矩阵范数, 称为 l_2 范数或 *Frobenius* 范数 (F 范数)。

实际上, 这个范数就是 $\text{vec}(\mathbf{A})$ 的 l_2 范数。

在数据科学中, 有时还用到 $p, q-$ 矩阵范数。

$$\|\mathbf{A}\|_{1,2} = \left(\sum_{j=1}^n \|\mathbf{a}_j\|_1^2 \right)^{\frac{1}{2}} = \left(\sum_{j=1}^n \left(\sum_{i=1}^m |A_{ij}| \right)^2 \right)^{\frac{1}{2}}$$

$$\|\mathbf{A}\|_{2,1} = \sum_{j=1}^n \|\mathbf{a}_j\|_2 = \sum_{j=1}^n \left(\sum_{i=1}^m |A_{ij}|^2 \right)^{\frac{1}{2}}$$

$$\|\mathbf{A}\|_{p,q} = \left(\sum_{j=1}^n \left(\sum_{i=1}^m |A_{ij}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}}$$

$$\|\mathbf{A}\|_* = \text{Tr}(\sqrt{\mathbf{A}^\top \mathbf{A}})$$

考虑到矩阵乘法的重要地位, 因此讨论矩阵范数时一般附加“相容性”条件。

定义 3.1.22. 若矩阵范数 $\|\cdot\|$ 满足:

$$\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\|, \text{ 对任意 } \mathbf{A} \in \mathbb{R}^{m \times p}, \mathbf{B} \in \mathbb{R}^{p \times n}$$

则称矩阵范数满足相容性条件。

不满足相容性条件的矩阵范数我们可以称其为广义矩阵范数。

例 3.1.21. $\|\cdot\|_{m_1}$ 满足相容性条件。

$$\|\mathbf{AB}\|_{m_1} \leq \|\mathbf{A}\|_{m_1} \|\mathbf{B}\|_{m_1}, \text{ 对任意 } \mathbf{A} \in \mathbb{R}^{m \times p}, \mathbf{B} \in \mathbb{R}^{p \times n}$$

例 3.1.22. $\|\cdot\|_F$ 满足相容性条件。

$$\|\mathbf{AB}\|_F \leq \|\mathbf{A}\|_F \|\mathbf{B}\|_F, \text{ 对任意 } \mathbf{A} \in \mathbb{R}^{m \times p}, \mathbf{B} \in \mathbb{R}^{p \times n}$$

例 3.1.23. $\|\cdot\|_{m_\infty}$ 不满足相容性条件。

证明. 取

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

那么

$$\|\mathbf{A}^2\|_{m_\infty} = \|2\mathbf{A}\|_{m_\infty} = 2 \not\leq \|\mathbf{A}\|_{m_\infty}^2 = 1$$

我们只需要对 $\|\cdot\|_{m_\infty}$ 做一点修改, 就可以使其满足相容性条件:

$$\|\mathbf{A}\|_{m_\infty} := n \max_{1 \leq i \leq m, 1 \leq j \leq n} |a_{ij}|$$

□

算子范数

由于在大多数与估计有关的问题中, 矩阵和向量会同时参与讨论, 所以希望引进一种矩阵的范数, 它是和向量范数相联系并且和向量范数相容的。

定义 3.1.23. 若矩阵范数 $\|\cdot\|_M$ 和向量范数 $\|\cdot\|_v$ 满足

$$\|\mathbf{Ax}\|_v \leq \|\mathbf{A}\|_M \|\mathbf{x}\|_v, \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{x} \in \mathbb{R}^n,$$

则称矩阵范数 $\|\cdot\|_M$ 与向量范数 $\|\cdot\|_v$ 是相容的。

对于给定的任意向量范数, 我们都可以如下构造一个与该向量范数相容的矩阵范数。

定义 3.1.24. 称 $m \times n$ 矩阵空间上如下定义的范数 $\|\cdot\|$ 为从属于向量范数 $\|\cdot\|_v$ 的矩阵范数, 也称其为由向量范数 $\|\cdot\|_v$ 诱导出的算子范数

$$\begin{aligned} \|\mathbf{A}\| &= \max\{\|\mathbf{Ax}\|_v : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_v = 1\} \\ &= \max\left\{\frac{\|\mathbf{Ax}\|_v}{\|\mathbf{x}\|_v} : \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq 0\right\} \end{aligned}$$

显然, 该矩阵范数和向量范数 $\|\cdot\|_v$ 是相容的。

因为, 对任意 $\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq 0$,

$$\frac{\|\mathbf{Ax}\|_v}{\|\mathbf{x}\|_v} \leq \max\left\{\frac{\|\mathbf{Ax}\|_v}{\|\mathbf{x}\|_v} : \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq 0\right\} = \|\mathbf{A}\|$$

所以 $\|\mathbf{Ax}\|_v \leq \|\mathbf{A}\|_M \|\mathbf{x}\|_v$ 。

我们有如下定理:

定理 3.1.6. 算子范数都满足相容性条件。

证明. 设矩阵范数 $\|\cdot\|$ 是由向量范数 $\|\cdot\|_v$ 诱导出的算子范数, $A \in \mathbb{R}^{m \times p}$, $B \in \mathbb{R}^{p \times n}$, $x \in \mathbb{R}^n$,

$$\|AB\| = \max_{\|x\|=1} \|ABx\|_v \leq \max_{\|x\|=1} \|A\| \|Bx\|_v = \|A\| \max_{\|x\|=1} \|Bx\|_v = \|A\| \|B\|$$

□

经常利用向量的 l_p -范数诱导出算子范数:

$$\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$

定理 3.1.7. 设 $A \in \mathbb{R}^{m \times n}$, $p = 1, \infty, 2$ 时, 向量的 l_p -范数诱导出的算子范数分别为

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|$$

$$\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|$$

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$$

当 $A = O$ 时, 以上三式显然成立。假定 $A \neq O$, 对以上的三个范数进行证明。

1 范数证明

证明. 对于 1 范数, 将给定的 $A \in \mathbb{R}^{m \times n}$ 按列分块 $A = [a_1, \dots, a_n]$, 并记 $\delta = \|a_{j_0}\|_1 = \max_{1 \leq j \leq n} \|a_j\|_1$, 则对任意满足 $\|x\|_1 = \sum_{i=1}^n |x_i| = 1$ 的 $x \in \mathbb{R}^n$, 有

$$\begin{aligned} \|Ax\|_1 &= \left\| \sum_{j=1}^n x_j a_j \right\| \leq \sum_{j=1}^n |x_j| \|a_j\|_1 \\ &\leq \sum_{j=1}^n |x_j| \max_{1 \leq j \leq n} \|a_j\|_1 = \|a_{j_0}\|_1 = \delta \end{aligned}$$

此处我们证明了 $\|A\|_1 := \max_{\|x\|_1=1} \|Ax\|_1 \leq \delta$ 。

此外, 令 x 为第 j_0 个元素为 1, 其余分量为 0 的向量 e_{j_0} , 则有 $\|e_{j_0}\|_1 = 1$, 而且

$$\|Ae_{j_0}\|_1 = \|a_{j_0}\|_1 = \delta$$

这样我们证明了存在满足 $\|x\|_1 = 1$ 的 x , 使得 $\|Ax\|_1 = \delta$ 。

因此有

$$\|A\|_1 = \max_{\|x\|_1=1} \|Ax\|_1 = \delta = \max_{1 \leq j \leq n} \|a_j\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|.$$

□

∞ 范数证明

证明. 对于 ∞ 范数, 记

$$\eta = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|,$$

则对任意满足 $\|\mathbf{x}\|_\infty = 1$ 的 $\mathbf{x} \in \mathbb{R}^n$, 有

$$\|\mathbf{Ax}\|_\infty = \max_{1 \leq i \leq m} \left| \sum_{j=1}^n a_{ij}x_j \right| \leq \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}| |x_j| \leq \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}| = \eta$$

此处我们证明了 $\|\mathbf{A}\|_\infty := \max_{\|\mathbf{x}\|_\infty=1} \|\mathbf{Ax}\|_\infty \leq \eta$ 。

设 \mathbf{A} 的第 k 行的元素的绝对值之和最大, 即 $\eta = \sum_{j=1}^n |a_{kj}|$ 。令

$$\tilde{\mathbf{x}} = (sgn(a_{k1}), \dots, sgn(a_{kn}))^\top$$

则 $\mathbf{A} \neq \mathbf{0}$ 蕴含 $\|\tilde{\mathbf{x}}\|_\infty = 1$, 有 $\|\mathbf{A}\tilde{\mathbf{x}}\|_\infty = \sum_{j=1}^n |a_{kj}| = \eta$ 。

这里证明了存在满足 $\|\mathbf{x}\|_\infty = 1$ 的 \mathbf{x} , 使得 $\|\mathbf{Ax}\|_\infty = \eta$ 。

则

$$\|\mathbf{A}\|_\infty = \eta = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|.$$

□

2 范数证明

证明. 对于 2 范数, 应有

$$\begin{aligned} \|\mathbf{A}\|_2 &= \max_{\|\mathbf{x}\|_2=1} \|\mathbf{Ax}\|_2 = \max_{\|\mathbf{x}\|_2=1} [(\mathbf{Ax})^\top \mathbf{Ax}]^{\frac{1}{2}} \\ &= \max_{\|\mathbf{x}\|_2=1} [\mathbf{x}^\top (\mathbf{A}^\top \mathbf{A}) \mathbf{x}]^{\frac{1}{2}} \end{aligned}$$

注意, $\mathbf{A}^\top \mathbf{A}$ 是半正定矩阵, 设其特征值为

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0,$$

以及其对应的正交规范特征向量为 $\mathbf{q}_1, \dots, \mathbf{q}_n \in \mathbb{R}^n$ 。

则对任一满足 $\|\mathbf{x}\|_2 = 1$ 的向量 $\mathbf{x} \in \mathbb{R}^n$ 有

$$\begin{aligned} \mathbf{x} &= \sum_{i=1}^n \alpha_i \mathbf{q}_i \\ \sum_{i=1}^n \alpha_i^2 &= 1 \end{aligned}$$

于是, 有

$$\mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} = \sum_{i=1}^n \lambda_i \alpha_i^2 \leq \lambda_1$$

这里我们证明了 $\|A\|_2 = \max_{\|\mathbf{x}\|_2=1} [\mathbf{x}^T (A^T A) \mathbf{x}]^{\frac{1}{2}} \leq \sqrt{\lambda_1}$ 。

另一方面，若取 $\mathbf{x} = \mathbf{q}_1$ ，则有

$$\mathbf{x}^T A^T A \mathbf{x} = \mathbf{q}_1^T A^T A \mathbf{q}_1 = \mathbf{q}_1^T \lambda_1 \mathbf{q}_1 = \lambda_1$$

这里我们证明了存在满足 $\|\mathbf{x}\|_2 = 1$ 的 \mathbf{x} ，使得 $\|A\mathbf{x}\|_2 = \sqrt{\lambda_1}$ 。

所以

$$\|A\|_2 = \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2 = \sqrt{\lambda_1} = \sqrt{\lambda_{\max}(A^T A)}$$

□

我们通常分别称矩阵的 1 范数、 ∞ 范数和 2 范数为列和范数、行和范数和谱范数。显然矩阵列和范数与行和范数容易计算，而矩阵的谱范数不易计算，它需要计算 $A^T A$ 的最大特征值，但是谱范数具有几个好的性质，使它在理论研究中很有用处。下面给出谱范数几个常用的性质。

定理 3.1.8. 设 $A \in \mathbb{R}^{n \times n}$ ，则

$$(1) \|A\|_2 = \max\{|\mathbf{y}^T A \mathbf{x}| : x, y \in \mathbb{C}^n, \|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = 1\};$$

$$(2) \|A^T\|_2 = \|A\|_2 = \sqrt{\|A^T A\|_2};$$

(3) 对于任意的正交矩阵 U 和 V 有， $\|U\mathbf{A}\|_2 = \|V\mathbf{A}\|_2 = \|A\|_2$ 。

例 3.1.24. 设矩阵 $A = \begin{pmatrix} 2 & -1 \\ -2 & 4 \end{pmatrix}$ ，求 $\|A\|_p$, ($p = 1, 2, \infty$) 以及 $\|A\|_F$

$$\|A\|_1 = \max\{2 + |-2|, |-1| + 4\} = 5$$

$$\|A\|_\infty = \max\{2 + |-1|, |-2| + 4\} = 6$$

$$\text{因为 } A^T A = \begin{pmatrix} 2 & -2 \\ -1 & 4 \end{pmatrix} \begin{pmatrix} 2 & -1 \\ -2 & 4 \end{pmatrix} = \begin{pmatrix} 8 & -10 \\ -10 & 17 \end{pmatrix} \text{ 由 } |\mathbf{I}\lambda - A^T A| = \begin{vmatrix} \lambda - 8 & 10 \\ 10 & \lambda - 17 \end{vmatrix} = 0 \text{ 解}$$

得 $\lambda_1 = 23.466, \lambda_2 = 1.534$ 故 $\|A\|_2 = \sqrt{23.466} = 4.844$

$$\|A\|_F = (2^2 + (-1)^2 + (-2)^2 + 4^2)^{\frac{1}{2}} = 5$$

算子范数的几何意义

例 3.1.25. 对应于 $p = 1, 2, \infty$ 三种向量范数的单位球面 $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^2 | \|\mathbf{x}\|_p = 1\}$ 在矩阵

$$A = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix}$$

作用下的效果分别为

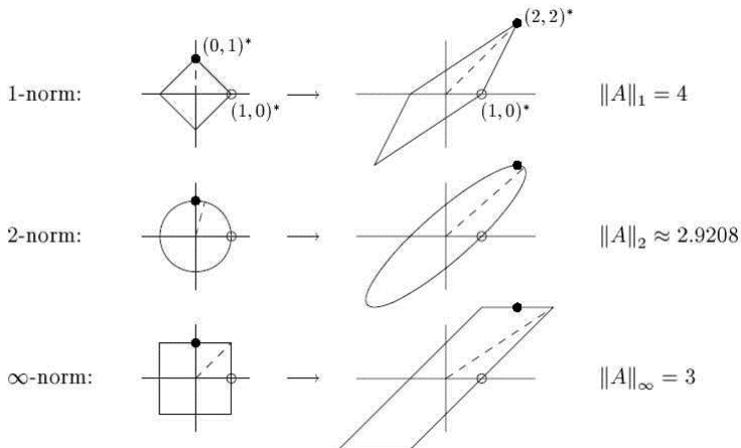


图 3.9: 不同向量范数下, 单位球面在矩阵作用下的变换。

3.1.7 范数在机器学习中的应用

在第一章中, 对于监督学习问题, 常常将其等价为求下列函数的最小值问题:

$$R_{srm}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f)$$

其中 y_i 是特征 x_i 的标签, 而 $f(x_i)$ 则是模型 f 对于特征 x_i 给出的一个预测值; $L(y_i, f(x_i))$ 是损失函数, 用于衡量单个样本预测值和真实值的误差; $\frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i))$ 是误差项(也称为代价函数), 误差项主要用来衡量输出的预测值和真实值之间的整体误差; $\lambda J(f)$ 是正则化项, 正则化项主要用于防止模型过拟合。

那么在监督学习中损失函数 L 和正则化函数 J 具有什么形式呢?

损失函数 L 的形式按照是否应用了距离度量, 一般可分为两种:

- 基于距离度量的损失
- 非距离度量形式的损失

基于距离度量的损失常利用向量空间中定义的距离函数, 度量两个向量间的差异。常见的距离相关损失函数有 0-1 损失函数、绝对损失函数、平方损失函数以及以上损失函数衍生出的各种变体。而非距离度量形式的损失常应用于概率模型。

结构风险中引入正则项的主要目的之一是防止过拟合。我们来具体看一下过拟合现象。

例 3.1.26. 考虑平面上有一系列点, 它们是由带有噪声的四次曲线产生的。我们分别用一次函数, 4 次多项式函数和 6 次多项式函数来拟合这些点。

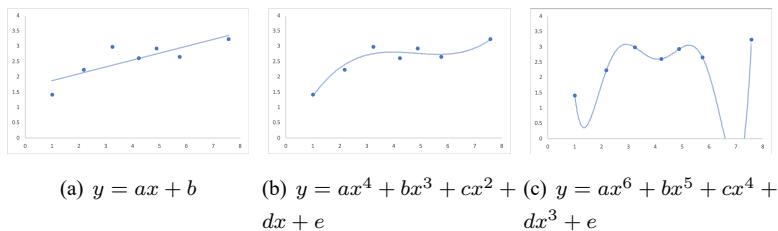


图 3.10: 欠拟合、正常拟合、过拟合

- 欠拟合的模型因为模型假设过于简单，如图3.10(a)，而无法反应数据的真实情况。
- 若增加模型的复杂性则可得一个合适的拟合，如图3.10(b)，从而能够很好地反应数据的分布和趋势。
- 若继续增加模型的复杂性就会产生过拟合的现象，如图3.10(c)。这种模型不仅仅拟合了数据，并且还拟合了噪音。这将使得模型在新数据上表现很差。

欠拟合问题易解决，但是过拟合，则需要通过其他一些手段——如正则化来解决。在实际中，正则化方法使得模型在训练误差很小的情况下，模型本身的复杂度也较小，即

$$\left\{ \begin{array}{l} \min Loss(\theta, \mathbf{X}, \mathbf{Y}) \\ \min J(\theta) \end{array} \right.$$

其中 $Loss(\theta, \mathbf{X}, \mathbf{Y})$ 是经验风险， $J(\theta)$ 是正则化项， θ 是模型的参数，可代表模型。通常我们将上述双目标函数优化任务转化为下列优化任务

$$\min Loss(\theta, \mathbf{X}, \mathbf{Y}) + \lambda J(\theta)$$

这个式子就是我们之前提到的结构风险， λ 用于调节经验风险和正则化项的关系。那么这个正则化项该如何选择呢？

正则化：范数的选择

- 在前面多项式拟合的例子中，想要避免过拟合，则需让模型不出现用更高次函数去拟合四次函数产生的带有噪声的数据的情况，高次函数拟合效果虽好但也拟合了数据噪声
- 我们应让模型尽可能表现为用待定系数的四次函数去拟合四次函数产生的带有噪声的数据。如果要让模型不出现过拟合情况，需要让模型参数向量 θ 中元素个数尽可能少，对于拟合四次函数的问题来说，模型求解得到的参数向量中元素个数应该是 4，而不是其它。
- 一个向量的元素个数正好是向量的 l_0 范数，因此让模型参数向量 θ 中元素的个数最小化其实等于优化 $\min \|\theta\|_0$ ，这样就建立了过拟合解决方案和范数的联系：

$$\min Loss(\theta, \mathbf{X}, \mathbf{Y}) + \lambda \|\theta\|_0$$

损失函数和正则化项中的范数：常用的向量范数

- l_0 范数（并不满足范数的定义）

$$\|\mathbf{x}\|_0 = \sum_{i=1}^n \mathcal{I}(x_i \neq 0)$$

- l_1 范数

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$$

- l_2 范数

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}}$$

l_0 范数指向量中非 0 的元素的个数，优化 l_0 范数可以得到一些稀疏解。选择 l_0 范数有很多优势，比如稀疏解存储成本低、自动实现特征选择 (Feature Selection)、可解释性 (Interpretability) 强。但是 l_0 范数很难优化求解，是个 NP-hard 问题。

l_1 范数是 l_0 范数的最优凸近似，而且它比 l_0 范数要容易优化求解，所以 l_1 范数被称为“稀疏规则算子” (Lasso)。它常常可以应用在稀疏编码、特征选择和压缩感知中。

在一定条件下，以

$$\begin{array}{ccc} \min \|\mathbf{x}\|_1 & \xleftrightarrow{\text{概率 1 意义下等价}} & \min \|\mathbf{x}\|_0 \\ s.t. \mathbf{Ax} = \mathbf{b} & & s.t. \mathbf{Ax} = \mathbf{b} \end{array}$$

l_1 范数和 l_0 范数都可以实现稀疏， l_1 因具有比 l_0 更好的优化求解特性而被广泛应用。

l_2 范数又称“岭回归” (Ridge Regression) 或“权值衰减 (weight decay)”，最小化 l_2 范数，可以使得参数向量的元素值都很小，大都接近于 0。它的好处是可以改善过拟合、易于优化。

关于“过拟合”：在数学上称为“病态” (ill-condition)：即函数的输入改变一点点，输出却改变非常大。 l_2 范数限制了参数都很小，实际上就限制了多项式各分量的影响很小，一定程度上避免了模型出现“病态”的情况。

与 l_2 范数相比， l_1 范数更有可能得到值为 0 的解，所以导致稀疏。 l_2 范数得到的解在各个分量上更为均衡。

l_1 范数存在不可导点，这导致了在这一点上无法进行有效地优化。而 l_2 范数是光滑的。

但正因为如此， l_2 范数鲁棒性更差一些。因为对于异常值 l_2 范数倾向于把它的影响变得更大。

损失函数和正则化项中的范数：常用的矩阵范数

F 范数

$$\|\mathbf{A}\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2 \right)^{\frac{1}{2}} = \sqrt{\text{Tr}(\mathbf{A}^\top \mathbf{A})}$$

- 最常用的矩阵范数为 F 范数。最小化矩阵的 F 范数，会使得矩阵的每个元素都很小，接近于 0
- $\|A - B\|_F$ 可度量矩阵 A, B 之间的差异，最小化可使得两者尽可能的相等

p 范数

$$\|\mathbf{A}\|_p = \|\text{vec}(\mathbf{A})\|_p = \left(\sum_{j=1}^n \sum_{i=1}^m |A_{ij}|^p \right)^{\frac{1}{p}}$$

- 当 $p = 1$ 时，为矩阵 l_1 范数，最小化 $\|\mathbf{A}\|_1$ 能让矩阵 \mathbf{A} 元素稀疏。稀疏矩阵的优点：计算速度更快、存储成本低、可解释性强（例如：文本分类中，可知哪些词对类别起重要作用）
- 当 $p = 2$ 时，为矩阵 l_2 范数，也即 F 范数

$\|\mathbf{A}\|_{2,1}$ 和 $\|\mathbf{A}\|_{1,2}$ 范数的含义

$$\|\mathbf{A}\|_{2,1} = \sum_{j=1}^n \|\mathbf{a}_j\|_2 = \sum_{j=1}^n \left(\sum_{i=1}^m |A_{ij}|^2 \right)^{\frac{1}{2}}$$

最小化 $\|\mathbf{A}\|_{2,1}$ 范数能让矩阵 \mathbf{A} 不同行之间（列向量）稀疏，在机器学习领域这属于 Group Lasso。应用于文本分类领域：Lasso 对应于找出关键词，Group Lasso 找出关键句子，Hierarchical Lasso 找出关键段。

$$\|\mathbf{A}\|_{1,2} = \left(\sum_{j=1}^n \|\mathbf{a}_j\|_1^2 \right)^{\frac{1}{2}} = \left(\sum_{j=1}^n \left(\sum_{i=1}^m |A_{ij}| \right)^2 \right)^{\frac{1}{2}}$$

最小化 $\|\mathbf{A}\|_{1,2}$ 范数能让矩阵行内元素互斥，也即行内存 0 元素但不能全为 0。这可以应用在特征选择的时候不同的类别可以选择互斥的特征。

核范数

$$\|\mathbf{A}\|_* = \text{Tr}(\sqrt{\mathbf{A}^T \mathbf{A}})$$

- 核范数指矩阵奇异值的和，英文为 Nuclear norm
- 最小化核范数 $\|\cdot\|_*$ 可以导致低秩矩阵 (Low-Rank)。低秩矩阵的应用：矩阵填充 (Matrix Completion)，例如：推荐系统、鲁棒 PCA、背景建模、变换不变低秩纹理 (TIIT)

从优化或者数值计算的角度来说，范数有助于处理“病态”的问题或条件书不好的问题，与方程组解的敏感性相关。我们在 5.1 节会进一步介绍。

3.2 正交与投影

在数据科学的许多工程应用（如信号降噪滤波、数据降维、主成分分析、时间序列分析）中，许多问题的最优求解都可归结为数据在某个子空间的投影问题。图3.11展示了将三维空间中的向量投影到二维平面上。本节我们介绍投影这一在数据分析中极为重要的数学工具。

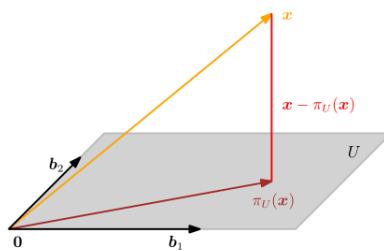


图 3.11: 将三维空间中的向量投影到二维平面上

3.2.1 矩阵的四个基本子空间

为了更好的理解子空间与投影，我们先讨论四个基本子空间：

1. 列空间: $\text{Col}(\mathbf{A})$
2. 行空间: $\text{Row}(\mathbf{A}) = \text{Col}(\mathbf{A}^T)$
3. 零空间: $\text{Null}(\mathbf{A})$
4. 左零空间: $\text{Null}(\mathbf{A}^T)$

四个基本子空间也是线性代数中非常重要的概念。为方便叙述，对于矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ ，其 m 个行向量、 n 个列向量分别记作

$$\begin{aligned} \mathbf{r}_1 &= [a_{11}, a_{12}, \dots, a_{1n}]^T \\ \mathbf{r}_2 &= [a_{21}, a_{22}, \dots, a_{2n}]^T \\ &\vdots \\ \mathbf{r}_m &= [a_{m1}, a_{m2}, \dots, a_{mn}]^T \end{aligned} \quad \mathbf{a}_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix}, \mathbf{a}_2 = \begin{pmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{pmatrix}, \dots, \mathbf{a}_n = \begin{pmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{pmatrix}$$

即 $\mathbf{A} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m)^T = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)$ 。

定义 3.2.1. 列空间是其列向量 $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\}$ 的所有线性组合的集合，它是 \mathbb{R}^m 的一个子空间，用符号 $\text{Col}(\mathbf{A})$ 表示，即有

$$\text{Col}(\mathbf{A}) = \left\{ \mathbf{y} \in \mathbb{R}^m \mid \mathbf{y} = \sum_{j=1}^n \alpha_j \mathbf{a}_j, \alpha_j \in \mathbb{R} \right\} = \text{span}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\} \quad (3.4)$$

定义 3.2.2. 行空间是其行向量 $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m\}$ 的所有线性组合的集合，它是 \mathbb{R}^n 的一个子空间，用符号 $\text{Row}(\mathbf{A})$ 表示，也可以用 $\text{Col}(\mathbf{A}^T)$ 表示，有

$$\text{Row}(\mathbf{A}) = \text{Col}(\mathbf{A}^T) = \left\{ \mathbf{y} \in \mathbb{R}^n \mid \mathbf{y} = \sum_{i=1}^m \beta_i \mathbf{r}_i, \beta_i \in \mathbb{R} \right\} = \text{span}\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m\} \quad (3.5)$$

定义 3.2.3. 零空间是所有满足齐次线性方程组 $\mathbf{Ax} = \mathbf{0}$ 的解向量集合，它是 \mathbb{R}^n 的一个子空间，用符号 $\text{Null}(\mathbf{A})$ 表示，即有

$$\text{Null}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{Ax} = \mathbf{0}\} \quad (3.6)$$

定义 3.2.4. 左零空间是所有满足齐次线性方程组 $\mathbf{A}^T \mathbf{y} = \mathbf{0}$ 的解向量集合，它是 \mathbb{R}^m 的一个子空间，用符号 $\text{Null}(\mathbf{A}^T)$ 表示，即有

$$\text{Null}(\mathbf{A}^T) = \{\mathbf{y} \in \mathbb{R}^m | \mathbf{A}^T \mathbf{y} = \mathbf{0}\} \quad (3.7)$$

给定一个矩阵，为了获得其四个基本子空间，我们需要用到以下结论：

定理 3.2.1. 1. 一系列初等行变换不改变矩阵的行空间。

2. 一系列初等行变换不改变矩阵的零空间。

3. 一系列初等列变换不改变矩阵的列空间。

4. 一系列初等列变换不改变矩阵的左零空间。

证明. [1] 容易验证，任何一种初等行变换都不改变行空间

- 对于 I 型初等行变换（用非零常数乘某一行）：

$$\text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_i, \dots, \mathbf{r}_m\} = \text{span}\{\mathbf{r}_1, \dots, c\mathbf{r}_i, \dots, \mathbf{r}_m\}$$

- 对于 II 型初等行变换（某一行的 c 倍加到另一行）：

$$\text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_i + c\mathbf{r}_j, \dots, \mathbf{r}_j, \dots, \mathbf{r}_m\} = \text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_i, \dots, \mathbf{r}_j, \dots, \mathbf{r}_m\}$$

对任意 $\mathbf{y} \in \text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_i, \dots, \mathbf{r}_j, \dots, \mathbf{r}_m\}$ 存在 β_1, \dots, β_m ，使得

$$\mathbf{y} = \beta_1 \mathbf{r}_1 + \dots + \beta_i \mathbf{r}_i + \dots + \beta_j \mathbf{r}_j + \dots + \beta_m \mathbf{m}$$

$$= \beta_1 \mathbf{r}_1 + \dots + \beta_i (\mathbf{r}_i + c\mathbf{r}_j) + \dots + (\beta_j - c\beta_i) \mathbf{r}_j + \dots + \beta_m \mathbf{m}$$

可以推出 $\mathbf{y} \in \text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_i + c\mathbf{r}_j, \dots, \mathbf{r}_j, \dots, \mathbf{r}_m\}$

- 对于 III 型初等行变换（互换矩阵中两行的位置）：

$$\text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_i, \dots, \mathbf{r}_j, \dots, \mathbf{r}_m\} = \text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_j, \dots, \mathbf{r}_i, \dots, \mathbf{r}_m\}$$

[2] 令 \mathbf{E}_i 是对应于矩阵 \mathbf{A} 的第 i 次初等行变换的初等矩阵。由初等行变换可逆。于是，

$$\mathbf{Bx} = (\mathbf{E}_k \mathbf{E}_{k-1} \cdots \mathbf{E}_1 \mathbf{A})\mathbf{x} = \mathbf{0} \Leftrightarrow \mathbf{Ax} = \mathbf{0}$$

即齐次线性方程 $\mathbf{Bx} = \mathbf{0}$ 与 $\mathbf{Ax} = \mathbf{0}$ 具有相同的解向量，从而 \mathbf{A} 经过若干次初等行变换后得到的矩阵 \mathbf{B} 与 \mathbf{A} 具有相同的零空间。

命题 [3] 和命题 [4] 也可以用类似的方法证明。□

例 3.2.1. 求 3×3 矩阵

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 1 \\ -1 & -1 & 1 \\ 1 & 4 & 5 \end{pmatrix}$$

的行空间、列空间、零空间和左零空间。

解. 依次进行初等列变换, 得到列简约阶梯型矩阵:

$$\left(\begin{array}{ccc} 1 & 2 & 1 \\ -1 & -1 & 1 \\ 1 & 4 & 5 \end{array} \right) \xrightarrow{\substack{C_2-2C_1 \\ C_3-C_1}} \left(\begin{array}{ccc} 1 & 0 & 0 \\ -1 & 1 & 2 \\ 1 & 2 & 4 \end{array} \right) \xrightarrow{\substack{C_1+C_2 \\ C_3-2C_2}} \mathbf{A}_C = \left(\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 2 & 0 \end{array} \right)$$

由此得到两个线性无关的列向量 $\mathbf{c}_1 = (1, 0, 3)^T, \mathbf{c}_2 = (0, 1, 2)^T$, 它们是列空间 $Col(\mathbf{A})$ 的基

$$Col(\mathbf{A}) = \text{span}\{(1, 0, 3)^T, (0, 1, 2)^T\}$$

由于一系列初等列变换不改变左零空间, 根据 \mathbf{A}_C , 知 $-3r_1 - 2r_2 + r_3 = 0$ 。

那么我们就可以根据 \mathbf{A}_C 的主元位置, 矩阵 \mathbf{A} 的主元行是第 1 行和第 2 行, 即行空间 $Col(\mathbf{A}^T)$ 可以写作

$$Col(\mathbf{A}^T) = \text{span}\{(1, 2, 1)^T, (-1, -1, 1)^T\}$$

对 \mathbf{A} 进行行初等变换

$$\left(\begin{array}{ccc} 1 & 2 & 1 \\ -1 & -1 & 1 \\ 1 & 4 & 5 \end{array} \right) \xrightarrow{\substack{R_3-R_1 \\ R_2+R_1}} \left(\begin{array}{ccc} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 2 & 4 \end{array} \right) \xrightarrow{\substack{R_3-2R_2 \\ R_1-2R_2}} \mathbf{A}_R = \left(\begin{array}{ccc} 1 & 0 & -3 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{array} \right)$$

\mathbf{A} 的秩为 2。解方程组 $\mathbf{A}_R \mathbf{x} = \mathbf{0}$ 得到 $\mathbf{x} = k(3, -2, 1)^T$

$$Null(\mathbf{A}) = \text{span}\{(3, -2, 1)^T\}$$

所以零空间维数为 1。

类似地, 我们求解 $\mathbf{A}_C^T \mathbf{x} = \mathbf{0}$ 得到 $\mathbf{x} = k(3, 2, -1)^T$ 所以

$$Null(\mathbf{A}^T) = \text{span}\{(3, 2, -1)^T\}$$

左零空间的维数也是 1。

四个基本子空间的基

我们接下来的目标是: 求四个基本子空间的基和维数。线性代数的课程中我们学习过矩阵的行秩等于列秩。我们有如下定理:

定理 3.2.2. 设 $\mathbf{A} \in \mathbb{R}^{m \times n}$, 则 $\dim(Col(\mathbf{A})) = \dim(Row(\mathbf{A})) = \text{rank}(\mathbf{A})$

定理 3.2.3. 设 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 则

$$\dim(Null(\mathbf{A})) = n - \text{rank}(\mathbf{A})$$

证明. 我们令 $r = \text{rank}(\mathbf{A})$, 根据定义 $\text{Null}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{Ax} = \mathbf{0}\}$ 我们对 \mathbf{A} 做行初等变换并交换其中的一些列, 将 \mathbf{A} 可以变换为

$$\mathbf{A}' = \begin{pmatrix} \mathbf{I} & \mathbf{B} \\ \mathbf{O} & \mathbf{O} \end{pmatrix} = \begin{pmatrix} 1 & b_{11} & b_{12} & \dots & b_{1,n-r} \\ & 1 & b_{21} & b_{22} & \dots & b_{2,n-r} \\ & \ddots & \vdots & \vdots & & \vdots \\ & & 1 & b_{r1} & b_{r2} & \dots & b_{r,n-r} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

显然 $\mathbf{A}'\mathbf{x} = \mathbf{0}$ 有以下 $n - r$ 个解

$$\mathbf{x}^{(1)} = \begin{pmatrix} b_{11} \\ \vdots \\ b_{r1} \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \mathbf{x}^{(2)} = \begin{pmatrix} b_{12} \\ \vdots \\ b_{r2} \\ 0 \\ -1 \\ \vdots \\ 0 \end{pmatrix}, \dots, \mathbf{x}^{(n-r)} = \begin{pmatrix} b_{1,n-r} \\ \vdots \\ b_{r,n-r} \\ 0 \\ 0 \\ \vdots \\ -1 \end{pmatrix}$$

并且容易看出向量组 $(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n-r)})$ 是一个极大线性无关组。

再注意到, 如果 \mathbf{x} 是方程 $\mathbf{A}'\mathbf{x} = \mathbf{0}$ 的解, 那么当 $x_{r+1}, x_{r+2}, \dots, x_n$ 取定时, 可以唯一确定 \mathbf{x} 。换句话说 $\{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}'\mathbf{x} = \mathbf{0}\}$ 的维数最大为 $n - r$ 。

综上 $\mathbf{A}'\mathbf{x} = \mathbf{0}$ 解空间的维数为 $n - r$, 即 $\mathbf{Ax} = \mathbf{0}$ 解空间的维数为 $n - r$, 即

$$\dim(\text{Null}(\mathbf{A})) = n - r$$

□

上述的证明过程实际上也就是我们刚刚求解矩阵 \mathbf{A} 零空间 $\text{Null}(\mathbf{A})$ 基底和维数的过程。由此得到秩定理, 描述了矩阵的秩与其零空间维数之间的关系。

定理 3.2.4. 矩阵 $\mathbf{A}_{m \times n}$ 的列空间和行空间的维数相等。这个共同的维数就是矩阵 \mathbf{A} 的秩 $\text{rank}(\mathbf{A})$, 它与零空间维数之间有下列关系:

$$\dim(\text{Col}(\mathbf{A})) + \dim(\text{Null}(\mathbf{A})) = n \quad (3.8)$$

利用上述定理我们立刻可以得到以下推论

推论 3.2.1. 设 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 则

$$\dim(\text{Null}(\mathbf{A}^T)) = m - \text{rank}(\mathbf{A})$$

3.2.2 四个基本子空间的正交性

在子空间分析中，两个子空间之间的关系由这两个子空间的元素（即向量）之间的关系刻画。我们将继续讨论四个基本子空间之间的关系。设 $A \in \mathbb{R}^{m \times n}$, A 的四个基本子空间中, $\text{Col}(A), \text{Null}(A^T)$ 都是 \mathbb{R}^m 的子空间, 它们是否有交集? $\text{Col}(A^T), \text{Null}(A)$ 都是 \mathbb{R}^n 的子空间, 它们是否有交集?

定理 3.2.5. 设 $A \in \mathbb{R}^{m \times n}$,

$$\text{Col}(A) \cap \text{Null}(A^T) = \{\mathbf{0}\}$$

$$\text{Col}(A^T) \cap \text{Null}(A) = \{\mathbf{0}\}$$

证明. 设 $v \in \text{Col}(A^T) \cap \text{Null}(A)$, 即 v 在 $A = (r_1, r_2, \dots, r_m)^T$ 的行空间中且 $Av = \mathbf{0}$ 。

设 $v = a_1r_1 + a_2r_2 + \dots + a_mr_m$, 则

$$Av = \mathbf{0} \implies r_1^T v = 0, \dots, r_m^T v = 0 \implies v^T v = 0 \implies v = \mathbf{0}$$

即

$$\text{Col}(A^T) \cap \text{Null}(A) = \{\mathbf{0}\}.$$

同理 $\text{Col}(A) \cap \text{Null}(A^T) = \{\mathbf{0}\}$ 。□

定义 3.2.5. 设 S 和 T 是 \mathbb{R}^n 的两个子空间。如果

$$S \cap T = \{\mathbf{0}\}$$

我们称 S 和 T 无交连。

列空间和左零空间是无交连的, 行空间和零空间是无交连的。

定义 3.2.6. 设 S 和 T 是 \mathbb{R}^n 的两个子空间。如果对于 $\forall v \in S, \forall w \in T$, 均有

$$v^T w = 0$$

我们说 S 垂直于 T , T 垂直于 S , 记做 $S \perp T, T \perp S$ 。

或者说, 子空间 S 和子空间 T 是正交的。

定理 3.2.6. 正交的两个子空间必定是无交连的。

证明. 假设 \mathbb{R}^n 中的两个子空间 S, T 不是无交连的

则

$$\exists v \neq \mathbf{0}, v \in S \cap T$$

而

$$v^T v \neq 0$$

因而 S 和 T 不正交。从而正交的两个子空间必是无交连的。□

显然，无交连的子空间不一定是正交的。如 $\text{span}\{(1, 1)^T\}$ 和 $\text{span}\{(1, 0)^T\}$ 。那么列空间和左零空间，行空间和零空间是正交的么？

例 3.2.2. 设 A 是 $m \times n$ 阶阵，则 $\text{Col}(A)$ 和 $\text{Null}(A^T)$ 正交， $\text{Col}(A^T)$ 和 $\text{Null}(A)$ 正交。

证明. 对 $\forall v \in \text{Null}(A^T)$ ，则

$$v^T A = \mathbf{0} \implies v^T a_1 = 0, v^T a_2 = 0, \dots, v^T a_n = 0$$

对 $\forall w \in \text{Col}(A)$ ，有 $w = \alpha_1 a_1 + \alpha_2 a_2 + \dots + \alpha_n a_n$:

$$v^T w = \alpha_1 v^T a_1 + \alpha_2 v^T a_2 + \dots + \alpha_n v^T a_n = 0$$

因此， $\text{Null}(A^T) \perp \text{Col}(A)$ ， $\text{Col}(A)$ 和 $\text{Null}(A^T)$ 正交。

将 A 换成 A^T ，我们得到 $\text{Col}(A^T) \perp \text{Null}(A)$ ， $\text{Col}(A^T)$ 和 $\text{Null}(A)$ 正交。□

相对于正交，正交补是两个子空间更强的一种关系。

定义 3.2.7. 设 $V \subset \mathbb{R}^n$ 是一个子空间， V 在 \mathbb{R}^n 中的正交补定义为集合

$$\{w \in \mathbb{R}^n | v^T w = 0, \forall v \in V\}$$

记作 V^\perp 。

也就是说 V 的正交补空间是 \mathbb{R}^n 中所有和 V 正交的向量构成的集合。

显然一个空间和它的正交补空间是正交的，即 $V \perp V^\perp$ 。

显然 V 与 V^\perp 的和是直和，因此，对于 \mathbb{R}^n 中的任意向量 x 可以唯一的分解成如下形式：

$$x = x_1 + x_2$$

其中 $x_1 \in V, x_2 \in V^\perp$ 并且 $x_1^T x_2 = 0$ 。这种分解形式叫做向量的正交分解。

定理 3.2.7. 证明： $\text{Col}(A^T)^\perp = \text{Null}(A)$ ， $\text{Col}(A)^\perp = \text{Null}(A^T)$ 。

证明. 我们已经知道， $\text{Col}(A^T)$ 和 $\text{Null}(A)$ 是正交的，也就是说

$$\text{Null}(A) \subseteq \text{Col}(A^T)^\perp$$

对 $\forall x \in \text{Col}(A^T)^\perp$ ， x 和 $\text{Col}(A^T)$ 中的任意向量正交，那么：

$$x^T r_1 = 0, x^T r_2 = 0, \dots, x^T r_m = 0$$

即 $Ax = \mathbf{0}$ 。说明 $x \in \text{Null}(A)$ 。也即

$$\text{Col}(A^T)^\perp \subseteq \text{Null}(A)$$

因此 $\text{Col}(A^T)^\perp = \text{Null}(A)$ 。同样可以证明 $\text{Col}(A)^\perp = \text{Null}(A^T)$ 。□

顾名思义，子空间 V 在向量空间 \mathbb{R}^n 的正交补空间 V^\perp 含有正交和补充双重含义：

1. 子空间 V^\perp 与 V 正交：

2. 向量空间 \mathbb{R}^n 是子空间 V 与 V^\perp 的直和, 即 $\mathbb{R}^n = V \oplus V^\perp$ 。这表明, 向量空间 \mathbb{R}^n 是由子空间 V 补充 V^\perp 而成。

正交补空间是一个比正交子空间更严格的概念: 当向量空间 \mathbb{R}^n 和子空间 V 给定之后, 和 V 正交的空间不一定是唯一的, 但是 V 的正交补 V^\perp 是唯一的。

我们将本节内容总结成线性代数基本定理, 图3.12展示了四个基本子空间的关系。

定理 3.2.8. (线性代数基本定理) 若 A 是 $m \times n$ 矩阵,

- 1 [正交角度] $\text{Col}(A^T) \perp \text{Null}(A)$, $\text{Col}(A) \perp \text{Null}(A^T)$,
- 2 [扩张角度] $\text{Col}(A^T) \oplus \text{Null}(A) = \mathbb{R}^n$, $\text{Col}(A) \oplus \text{Null}(A^T) = \mathbb{R}^m$,
- 3 [维数角度] $\dim \text{Col}(A^T) + \dim \text{Null}(A) = n$, $\dim \text{Col}(A) + \dim \text{Null}(A^T) = m$.

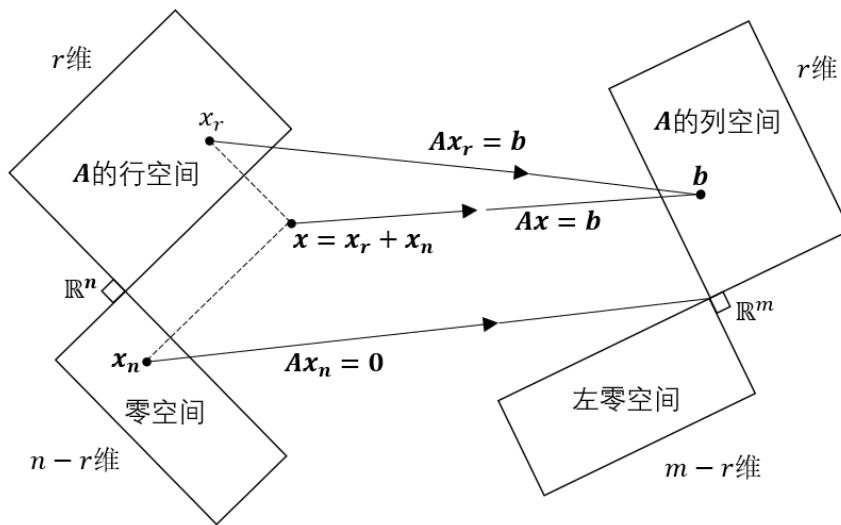


图 3.12: 四个子空间

3.2.3 正交投影

投影是一类重要的线性变换。投影在图形学、编码理论、统计和机器学习中起着重要作用。在机器学习中, 我们经常处理高维数据。高维数据通常很难分析或可视化。但是, 高维数据通常具有以下属性: 只有少数维包含大多数信息, 而其它大多数维对于描述数据的关键属性也不是必需的。当我们压缩或可视化高维数据时, 我们将丢失信息。为了最大程度地减少这种压缩损失, 我们理想地希望在数据中找到最有用的信息维度。然后, 我们可以将原始的高维数据投

影到低维特征空间上，并在此低维空间中进行操作，以了解有关数据集的更多信息并提取模式。例如机器学习中主成分分析（PCA）、深度学习中深度自动编码器大量采用了降维的想法。

定义 3.2.8. 设 \mathbb{V} 是一向量空间, $\mathbb{U} \subseteq \mathbb{V}$ 是 \mathbb{V} 的一个子空间。如果线性映射 $\pi: \mathbb{V} \rightarrow \mathbb{U}$ 满足

$$\pi^2 = \pi \circ \pi = \pi$$

则称 π 为投影。

设 π 对应的矩阵 P_π , 显然 P_π 满足 $P_\pi^2 = P_\pi$, 称 P_π 为投影矩阵。

正如阳光照出人的影子, 如果我们按照影子的大小做个假人摆在影子的地方, 那么这个假人的影子和原来的影子是一样的。投影包括中心投影, 斜投影和正交投影。本节, 我们主要关注正交投影。

定义 3.2.9. 给定定义了标准内积和欧氏距离的向量空间 \mathbb{R}^n 中的向量 \mathbf{x} , \mathbb{U} 是 \mathbb{R}^n 的子空间, 求 $\mathbf{y} \in \mathbb{U}$, 使得 $\|\mathbf{y} - \mathbf{x}\|$ 最小, 即

$$\pi_{\mathbb{U}}(\mathbf{x}) = \arg \min_{\mathbf{y} \in \mathbb{U}} \|\mathbf{y} - \mathbf{x}\|,$$

称向量 \mathbf{y} 为向量 \mathbf{x} 在子空间 \mathbb{U} 的正交投影。

因为可以对 \mathbf{x} 正交分解, $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$, 其中 $\mathbf{x}_1 \in \mathbb{U}$, $\mathbf{x}_2 \in \mathbb{U}^\perp$ 。所以

$$\|\mathbf{y} - \mathbf{x}\|^2 = \|\mathbf{y} - (\mathbf{x}_1 + \mathbf{x}_2)\|^2 = \|(\mathbf{x}_1 - \mathbf{y}) + \mathbf{x}_2\|^2.$$

而 $\mathbf{x}_1 - \mathbf{y} \in \mathbb{U}$, $\mathbf{x}_2 \in \mathbb{U}^\perp$, 所以 $\|(\mathbf{x}_1 - \mathbf{y}) + \mathbf{x}_2\|^2 = \|\mathbf{x}_1 - \mathbf{y}\|^2 + \|\mathbf{x}_2\|^2$ 。所以我们只需令 $\mathbf{y} = \mathbf{x}_1$ 即可, 那么 $\mathbf{x}_2 = \mathbf{x} - \mathbf{y} = \mathbf{x} - \pi_{\mathbb{U}}(\mathbf{x}) \in \mathbb{U}^\perp$ 。

投影到 1 维子空间

接下来, 我们看一下如何寻找一个投影矩阵 P_π 使得向量投影到某个 1 维子空间上。

假设给定 \mathbb{R}^n 中一条通过原点的直线 (1 维子空间), 其具有基向量 \mathbf{b} , 相应的基底矩阵表示为 $\mathbf{B} = [\mathbf{b}]$, 也就是说这组基中仅有一个向量。

这条直线是由 \mathbf{b} 张成的一维子空间 $\mathbb{U} = \text{Col}(\mathbf{B}) \subseteq \mathbb{R}^n$ 。

假设 $\mathbf{x} \in \mathbb{R}^n$, 当把 \mathbf{x} 投影到 \mathbb{U} 时, 我们想寻找一个点 $\pi_{\mathbb{U}}(\mathbf{x}) \in \mathbb{U}$ 最接近 \mathbf{x} , 即

$$\pi_{\mathbb{U}}(\mathbf{x}) = \arg \min_{\mathbf{y} \in \mathbb{U}} \|\mathbf{y} - \mathbf{x}\|$$

因为 $\pi_{\mathbb{U}}(\mathbf{x}) \in \mathbb{U}$, 又 $\mathbb{U} = \text{Col}(\mathbf{B}) = \text{span}\{\mathbf{b}\}$, 所以 $\pi_{\mathbb{U}}(\mathbf{x}) = \lambda \mathbf{b}$, $\lambda \in \mathbb{R}$ 。

我们将结合 $\mathbf{x} - \pi_{\mathbb{U}}(\mathbf{x}) \in \mathbb{U}^\perp$, 逐步确定坐标 λ , 投影 $\pi_{\mathbb{U}}(\mathbf{x}) \in \mathbb{U}$ 和 $\pi_{\mathbb{U}}$ 的投影矩阵 P_π 。

1. 确定 λ 因为 $\pi_{\mathbb{U}}(\mathbf{x}) \in \text{Col}(\mathbf{B})$ 是 \mathbf{x} 的投影, 所以 $\mathbf{x} - \pi_{\mathbb{U}}(\mathbf{x}) \in \text{Col}(\mathbf{B})^\perp = \text{Null}(\mathbf{B}^T)$, 有

$$\mathbf{b}^T(\mathbf{x} - \pi_{\mathbb{U}}(\mathbf{x})) = 0 \iff \mathbf{b}^T \mathbf{x} - \lambda \mathbf{b}^T \mathbf{b} = 0$$

从而

$$\lambda = \frac{\mathbf{b}^T \mathbf{x}}{\mathbf{b}^T \mathbf{b}}$$

或者利用内积和范数表示可得

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

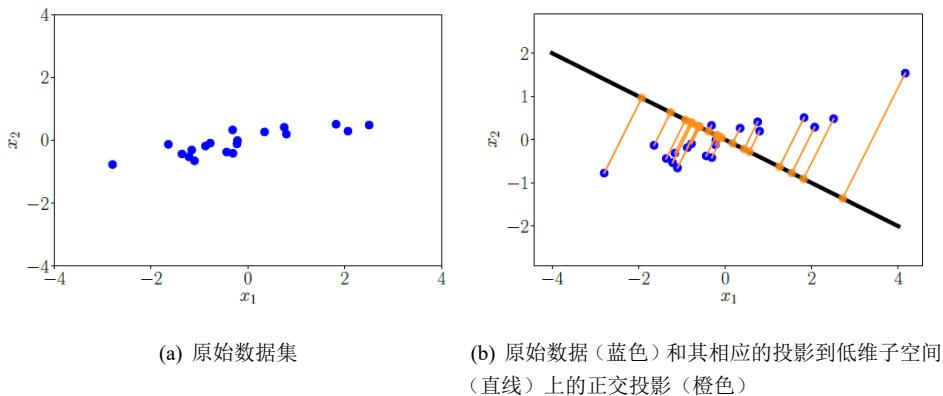


图 3.13: 将 2 维空间的点投影到 1 维子空间上。

2. 确定 $\pi_{\mathbb{U}}(\mathbf{x})$ 因为 $\pi_{\mathbb{U}}(\mathbf{x}) = \lambda \mathbf{b}$, 由上面的结论可得:

$$\pi_{\mathbb{U}}(\mathbf{x}) = \frac{\langle \mathbf{x}, \mathbf{b} \rangle}{\|\mathbf{b}\|^2} \mathbf{b} = \frac{\mathbf{b}^T \mathbf{x}}{\|\mathbf{b}\|^2} \mathbf{b}$$

我们可以给出 $\pi_{\mathbb{U}}(\mathbf{x})$ 的长度

$$\begin{aligned} \|\pi_{\mathbb{U}}(\mathbf{x})\| &= \|\lambda \mathbf{b}\| = |\lambda| \|\mathbf{b}\| \\ &= |\cos \omega| \|\mathbf{x}\| \|\mathbf{b}\| \frac{\|\mathbf{b}\|}{\|\mathbf{b}\|^2} \\ &= |\cos \omega| \|\mathbf{x}\| \end{aligned}$$

其中 ω 是 \mathbf{x} 和 \mathbf{b} 之间的夹角, $\cos \omega = \frac{\mathbf{b}^T \mathbf{x}}{\|\mathbf{b}\| \|\mathbf{x}\|}$ 。

3. 确定投影矩阵 \mathbf{P}_{π} 投影矩阵 \mathbf{P}_{π} 是投影 $\pi_{\mathbb{U}}(\mathbf{x})$ 对应的变换矩阵, 那么就有 $\pi_{\mathbb{U}}(\mathbf{x}) = \mathbf{P}_{\pi} \mathbf{x}$, 则有

$$\pi_{\mathbb{U}}(\mathbf{x}) = \lambda \mathbf{b} = \mathbf{b} \lambda = \mathbf{b} \frac{\mathbf{b}^T \mathbf{x}}{\|\mathbf{b}\|^2} = \frac{\mathbf{b} \mathbf{b}^T}{\|\mathbf{b}\|^2} \mathbf{x}$$

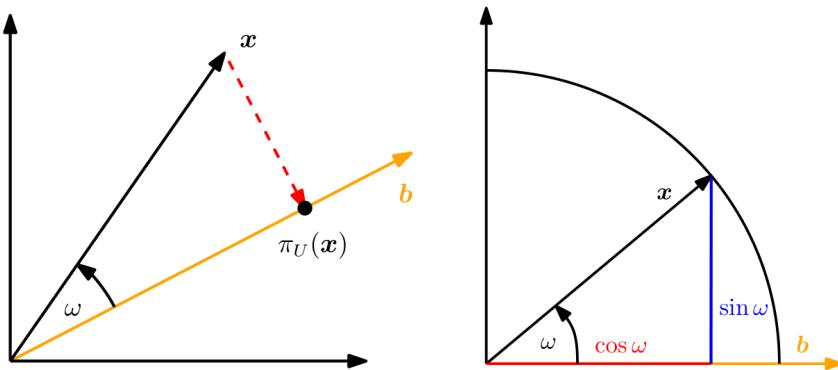
我们立刻可以看出

$$\mathbf{P}_{\pi} = \frac{\mathbf{b} \mathbf{b}^T}{\|\mathbf{b}\|^2}$$

例 3.2.3. 确定投影到 \mathbb{R}^3 的子空间 $\text{span}\{\mathbf{b}\}$ 上的投影矩阵 \mathbf{P}_{π} , 其中 $\mathbf{b} = (1, 2, 2)^T$ 。

由上面的结论可得

$$\mathbf{P}_{\pi} = \frac{\mathbf{b} \mathbf{b}^T}{\mathbf{b}^T \mathbf{b}} = \frac{1}{9} \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 2 \end{pmatrix} = \frac{1}{9} \begin{pmatrix} 1 & 2 & 2 \\ 2 & 4 & 4 \\ 2 & 4 & 4 \end{pmatrix}$$

(a) $x \in \mathbb{R}^2$ 映射到以 b 为基的子空间 \mathbb{U} 上。(b) 将 2 维空间中的满足 $\|x\| = 1$ 的单位向量 x 映射到 b 张成的子空间 \mathbb{U} 上。

给定向量 $x = (1, 1, 1)^T$ 其投影为

$$\pi_{\mathbb{U}}(x) = \mathbf{P}_{\pi}x = \frac{1}{9} \begin{pmatrix} 1 & 2 & 2 \\ 2 & 4 & 4 \\ 2 & 4 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{9} \begin{pmatrix} 5 \\ 10 \\ 10 \end{pmatrix} \in \text{Col} \left(\begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} \right)$$

接下来，我们考虑更一般的情况。

投影到一般子空间

我们将 \mathbb{R}^m 中的向量 $x \in \mathbb{R}^m$ 投影到更高维的子空间 $\mathbb{U} \subseteq \mathbb{R}^m$ 中，其中 $\dim(\mathbb{U}) = n \geq 1$ 。

设 $B = (b_1, \dots, b_n)$ 是子空间 \mathbb{U} 的一个有序基底。 \mathbb{U} 上的任何投影 $\pi_{\mathbb{U}}(x)$ 必须是 \mathbb{U} 中的一个元素。故有

$$\pi_{\mathbb{U}}(x) = \sum_{i=1}^n \lambda_i b_i$$

和一维情况一样，我们将逐步确定 $\lambda_1, \dots, \lambda_n$, $\pi_{\mathbb{U}}(x)$ 和投影矩阵 \mathbf{P}_{π} 。

1. 确定 $\lambda_1, \dots, \lambda_n$ 设

$$\pi_{\mathbb{U}}(x) = \sum_{i=1}^n \lambda_i b_i = B\lambda \in \text{Col}(B)$$

最接近 $x \in \mathbb{R}^m$ ，其中 $B = [b_1, \dots, b_n] \in \mathbb{R}^{m \times n}$, $\lambda = [\lambda_1, \dots, \lambda_n]^T \in \mathbb{R}^n$ 。

因为 $\pi_{\mathbb{U}}(\mathbf{x})$ 是 \mathbf{x} 的投影，所以 $\mathbf{x} - \pi_{\mathbb{U}}(\mathbf{x}) \in \text{Col}(\mathbf{B})^\perp = \text{Null}(\mathbf{B}^T)$

$$\mathbf{b}_1^T(\mathbf{x} - \pi_{\mathbb{U}}(\mathbf{x})) = \langle \mathbf{b}_1, \mathbf{x} - \pi_{\mathbb{U}}(\mathbf{x}) \rangle = 0$$

$$\mathbf{b}_2^T(\mathbf{x} - \pi_{\mathbb{U}}(\mathbf{x})) = \langle \mathbf{b}_2, \mathbf{x} - \pi_{\mathbb{U}}(\mathbf{x}) \rangle = 0$$

⋮

$$\mathbf{b}_n^T(\mathbf{x} - \pi_{\mathbb{U}}(\mathbf{x})) = \langle \mathbf{b}_n, \mathbf{x} - \pi_{\mathbb{U}}(\mathbf{x}) \rangle = 0$$

使用矩阵可以将上式改写成

$$\mathbf{b}_1^T(\mathbf{x} - \mathbf{B}\lambda) = 0$$

⋮

$$\mathbf{b}_n^T(\mathbf{x} - \mathbf{B}\lambda) = 0$$

故有

$$\begin{pmatrix} \mathbf{b}_1^T \\ \vdots \\ \mathbf{b}_n^T \end{pmatrix} (\mathbf{x} - \mathbf{B}\lambda) = \mathbf{0} \iff \mathbf{B}^T(\mathbf{x} - \mathbf{B}\lambda) = \mathbf{0} \iff \mathbf{B}^T\mathbf{B}\lambda = \mathbf{B}^T\mathbf{x}$$

最终的方程我们称之为正规方程。因为 $\mathbf{b}_1, \dots, \mathbf{b}_n$ 是 \mathbb{U} 的基。因此 $\mathbf{B}^T\mathbf{B}$ 是可逆的 ($\mathbf{B}^T\mathbf{B}\mathbf{y} = \mathbf{0} \implies \mathbf{y}^T\mathbf{B}^T\mathbf{B}\mathbf{y} = 0 \implies \mathbf{B}\mathbf{y} = \mathbf{0}$)。也就是说

$$\lambda = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{x}$$

2. 确定 $\pi_{\mathbb{U}}(\mathbf{x})$

$$\lambda = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{x}$$

λ 也就是 $\pi_{\mathbb{U}}(\mathbf{x})$ 在有序基底 \mathbf{B} 下的坐标。

$$\pi_{\mathbb{U}}(\mathbf{x}) = \mathbf{B}\lambda = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{x}$$

3. 确定 \mathbf{P}_{π} 由上面的讨论容易看出

$$\mathbf{P}_{\pi} = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T$$

例 3.2.4. 已知 \mathbb{R}^3 中的子空间 $\mathbb{U} = \text{span} \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \right\}$ 和向量 $\mathbf{x} = \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix}$ ，确定 \mathbf{x} 投影到 \mathbb{U} 上的坐标 λ 和投影点 $\pi_{\mathbb{U}}(\mathbf{x})$ 和投影矩阵 \mathbf{P}_{π}

解. 首先确定 $\mathbf{B} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}$

其次计算

$$\mathbf{B}^T \mathbf{B} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 3 & 3 \\ 3 & 5 \end{pmatrix}, \quad \mathbf{B}^T \mathbf{x} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \end{pmatrix}$$

然后只需要解方程 $\mathbf{B}^T \mathbf{B} \lambda = \mathbf{B}^T \mathbf{x}$ 得到 λ ,

$$\begin{pmatrix} 3 & 3 \\ 3 & 5 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \end{pmatrix} \iff \lambda = \begin{pmatrix} 5 \\ -3 \end{pmatrix}$$

故投影点 $\pi_{\mathbb{U}}(\mathbf{x}) = \mathbf{B}\lambda = \begin{pmatrix} 5 \\ 2 \\ -1 \end{pmatrix}$ 。最后

$$\mathbf{P}_{\pi} = \mathbf{B}(\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T = \frac{1}{6} \begin{pmatrix} 5 & 2 & -1 \\ 2 & 2 & 2 \\ -1 & 2 & 5 \end{pmatrix}$$

我们还可以验证 $\mathbf{P}_{\pi}^2 = \mathbf{P}_{\pi}$

投影到仿射子空间

到目前为止，我们讨论了如何将向量投影到低维子空间 \mathbb{U} 上。下面，我们将讨论如何将向量投影到仿射子空间上。

高情勿外传

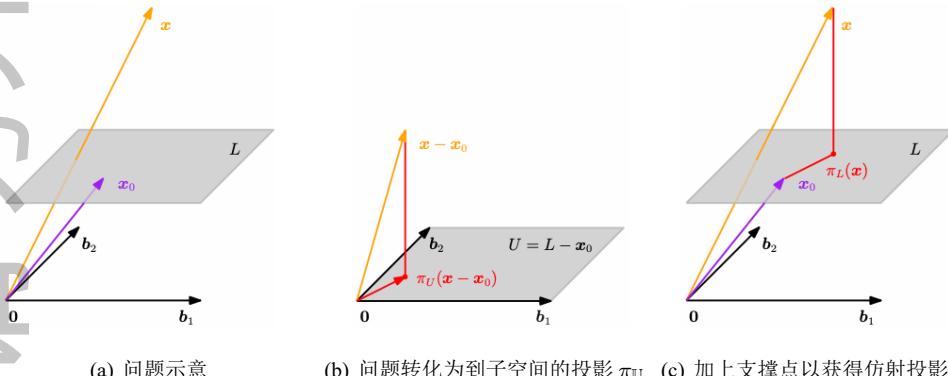


图 3.14: 投影到仿射空间。

考虑图 (a)。给定一个仿射空间 $\mathbb{L} = \mathbf{x}_0 + \mathbb{U}$, 其中 $\mathbf{b}_1, \mathbf{b}_2$ 是 \mathbb{U} 的基向量。为了确定 \mathbf{x} 在 \mathbb{L} 上的正交投影 $\pi_{\mathbb{L}}(\mathbf{x})$ 。我们将问题转化为我们知道如何解决的问题：投影到向量子空间上。为此，我们从 \mathbf{x} 和 \mathbb{L} 中减去支撑点 \mathbf{x}_0 ，所以 $\mathbb{L} - \mathbf{x}_0 = \mathbb{U}$ 恰好是向量子空间 \mathbb{U} 。

现在，我们可以用前面讨论过的在子空间上的正交投影，来获得投影 $\pi_{\mathbb{U}}(\mathbf{x} - \mathbf{x}_0)$ ，如图 (b) 所示。

最后我们通过添加 \mathbf{x}_0 将该投影转换回 \mathbb{L} ，这样我们就可以得出仿射空间 \mathbb{L} 上的正交投影为

$$\pi_{\mathbb{L}}(\mathbf{x}) = \mathbf{x}_0 + \pi_{\mathbb{U}}(\mathbf{x} - \mathbf{x}_0)$$

3.3 正交基与 Gram-Schmidt 正交化

3.3.1 标准正交基

线性代数中已经学过，线性空间中的向量可以由该空间的一组基表示。

定义 3.3.1. [标准正交基] 设 n 维向量 $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_r$ 是向量空间 $\mathbb{V} (\mathbb{V} \subset \mathbb{R}^n)$ 的一个基，如果 $\mathbf{e}_1, \dots, \mathbf{e}_r$ 两两正交，且都是单位向量，即对于 $\forall i, j = 1, \dots, r$ ，有

$$\langle \mathbf{e}_i, \mathbf{e}_j \rangle = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

则称 $\mathbf{e}_1, \dots, \mathbf{e}_r$ 是 \mathbb{V} 的一个规范(标准)正交基，有时也简称做正交基。

若 $\mathbf{e}_1, \dots, \mathbf{e}_r$ 是 \mathbb{V} 的一个规范正交基，那么 \mathbb{V} 中任意向量 \mathbf{a} 可以由 $\mathbf{e}_1, \dots, \mathbf{e}_r$ 线性表示，设表示为

$$\mathbf{a} = \lambda_1 \mathbf{e}_1 + \lambda_2 \mathbf{e}_2 + \dots + \lambda_r \mathbf{e}_r,$$

为求其中的系数 $\lambda_i (i = 1, \dots, r)$ ，可以计算 \mathbf{e}_i 与 \mathbf{a} 的内积，有

$$\langle \mathbf{e}_i, \mathbf{a} \rangle = \langle \mathbf{e}_i, \lambda_1 \mathbf{e}_1 + \lambda_2 \mathbf{e}_2 + \dots + \lambda_r \mathbf{e}_r \rangle = \lambda_1 \langle \mathbf{e}_i, \mathbf{e}_1 \rangle + \lambda_2 \langle \mathbf{e}_i, \mathbf{e}_2 \rangle + \dots + \lambda_r \langle \mathbf{e}_i, \mathbf{e}_r \rangle = \lambda_i$$

即

$$\lambda_i = \langle \mathbf{a}, \mathbf{e}_i \rangle$$

利用这个公式能方便地求得向量的坐标。因此，我们给向量空间取基时常常取标准正交基。接下来我们应用投影的思想，确定 $\text{Col}(\mathbf{A})$ 中的一组标准正交基。

3.3.2 Gram-Schmidt 正交化

设 $\mathbf{a}_1, \dots, \mathbf{a}_r$ 是向量空间 \mathbb{V} 的一个基：我们的目的是找到一组正交基 $\mathbf{e}_1, \dots, \mathbf{e}_r$ 使得

$$\text{span}\{\mathbf{e}_1, \dots, \mathbf{e}_r\} = \text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_r\}$$

我们可以这样做，我们先取 \mathbf{a}_1 作为一个基，记为 \mathbf{b}_1 。那么 \mathbf{a}_2 可以正交分解

$$\mathbf{a}_2 = \mathbf{a}_2^{(1)} + \mathbf{a}_2^{(2)},$$

其中 $\mathbf{a}_2^{(1)} \in \text{Col}((\mathbf{a}_1))$, $\mathbf{a}_2^{(2)} \in \text{Null}((\mathbf{a}_1)^T)$ 。利用投影公式：

$$\mathbf{a}_2^{(1)} = \frac{\langle \mathbf{b}_1, \mathbf{a}_2 \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} \mathbf{b}_1$$

$$\mathbf{a}_2^{(2)} = \mathbf{a}_2 - \frac{\langle \mathbf{b}_1, \mathbf{a}_2 \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} \mathbf{b}_1$$

我们记 $\mathbf{a}_2^{(2)}$ 为 \mathbf{b}_2 。并把 \mathbf{b}_2 添加到正交基中, $\text{span}\{\mathbf{a}_1, \mathbf{a}_2\} = \text{span}\{\mathbf{b}_1, \mathbf{b}_2\}$ 。注意这里 $\mathbf{b}_1, \mathbf{b}_2$ 还不是标准正交基。

假设我们已经有了一组有序正交基底 $\mathbf{B}_k = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k)$, 那么 \mathbf{a}_{k+1} 可以正交分解

$$\mathbf{a}_{k+1} = \mathbf{a}_{k+1}^{(1)} + \mathbf{a}_{k+1}^{(2)},$$

其中 $\mathbf{a}_{k+1}^{(1)} \in \text{Col}(\mathbf{B}_k)$, $\mathbf{a}_{k+1}^{(2)} \in \text{Null}(\mathbf{B}_k^T)$ 。利用投影公式:

$$\begin{aligned} \mathbf{a}_{k+1}^{(1)} &= \pi_{\text{Col}(\mathbf{B}_k)}(\mathbf{a}_{k+1}) = \mathbf{B}_k(\mathbf{B}_k^T \mathbf{B}_k)^{-1} \mathbf{B}_k^T \mathbf{a}_{k+1} \\ &= (\mathbf{b}_1, \dots, \mathbf{b}_k) \begin{pmatrix} \langle \mathbf{b}_1, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_1, \mathbf{b}_k \rangle \\ \cdots & \ddots & \cdots \\ \langle \mathbf{b}_k, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_k, \mathbf{b}_k \rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle \mathbf{b}_1, \mathbf{a}_{k+1} \rangle \\ \vdots \\ \langle \mathbf{b}_k, \mathbf{a}_{k+1} \rangle \end{pmatrix} \end{aligned}$$

而 $\{\mathbf{b}_1, \dots, \mathbf{b}_k\}$ 是相互正交的。也就是说若 $i \neq j$, $\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0$ 。

所以

$$\begin{aligned} \mathbf{a}_{k+1}^{(1)} &= (\mathbf{b}_1, \dots, \mathbf{b}_k) \begin{pmatrix} \langle \mathbf{b}_1, \mathbf{b}_1 \rangle & & \\ & \ddots & \\ & & \langle \mathbf{b}_k, \mathbf{b}_k \rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle \mathbf{b}_1, \mathbf{a}_{k+1} \rangle \\ \vdots \\ \langle \mathbf{b}_k, \mathbf{a}_{k+1} \rangle \end{pmatrix} \\ &= \frac{\langle \mathbf{b}_1, \mathbf{a}_{k+1} \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} \mathbf{b}_1 + \frac{\langle \mathbf{b}_2, \mathbf{a}_{k+1} \rangle}{\langle \mathbf{b}_2, \mathbf{b}_2 \rangle} \mathbf{b}_2 + \cdots + \frac{\langle \mathbf{b}_k, \mathbf{a}_{k+1} \rangle}{\langle \mathbf{b}_k, \mathbf{b}_k \rangle} \mathbf{b}_k \end{aligned}$$

而 $\mathbf{a}_{k+1}^{(2)} = \mathbf{a}_{k+1} - \mathbf{a}_{k+1}^{(1)}$, 我们记 $\mathbf{a}_{k+1}^{(2)}$ 为 \mathbf{b}_{k+1} , 并把 \mathbf{b}_{k+1} 添加到正交基中, $\text{span}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{k+1}\} = \text{span}\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{k+1}\}$ 。

以此类推, 我们可以得到 $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r\}$ 使得

$$\text{span}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r\} = \text{span}\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r\}.$$

只需要再把这组基单位化即可。

Gram-Schmidt 正交化 总结之前的过程, 可以通过以下方法求得 \mathbb{V} 的一个规范正交基 $\mathbf{e}_1, \dots, \mathbf{e}_r$ 。

这种方法称为 Gram-Schmidt 正交化。

取

$$\mathbf{b}_1 = \mathbf{a}_1;$$

$$\mathbf{b}_2 = \mathbf{a}_2 - \frac{\langle \mathbf{b}_1, \mathbf{a}_2 \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} \mathbf{b}_1;$$

.....

$$\mathbf{b}_r = \mathbf{a}_r - \frac{\langle \mathbf{b}_1, \mathbf{a}_r \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} \mathbf{b}_1 - \frac{\langle \mathbf{b}_2, \mathbf{a}_r \rangle}{\langle \mathbf{b}_2, \mathbf{b}_2 \rangle} \mathbf{b}_2 - \cdots - \frac{\langle \mathbf{b}_{r-1}, \mathbf{a}_r \rangle}{\langle \mathbf{b}_{r-1}, \mathbf{b}_{r-1} \rangle} \mathbf{b}_{r-1}$$

然后把它们单位化，取

$$\mathbf{e}_1 = \frac{1}{\|\mathbf{b}_1\|} \mathbf{b}_1, \mathbf{e}_2 = \frac{1}{\|\mathbf{b}_2\|} \mathbf{b}_2, \dots, \mathbf{e}_r = \frac{1}{\|\mathbf{b}_r\|} \mathbf{b}_r$$

就是 \mathbb{V} 的一个规范正交基。

例 3.3.1. 求向量组 $\mathbf{a}_1 = (3, 1, 1)^T, \mathbf{a}_2 = (2, 2, 0)^T$ 的生成子空间的标准正交基。

取

$$\mathbf{b}_1 = (3, 1, 1)^T$$

$$\mathbf{b}_2 = \mathbf{a}_2 - \frac{\mathbf{b}_1^T \mathbf{a}_2}{\mathbf{b}_1^T \mathbf{b}_1} \mathbf{b}_1 = (2, 2, 0)^T - \frac{8}{11}(3, 1, 1)^T = \frac{-2}{11}(1, -7, 4)^T$$

$$\mathbf{e}_1 = \frac{1}{\sqrt{11}}(3, 1, 1)^T$$

$$\mathbf{e}_2 = \frac{1}{\sqrt{66}}(1, -7, 4)^T$$

故标准正交基为 $\mathbf{e}_1, \mathbf{e}_2$: 即,

$$\frac{1}{\sqrt{11}}(3, 1, 1)^T, \frac{1}{\sqrt{66}}(1, -7, 4)^T$$

正交和投影是基础性概念, 与超定系统的最小二乘解, 并与机器学习中的降维、分类或回归都有紧密联系, 我们在第 5 章中进一步给出。

3.4 具有特殊结构和性质的矩阵

本节我们介绍一些特殊结构的正交矩阵, 包括旋转矩阵、反射矩阵和信号处理中常见的矩阵。特别由旋转和反射引出的 Householder 变换矩阵和 Givens 变换矩阵将用于下一章构造矩阵的正交分解。

3.4.1 特殊的正交变换矩阵——旋转

旋转是如信号处理、机器学习、机器人学中的一个基本的研究对象, 学习旋转或从给定的一组样本中找到潜藏的旋转问题有许多实际应用 (包括计算机视觉、人脸识别、姿态估计、晶体物理学)。除了它们在实践领域重要性之外, 在理论上, 旋转具有一般映射不具有的性质。例如, 旋转是一种线性保角变换。在群论中, n 维空间的旋转矩阵构成了特殊正交群 $\mathcal{SO}(n)$ 。

旋转过程中, 线段的长度、直线间的夹角大小是保持不变的。旋转也是一种线性映射。在第 2 章中, 我们介绍过如何对一张图片进行旋转。本节, 我们从平面空间中的旋转出发, 推广到一般空间中的向量旋转。

平面上的旋转

在平面内，一个图形绕着一个定点旋转一定的角度得到另一个图形的变化叫做旋转。这个定点叫做旋转中心，旋转的角度叫做旋转角，如果一个图形上的点 A 经过旋转变为点 A'，那么这两个点叫做旋转的对应点。

旋转是一个线性映射，更具体地，可以看成欧氏空间的一个自同构，它把空间中元素映射为另外一个元素。

在一个平面中，如果我们说一个点绕原点旋转 $\theta > 0$ 保持以下约定：

- 原点是固定的点
- 一般，旋转方向规定为逆时针

例 3.4.1. 我们考虑定义在 \mathbb{R}^2 上平面直角坐标系的自然基底 $\left\{ \mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$

我们把旋转 θ 这个线性变换记为 Φ_θ ，容易得到：

$$\Phi_\theta(\mathbf{e}_1) = \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}, \quad \Phi_\theta(\mathbf{e}_2) = \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}$$

设 \mathbb{R}^2 中任一点 $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 \mathbf{e}_1 + x_2 \mathbf{e}_2$

那么 $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ 旋转 θ 后的坐标：

$$\Phi_\theta(\mathbf{x}) = x_1 \Phi_\theta(\mathbf{e}_1) + x_2 \Phi_\theta(\mathbf{e}_2) = [\Phi_\theta(\mathbf{e}_1), \Phi_\theta(\mathbf{e}_2)] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

所以平面上旋转 θ 的变换矩阵 \mathbf{R}_θ 为：

$$\mathbf{R}_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

三维空间中的旋转

例 3.4.2. 对于 \mathbb{R}^3 中的向量 \mathbf{x} ，设 \mathbb{R}^3 的三个基底分别为 $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ 。若 \mathbf{x} 绕 \mathbf{e}_3 旋转 θ ，记为 Φ_θ^3 ，类似 2 维空间的做法

$$\Phi_\theta^3(\mathbf{e}_1) = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix}, \quad \Phi_\theta^3(\mathbf{e}_2) = \begin{bmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{bmatrix}, \quad \Phi_\theta^3(\mathbf{e}_3) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

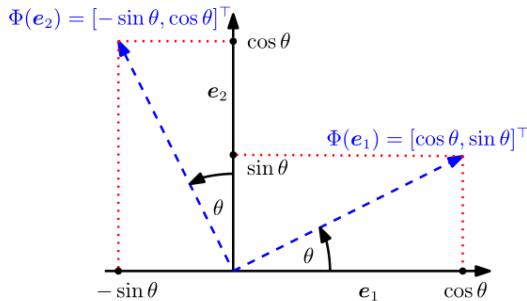


图 3.15: 平面中的旋转

因此, 绕 e_3 旋转 θ 的变换矩阵 R_θ^3 为:

$$R_\theta^3 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

例 3.4.3. 类似的, 绕 e_1 旋转 θ 的变换矩阵 R_θ^1 为:

$$R_\theta^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

绕 e_2 旋转 θ 的变换矩阵 R_θ^2 为:

$$R_\theta^2 = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

考虑 n 维空间中的旋转, 固定其中的 $n-2$ 维, 在 n 维空间中的 2 维子平面上旋转

高维空间中的旋转

在 n 维空间中, 我们可以固定其中的 $n-2$ 维, 在 n 维空间中的 2 维子平面上旋转。

定义 3.4.1. 令 \mathbb{V} 是 n 维欧氏向量空间, $\Phi : \mathbb{V} \rightarrow \mathbb{V}$ 是一线性变换, 其变换矩阵

$$R_{i,j}(\theta) := \begin{bmatrix} I_{i-1} & & & \\ & \cos \theta & -\sin \theta & \\ & \sin \theta & \cos \theta & \\ & & & I_{n-j} \end{bmatrix}$$

其中 $1 \leq i < j \leq n$, $\theta \in \mathbb{R}$ 。那么 $R_{i,j}$ 叫做 **Givens 旋转矩阵**。

2 维旋转是 $n=2$ 时 Givens 旋转的一个特殊情形。

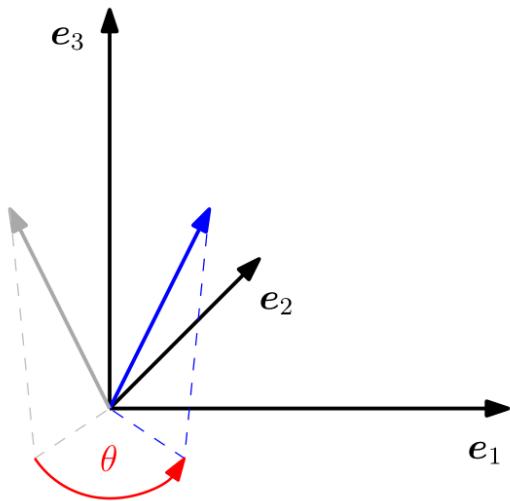


图 3.16: 三维空间中的旋转

旋转矩阵的性质

所有的旋转矩阵都是正交矩阵。但并不是所有的正交矩阵都是旋转矩阵。比如

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

是正交矩阵，但它不是一个旋转矩阵，事实上，它是一个使向量关于 \$x\$ 轴对称的反射(镜像)矩阵。

性质 3.4.1. 设 $\mathbf{R} \in \mathbb{R}^{n \times n}$, \mathbf{R} 是旋转矩阵当且仅当它是正交矩阵并且 $\det(\mathbf{R}) = 1$ 。

性质 3.4.2. 保距性：设 $\mathbf{R}_\theta \in \mathbb{R}^{n \times n}$ 是旋转矩阵, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, 有 $\|\mathbf{x} - \mathbf{y}\|_2 = \|\mathbf{R}_\theta(\mathbf{x}) - \mathbf{R}_\theta(\mathbf{y})\|_2$ 。

即空间中的两个点在旋转前后距离保持不变。

性质 3.4.3. 保角性：设 $\mathbf{R}_\theta \in \mathbb{R}^{n \times n}$ 是旋转矩阵, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$, 有 $\frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\langle \mathbf{R}_\theta(\mathbf{x}), \mathbf{R}_\theta(\mathbf{y}) \rangle}{\|\mathbf{R}_\theta(\mathbf{x})\| \|\mathbf{R}_\theta(\mathbf{y})\|}$ 。

即空间中的两个向量在旋转前后角度保持不变。 $\mathbf{R}_\theta(\mathbf{x}), \mathbf{R}_\theta(\mathbf{y})$ 的夹角与 \mathbf{x}, \mathbf{y} 的夹角相同。

性质 3.4.4. 多个旋转矩阵的乘积仍然是旋转矩阵。

性质 3.4.5. 仅在 2 维情形有可交换性即 $\mathbf{R}_\theta \mathbf{R}_\phi = \mathbf{R}_\phi \mathbf{R}_\theta$ 。

$$\begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{pmatrix} = \begin{pmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}$$

在 3 维或更高维, 交换性不成立, 比如在 3 维情形下 e_3 绕 e_3 旋转 $\pi/2$ 仍是 e_3 , 再绕 e_2 旋转 $\pi/2$ 会变换到 e_1 。如果 e_3 先绕 e_2 旋转 $\pi/2$ 到 e_1 , 再绕 e_3 旋转 $\pi/2$ 会变换到 e_2 。

群

我们之前介绍说，旋转矩阵构成了一个群。

群就是一个定义了满足封闭性、结合律、有单位元和逆元的二元运算的集合。具体说

定义 3.4.2. 考虑一个集合 \mathbb{G} 和定义在 \mathbb{G} 上二元运算 $\circ : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}$ ，如果 $G := (\mathbb{G}, \circ)$ 满足以下条件就被称为群：

- 1 封闭性： $\forall x, y \in \mathbb{G} : x \circ y \in \mathbb{G}$ ；
- 2 结合性： $\forall x, y, z \in \mathbb{G} : (x \circ y) \circ z = x \circ (y \circ z)$ ；
- 3 有单位元： $\exists e \in \mathbb{G}, \forall x \in \mathbb{G} : x \circ e = x$ 并且 $e \circ x = x$ ；
- 4 有逆元： $\forall x \in \mathbb{G}, \exists y \in \mathbb{G} : x \circ y = e$ 并且 $y \circ x = e$ 。我们一般把 x 的逆元记做 x^{-1} 。

此外，如果 $\forall x, y \in \mathbb{G} : x \circ y = y \circ x$ ，那么 $G = (\mathbb{G}, \circ)$ 称作阿贝尔群，或称作可交换群。

例 3.4.4. $(\mathbb{R}, +)$ 构成群，也是阿贝尔群。

例 3.4.5. (\mathbb{R}, \cdot) 不构成群，因为 0 在乘法中没有逆元。 $(\mathbb{R} \setminus \{0\}, \cdot)$ 构成一个群，也是阿贝尔群。

例 3.4.6. $\mathbb{R}^{n \times n}$ 中的所有正交矩阵构成的集合在矩阵乘法下构成群，称为正交群，记为 $O(n)$ 。

例 3.4.7. $\mathbb{R}^{n \times n}$ 中的所有旋转矩阵构成的集合在矩阵乘法下构成群，称为特殊正交群，记为 $SO(n)$ 。当 $n = 2$ 时， $SO(2)$ 是阿贝尔群。

例 3.4.8. $SO(n)$ 是一个特殊的正交群。

我们可以简单的进行验证：

- 单位元就是单位矩阵，即逆时针旋转 0 度的变换矩阵。
- 一个旋转矩阵逆元就是这个矩阵的逆，比如平面中逆时针旋转 90° 的逆元就是顺时针旋转 90° ，或者说逆时针旋转 270° 。
- 矩阵的乘法自然是满足结合律的。
- 而封闭性则意味着多个旋转矩阵的乘积仍然是旋转矩阵。

旋转矩阵的应用

正交普洛克路斯忒斯 (Procrustes) 问题和瓦赫巴 (Wahba) 问题

在古希腊神话里，有个强盗，叫普洛克路斯忒斯 (Procrustes)。他开设了家黑店，经常邀请过往客人，并告诉他们有一张正合适的床。实际上，他设置了两张铁床，一长一短。如果客人比较矮，他就强迫客人睡长床，并拉扯客人的身体，使其和床一样长。如果客人比较高，他就会强迫客人睡短床，并截断客人的腿。无论哪种情况，客人都会死掉。最后英雄忒修斯 (Theseus) 击败了普洛克路斯忒斯，强令他躺在自己的短床上，并把这个强盗伸出床外的腿砍掉了。

例 3.4.9. 正交 Procrustes 问题：使一组数据通过正交变换近似匹配另外一组数据。假设 $\mathbf{x}_t, \mathbf{y}_t, 1 \leq t \leq T$ 是 \mathbb{R}^n 中的单位向量。考虑一组实例 $\mathbf{x}_t, 1 \leq t \leq T$ ，目标是预测 $\hat{\mathbf{y}}_t = \mathbf{Q}\mathbf{x}_t$ ，它是 \mathbf{x}_t 正交变换后的数据。 \mathbf{y}_t 是 \mathbf{x}_t 旋转后的真实值，那么第 t 个实例的预测损失为 $L_t(\mathbf{R}) = \|\mathbf{R}\mathbf{x}_t - \mathbf{y}_t\|^2$ 。目标是使总的损失即 $L(\mathbf{Q}) = \sum_{t=1}^T \frac{1}{2} \|\mathbf{Q}\mathbf{x}_t - \mathbf{y}_t\|^2$ 最小。

为了解决这个问题，我们进行推导

$$\begin{aligned} \arg \min_{\mathbf{Q} \in \mathcal{O}(n)} \sum_{t=1}^T \frac{1}{2} \|\mathbf{Q}\mathbf{x}_t - \mathbf{y}_t\|^2 &= \arg \min_{\mathbf{Q} \in \mathcal{O}(n)} T - \left(\sum_{t=1}^T \mathbf{y}_t^\top \mathbf{Q}\mathbf{x}_t \right) \\ &= \arg \max_{\mathbf{Q} \in \mathcal{O}(n)} \text{Tr} \left(\left(\sum_{t=1}^T \mathbf{x}_t \mathbf{y}_t^\top \right) \mathbf{Q} \right) \end{aligned}$$

我们记 $\mathbf{S} := \sum_{t=1}^T \mathbf{x}_t \mathbf{y}_t^\top$ ，那么这个问题变形为

$$\arg \max_{\mathbf{Q} \in \mathcal{O}(n)} \text{Tr}(\mathbf{S}\mathbf{Q})$$

如果我们要求这个正交矩阵必须是旋转矩阵，那么这个问题形式为

$$\arg \min_{\mathbf{R} \in \mathcal{SO}(n)} \sum_{t=1}^T \frac{1}{2} \|\mathbf{R}\mathbf{x}_t - \mathbf{y}_t\|^2 = \arg \max_{\mathbf{R} \in \mathcal{SO}(n)} \text{Tr}(\mathbf{S}\mathbf{R})$$

此时问题变为 Wahba 问题。

我们可以通过奇异值分解的办法解决这两个问题。我们将会在下一章介绍奇异值分解。

3.4.2 反射矩阵

平面上的反射变换

下面我们考虑另外一种特殊的正交矩阵：反射矩阵。

问题 考虑二维平面中的一个向量 $\mathbf{x} = (x_1, x_2)^\top$, $\mathbf{b} = (\cos \theta, \sin \theta)^\top$, 如何求得向量 \mathbf{x} 关于子空间 $\text{span}\{\mathbf{b}\}$ 对称的向量 \mathbf{x}' ?

由于 \mathbf{x} 和 \mathbf{x}' 关于子空间 $\text{span}\{\mathbf{b}\}$ 对称，所以 $\frac{\mathbf{x} + \mathbf{x}'}{2} = \mathbf{u}$ 。其中 \mathbf{u} 是 \mathbf{x} 在子空间 $\text{span}\{\mathbf{b}\}$ 上的投影。 $\mathbf{v} \in (\text{span}\{\mathbf{b}\})^\perp$, $\mathbf{x} = \mathbf{u} + \mathbf{v}$, 那么

$$\mathbf{x}' = \mathbf{u} - \mathbf{v} = \mathbf{x} - 2\mathbf{v} = 2\mathbf{u} - \mathbf{x}$$

根据投影公式 $\mathbf{u} = \mathbf{b}\mathbf{b}^\top \mathbf{x}$, 那么

$$\begin{aligned} \mathbf{u} &= \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta \end{pmatrix} \mathbf{x} \\ &= \begin{pmatrix} \cos^2 \theta & \cos \theta \sin \theta \\ \cos \theta \sin \theta & \sin^2 \theta \end{pmatrix} \mathbf{x} \end{aligned}$$

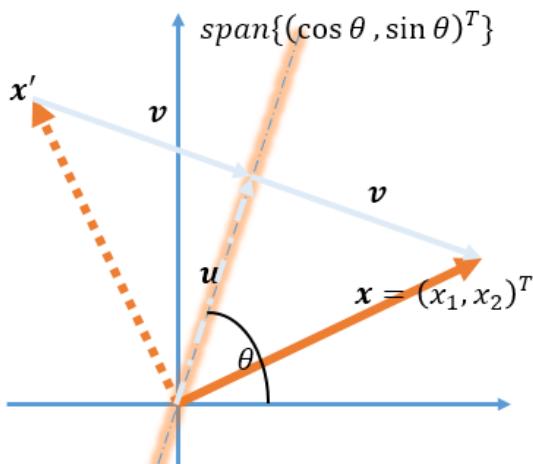


图 3.17: 平面上的镜象反射

则

$$\begin{aligned} \mathbf{x}' &= 2\mathbf{u} - \mathbf{x} = \begin{pmatrix} 2\cos^2\theta - 1 & 2\cos\theta\sin\theta \\ 2\cos\theta\sin\theta & 2\sin^2\theta - 1 \end{pmatrix} \mathbf{x} \\ &= \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix} \mathbf{x} \end{aligned}$$

令 $\phi = 2\theta$,

$$\mathbf{x}' = \begin{pmatrix} \cos \phi & \sin \phi \\ \sin \phi & -\cos \phi \end{pmatrix} \mathbf{x}$$

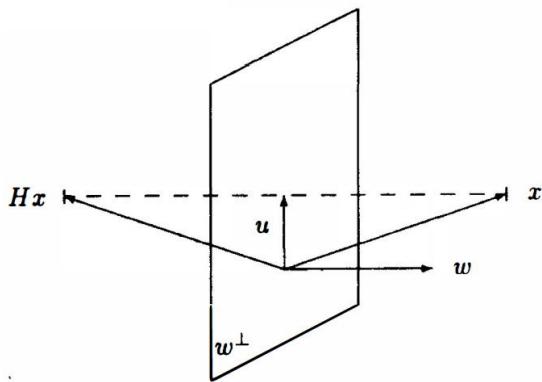
或者我们记 $(\text{span}\{\mathbf{b}\})^\perp$ 中的单位向量为 \mathbf{w} , 容易求得 $\mathbf{w} = (\sin \theta, -\cos \theta)^T$ 。 \mathbf{v} 是 \mathbf{x} 在 $\text{span}\{\mathbf{w}\}$ 上的投影, 即 $\mathbf{v} = \mathbf{w}\mathbf{w}^T\mathbf{x}$ 。利用

$$\begin{aligned} \mathbf{x}' &= \mathbf{x} - 2\mathbf{v} = (\mathbf{I} - 2\mathbf{w}\mathbf{w}^T)\mathbf{x} \\ \mathbf{x}' &= \begin{pmatrix} 1 - 2\sin^2\theta & 2\cos\theta\sin\theta \\ 2\cos\theta\sin\theta & 1 - 2\cos^2\theta \end{pmatrix} \mathbf{x} = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix} \mathbf{x} \end{aligned}$$

我们可以得到同样的结论。

高维空间上的反射变换

在 3 维空间中, 我们有时需要得到一个向量关于一个 2 维平面的镜像, 在 n 维空间中, 我们有时需要得到一个向量关于 $n-1$ 维超平面的镜像。这时我们根据平面的法向量 \mathbf{w} 可以很容易的求出关于 \mathbf{w} 垂直的超平面的镜象反射。

图 3.18: Householder 变换是关于 w 的垂直超平面的镜面反射。

定义 3.4.3. 设 $w \in \mathbb{R}^n$ 满足 $\|w\|_2 = 1$, 定义 $H \in \mathbb{R}^{n \times n}$ 为

$$H = I - 2ww^T \quad (3.9)$$

则称 H 为 **Householder** 变换矩阵。

Householder 变换也叫做初等反射矩阵或者镜像变换, 它是著名的数值分析专家 Householder 在 1958 年为讨论矩阵特征值问题而提出来的。我们可以利用 Householder 变换来进行高维空间上的反射变换。

下面的定理给出了 Householder 变换的一些简单而又十分重要的性质:

定理 3.4.1. 设 H 是由(3.9)定义的一个 Householder 变换, 那么 H 满足

(1) 对称性: $H^T = H$;

(2) 正交性: $H^T H = I$;

(3) 对合性: $H^2 = I$;

(4) 反射性: 对任意的 $x \in \mathbb{R}^n$, 如右图所示, Hx 是 x 关于 w 的垂直超平面的镜像反射。

证明. (1) 显然。 (2) 和 (3) 可由 (1) 导出。事实上, 我们有

$$\begin{aligned} H^T H &= H^2 = (I - 2ww^T)(I - 2ww^T) \\ &= I - 4ww^T + 4ww^Tww^T = I \end{aligned}$$

(4) 设 $x \in \mathbb{R}^n$, 则 x 可表示为 $x = u + \alpha w$

其中 $u \in \text{span}\{w\}^\perp$, $\alpha \in \mathbb{R}$ 。利用 $u^T w = 0$ 和 $w^T w = 1$, 可得

$$\begin{aligned} Hx &= (I - 2ww^T)(u + \alpha w) \\ &= u + \alpha w - 2ww^T u - 2\alpha w w^T w \\ &= u - \alpha w \end{aligned}$$

□

这就说明了 Hx 为 x 关于 $\text{span}\{w\}^\perp$ 的镜像反射。

Householder 矩阵的特征值和行列式

这里我们用一个很简单办法说明 Householder 矩阵的行列式是 -1 。

我们知道, $R^{n \times n}$ 中的 Householder 矩阵 $H = I - 2ww^T$ 将 x 变换到关于 w 的垂直超平面 $\text{span}\{w\}^\perp$ 的镜像上, 这里 w 仍是单位向量。而在 $\text{span}\{w\}^\perp$ 上的每个向量, 它们关于这个超平面的镜像仍是本身。也就是 $Hx = x$, 若 $x \in \text{span}\{w\}^\perp$ 。而 $\text{span}\{w\}^\perp$ 是 $n-1$ 维的, 也就是说 H 至少有 $n-1$ 个特征值是 1 。而在 $\text{span}\{w\}$ 上的每个向量 $x = \alpha w$, 它们关于这个超平面的镜像是 $-x$ 。因为

$$Hx = (I - 2ww^T)x = \alpha w - 2\alpha w = -\alpha w = -x, \text{若 } x \in \text{span}\{w\}.$$

那么 H 至少有 1 个特征值是 -1 。

H 一共有 n 个特征值, 所以 H 全部特征值有 $n-1$ 个特征值是 1 , 1 个特征值是 -1 。

方阵的行列式是它所有特征值的乘积, 那么 $\det(H) = -1$ 。

最后, 我们给出旋转变换和反射变换复合的性质。

- 旋转矩阵乘以旋转矩阵仍然是旋转矩阵。
- 反射矩阵乘以反射矩阵会得到旋转矩阵。
- 旋转矩阵乘以反射矩阵会得到反射矩阵。

3.4.3 信号处理中常见的正交矩阵

Haar 矩阵

哈尔小波变换 (英语: Haar wavelet) 是由数学家阿尔弗雷德·哈尔于 1909 年所提出的函数变换, 也是小波变换中最简单的一种变换, 也是最早提出的小波变换。

Haar 矩阵是哈尔小波变换的离散情形。Haar 矩阵中的每个元素都是 0 、 $+1$ 或者 -1 , 并且任意两行都是正交的。

$n = 2$ 时,

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Haar 矩阵的构造

当 $n = 4$ 时, 我们可以如下构造 Haar 矩阵:

(1) 取第一行全部为 1 , $[1, 1, 1, 1]$ 。我们可以把这个行向量看作用 $[1, 1]$ 替换掉了 H_2 的第一行 $[1, 1]$ 中的每个 1 。

(2) 我们用 $[1, 1]$ 替换掉 H_2 第二行中的每个 1 , 得到 H_4 的第二行 $[1, 1, -1, -1]$ 。

(3) 前两行中, 每一行的前两个元素, 每一行的后两个元素都是一样的, 所以我们取第三行 $[1, -1, 0, 0]$, 第四行 $[0, 0, 1, -1]$ 。

这样任意两行都是正交的。

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

当 $n = 8$ 时, 如下构造 Haar 矩阵:

(1) 我们将 \mathbf{H}_4 中的每个 1 都替换为 $[1, 1]$ 。得到 \mathbf{H}_8 的前四行:

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

(2) 可以看到前四行第 $2k-1, 2k, k = 1, 2, 3, 4$ 个元素是相同的, 所以余下的四行我们分别令其第 $2j-1, 2j$ 个元素分别是 $1, -1$, 其余元素为 0。

这样任意两行也都是正交的。

$$\mathbf{H}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

Haar 矩阵变换的特点

哈尔变换有以下几点特性:

- 不需要乘法 (只有相加或加减)
- 输入与输出个数相同
- 可以分析一个信号的局部特征
- 大部分运算为 0, 不用计算

Haar 矩阵变换常用于图像信号的压缩。

Hadamard 矩阵

Hadamard 矩阵是以法国数学家雅克·阿达马命名的方阵。其元素要么是 +1，要么是 -1，是信号处理中的一种重要的矩阵。

定义 3.4.4. $\mathbf{H}_n \in \mathbb{R}^{n \times n}$ 称为 Hadamard 矩阵，若它的所有元素取 +1 或者 -1，并且满足

$$\mathbf{H}_n \mathbf{H}_n^T = \mathbf{H}_n^T \mathbf{H}_n = n \mathbf{I}_n$$

其中 \mathbf{I}_n 是 n 阶单位矩阵。

- 用 -1 乘 Hadamard 矩阵的任意行或者任意列得到的结果仍然是 Hadamard 矩阵。
- 称第一列和第一行所有元素都是 +1 的 Hadamard 矩阵为规范化的 Hadamard 矩阵。
- $\frac{1}{\sqrt{n}} \mathbf{H}_n$ 是标准正交矩阵。

Hadamard 矩阵的构造

定理 3.4.2. 令 $n = 2^k, k = 1, 2, \dots$ ，则规范化的 Hadamard 矩阵具有构造公式：

$$\mathbf{H}_{2n} = \begin{pmatrix} \mathbf{H}_n & \mathbf{H}_n \\ \mathbf{H}_n & -\mathbf{H}_n \end{pmatrix}, \quad \mathbf{H}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

证明. 用数学归纳法证明：可以验证 $\mathbf{H}_2^T \mathbf{H}_2 = \mathbf{H}_2 \mathbf{H}_2^T = 2\mathbf{I}_2$ 。

假设 $n = 2^k$ 时 \mathbf{H}_{2^k} 是规范化的正交 Hadamard 矩阵，即有 $\mathbf{H}_{2^k}^T \mathbf{H}_{2^k} = \mathbf{H}_{2^k} \mathbf{H}_{2^k}^T = 2^k \mathbf{I}_{2^k}$ 。于是，对 $n = 2^{k+1}$ ，那么

$$\mathbf{H}_{2^{k+1}} = \begin{pmatrix} \mathbf{H}_{2^k} & \mathbf{H}_{2^k} \\ \mathbf{H}_{2^k} & -\mathbf{H}_{2^k} \end{pmatrix}$$

我们只需要验证 $\mathbf{H}_{2^{k+1}}$ 是 Hadamard 矩阵即可。

$$\begin{aligned} \mathbf{H}_{2^{k+1}}^T \mathbf{H}_{2^{k+1}} &= \begin{pmatrix} \mathbf{H}_{2^k}^T & \mathbf{H}_{2^k}^T \\ \mathbf{H}_{2^k}^T & -\mathbf{H}_{2^k}^T \end{pmatrix} \begin{pmatrix} \mathbf{H}_{2^k} & \mathbf{H}_{2^k} \\ \mathbf{H}_{2^k} & -\mathbf{H}_{2^k} \end{pmatrix} \\ &= \begin{pmatrix} 2 \cdot 2^k \mathbf{I}_{2^k} & \mathbf{O}_{2^k} \\ \mathbf{O}_{2^k} & 2 \cdot 2^k \mathbf{I}_{2^k} \end{pmatrix} \\ &= 2^{k+1} \mathbf{I}_{2^{k+1}} \end{aligned}$$

类似地，容易证明 $\mathbf{H}_{2^{k+1}} \mathbf{H}_{2^{k+1}}^T = 2^{k+1} \mathbf{I}_{2^{k+1}}$ 。又由于 \mathbf{H}_{2^k} 是规范化的，所以 $\mathbf{H}_{2^{k+1}}$ 也是规范化的。因此，定理对 $n = 2^{k+1}$ 也成立。□

例 3.4.10. 当 $n = 2^2 = 4$ 时，Hadamard 矩阵为

$$\mathbf{H}_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

草稿勿外传

例 3.4.11.

$$\mathbf{H}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

(1) 以 \mathbf{H}_2 为有序基底, 求向量 $(3, 4)^T$ 在这组基下的坐标。

(2) 求向量 $(3, 4)^T$ 经过 \mathbf{H}_2 线性变换得到的向量。

解. (1) 因为 $3 + 4 = 7$, $3 - 4 = -1$, 所以向量 $(3, 4)^T$ 在这组基下的坐标为

$$\frac{1}{2}(7, -1)^T$$

(2) 因为 $3 + 4 = 7$, $3 - 4 = -1$, 所以向量 $(3, 4)^T$ 经过 \mathbf{H}_2 线性变换得到的向量为

$$(7, -1)^T$$

Hadamard 在信号处理中的优势

当 \mathbf{H} 为 Hadamard 矩阵时, 若 \mathbf{H} 作为线性空间的一组基, 由于 Hadamard 矩阵是正交矩阵并且元素只取 $+1$ 或 -1 , 我们计算一个向量在这组基下的坐标只需要加减法, 并将最后的结果统一除以 \mathbf{H} 的阶数。

线性变换 $\mathbf{y} = \mathbf{Hx}$ 称为 Hadamard 变换。同样由于 Hadamard 矩阵的元素只取 $+1$ 或 -1 , 因此, 计算变换后的向量只需要加法和减法而不需要乘法。Hadamard 变换常用于移动通信中的编码。

Haar 矩阵和 Hadamard 矩阵的紧凑表示**矩阵的克罗内克 (Kronecker) 积**

定义 3.4.5. 设 $\mathbf{A} = (a_{ik})_{m \times n}$, $\mathbf{B} = (b_{rl})_{r \times s}$, 是域 \mathbb{K} 中的两个矩阵, 则矩阵 $\mathbf{C} = (c_{\lambda\mu})_{mr \times ns}$ 称为 \mathbf{A} 与 \mathbf{B} 的克罗内克积, 记作 $\mathbf{C} = \mathbf{A} \otimes \mathbf{B}$, 即

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{pmatrix}$$

性质 3.4.6. 克罗内克积的性质 假设下述和、积有意义, 则克罗内克积具有下述性质:

- $\mathbf{A} \otimes (\mathbf{B}_1 + \mathbf{B}_2) = \mathbf{A} \otimes \mathbf{B}_1 + \mathbf{A} \otimes \mathbf{B}_2$, $\mathbf{A} \in \mathbb{R}_{m \times n}$, $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}_{p \times q}$
- $(\mathbf{A}_1 + \mathbf{A}_2) \otimes \mathbf{B} = \mathbf{A}_1 \otimes \mathbf{B} + \mathbf{A}_2 \otimes \mathbf{B}$, $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}_{m \times n}$, $\mathbf{B} \in \mathbb{R}_{p \times q}$
- $\mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) = (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C}$, $\mathbf{A} \in \mathbb{R}_{m \times n}$, $\mathbf{B} \in \mathbb{R}_{p \times q}$, $\mathbf{C} \in \mathbb{R}_{k \times l}$
- $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$, $\mathbf{A} \in \mathbb{R}_{m \times n}$, $\mathbf{B} \in \mathbb{R}_{p \times q}$
- $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$, $\mathbf{A} \in \mathbb{R}_{m \times n}$, $\mathbf{B} \in \mathbb{R}_{p \times q}$, $\mathbf{C} \in \mathbb{R}_{n \times r}$, $\mathbf{D} \in \mathbb{R}_{q \times s}$

克罗内克积的幂和连乘积

由于克罗内克积满足结合律，我们可以定义克罗内克积的幂和连乘积。

记

$$\mathbf{X}^{\otimes n} = \underbrace{\mathbf{X} \otimes \mathbf{X} \otimes \cdots \otimes \mathbf{X}}_{n \text{ 次}}$$

记

$$\bigotimes_{i=1}^n \mathbf{X}_i = \mathbf{X}_1 \otimes \mathbf{X}_2 \otimes \cdots \otimes \mathbf{X}_n$$

Haar 矩阵和 Hadamard 矩阵的克罗内克积表示

记

$$\mathbf{H}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

利用克罗内克积，我们可以利用 n 阶 Haar 矩阵构造 $2n$ 阶的 Haar 矩阵：

$$\mathbf{H}_{2n} = \begin{pmatrix} \mathbf{H}_n \otimes (1, 1) \\ \mathbf{I}_n \otimes (1, -1) \end{pmatrix}$$

其中 \mathbf{I}_n 是 n 阶单位矩阵。

我们也可以利用克罗内克积如下构造 2^k 阶的 Hadamard 矩阵：

$$\mathbf{H}_{2^k} = \mathbf{H}_2^{\otimes k}$$

傅里叶矩阵

因为傅里叶矩阵通常是定义在复数域上的复矩阵，先介绍与复矩阵有关的概念。

对于一个矩阵，我们可以定义它的共轭矩阵。

定义 3.4.6. 设 $\mathbf{A} = (a_{ij})_{n \times n} \in \mathbb{C}^{n \times n}$ 为复矩阵，那么 \mathbf{A} 的共轭矩阵定义为

$$\overline{\mathbf{A}} = (\overline{a_{ij}})_{n \times n}.$$

性质 3.4.7. 共轭矩阵的性质 设 $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n \times n}$, $k \in \mathbb{C}$, 则

- $\overline{\mathbf{A} + \mathbf{B}} = \overline{\mathbf{A}} + \overline{\mathbf{B}}$;
- $\overline{(k\mathbf{A})} = \overline{k}\overline{\mathbf{A}}$;
- $\overline{\mathbf{AB}} = \overline{\mathbf{A}}\overline{\mathbf{B}}$;
- $\overline{(\mathbf{A}^{-1})} = (\overline{\mathbf{A}})^{-1}$ 。

我们把转置这个概念拓展一下。

定义 3.4.7. 设矩阵 $\mathbf{A} = (a_{ij})_{n \times n} \in \mathbb{C}^{n \times n}$, 那么矩阵 \mathbf{A} 的共轭转置矩阵为

$$\mathbf{A}^H = \overline{(\mathbf{A}^T)} = (\overline{\mathbf{A}})^T = (\overline{a_{ji}})_{n \times n}.$$

“H”是“Hermitian”的缩写。

例 3.4.12. 设矩阵

$$\mathbf{A} = \begin{pmatrix} 1 \\ -i \end{pmatrix}, \mathbf{B} = \begin{pmatrix} 1+i & 2+i \\ 1-i & 1-2i \end{pmatrix},$$

那么

$$\mathbf{A}^H = \begin{pmatrix} 1 & i \end{pmatrix}, \mathbf{B}^H = \begin{pmatrix} 1-i & 1+i \\ 2-i & 1+2i \end{pmatrix}.$$

共轭转置具有以下性质：

性质 3.4.8. 假设下述和、积、逆有意义，则矩阵的共轭转置具有下述性质：

- 设 $\mathbf{A} \in \mathbb{C}^{n \times m}, \mathbf{B} \in \mathbb{C}^{m \times n}$, 则 $(\mathbf{A} + \mathbf{B})^H = \mathbf{A}^H + \mathbf{B}^H$
- 设 $\mathbf{A} \in \mathbb{C}^{n \times m}, \mathbf{B} \in \mathbb{C}^{m \times k}$, 则 $(\mathbf{AB})^H = \mathbf{B}^H \mathbf{A}^H$
- 设 $\mathbf{A} \in \mathbb{C}^{n \times m}, k \in \mathbb{C}$, 则有 $(k\mathbf{A})^H = \bar{k}\mathbf{A}^H$
- 设 $\mathbf{A} \in \mathbb{C}^{n \times m}$, 则有 $(\mathbf{A}^H)^H = \mathbf{A}$
- 设 $\mathbf{A} \in \mathbb{C}^{n \times n}$ 可逆, 则有 $(\mathbf{A}^{-1})^H = (\mathbf{A}^H)^{-1}$

正如在 \mathbb{R}^n 上能够定义内积 \mathbb{C}^n 上也能定义内积。设 $\mathbf{u}, \mathbf{v} \in \mathbb{C}^n$ \mathbf{u}, \mathbf{v} 的内积定义为

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^H \mathbf{v}$$

这种内积有 Hermite 性即

$$\mathbf{u}^H \mathbf{v} = \overline{\mathbf{v}^H \mathbf{u}}$$

现在就可以将正交矩阵的概念扩展到复数域上形成酉矩阵。

定义 3.4.8. 设矩阵 $\mathbf{U} \in \mathbb{C}^{n \times n}$, 如果矩阵 \mathbf{U} 满足

$$\mathbf{U}^H \mathbf{U} = \mathbf{I},$$

那么我们称 \mathbf{U} 为酉矩阵。

容易知道正交矩阵都是酉矩阵。

例 3.4.13. 矩阵

$$\mathbf{U} = \begin{pmatrix} 2^{-1/2} & 2^{-1/2} & 0 \\ -2^{-1/2}i & 2^{-1/2}i & 0 \\ 0 & 0 & i \end{pmatrix}$$

是一个酉矩阵。

酉矩阵具有以下性质：

性质 3.4.9. 设矩阵 $\mathbf{U} \in \mathbb{C}^{n \times n}$ 是酉矩阵, 那么

- \mathbf{U} 可逆且 $\mathbf{U}^H = \mathbf{U}^{-1}$
- $|\det(\mathbf{U})| = 1$
- \mathbf{U}^H 也是酉矩阵
- $\|\mathbf{U}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$

下面我们给出傅里叶矩阵的定义。

定义 3.4.9. 如果矩阵 $\mathbf{F}_n \in \mathbb{C}^{n \times n}$ 为

$$\mathbf{F}_n = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \omega_n^3 & \omega_n^4 & \dots & \omega_n^{(n-1)} \\ 1 & \omega_n^2 & \omega_n^4 & \omega_n^6 & \omega_n^8 & \dots & \omega_n^{2(n-1)} \\ 1 & \omega_n^3 & \omega_n^6 & \omega_n^9 & \omega_n^{12} & \dots & \omega_n^{3(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \omega_n^{(n-1)} & \omega_n^{2(n-1)} & \omega_n^{3(n-1)} & \omega_n^{4(n-1)} & \dots & \omega_n^{(n-1)^2} \end{pmatrix}$$

其中 $\omega_n \in \mathbb{C}$, 且 $\omega_n = e^{i \frac{2\pi}{n}} = \cos \frac{2\pi}{n} + i \sin \frac{2\pi}{n}$ 是方程 $\omega_n^n = 1$ 的单位根, 那么矩阵 \mathbf{F}_n 称为 n 阶傅里叶矩阵。

显然 \mathbf{F}_n 是对称矩阵, j, k 位置元素为 $F_{jk} = \frac{1}{\sqrt{n}} \omega_n^{jk} = \frac{1}{\sqrt{n}} e^{\frac{2\pi i}{n} jk}$ 。

例 3.4.14. 二阶的傅里叶矩阵为

$$\mathbf{F}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & i^2 \end{pmatrix}$$

四阶的傅里叶矩阵为

$$\mathbf{F}_4 = \frac{1}{\sqrt{4}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 & i^3 \\ 1 & i^2 & i^4 & i^6 \\ 1 & i^3 & i^6 & i^9 \end{pmatrix}$$

定理 3.4.3. 傅里叶矩阵 \mathbf{F}_n 是酉矩阵, 即满足

$$\mathbf{F}_n^H \mathbf{F}_n = \mathbf{I}$$

证明. 设 \mathbf{f}_i 是 \mathbf{F}_n 第 i 列的列向量, 那么

$$\mathbf{f}_i^T \mathbf{f}_i = \left(\frac{1}{\sqrt{n}}\right)^2 \sum_{j=0}^{n-1} \omega_n^{ij} (\bar{\omega}_n^{ij}) = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{ij} \omega_n^{n-ij} = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^n = 1$$

而当 $i \neq k$ 时,

$$\mathbf{f}_i^T \mathbf{f}_k = \left(\frac{1}{\sqrt{n}}\right)^2 \sum_{j=0}^{n-1} \omega_n^{ij} (\bar{\omega}_n^{kj}) = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{ij} \omega_n^{n-kj} = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{n+(i-k)j} = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{(i-k)j}$$

不妨令 $i > k$, 那么我们只需要考察当 $0 < i < n$ 时的 $\frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{ij}$ 。注意到 $\omega_n^i \neq 1$ 是方程 $x^n - 1 = 0$ 的根, 所以将方程左边因式分解可得

$$(x - 1)(x^{n-1} + x^{n-2} + \cdots + x^2 + x + 1) = 0.$$

所以有

$$\sum_{j=0}^{n-1} (\omega_n^i)^j = 0,$$

即有

$$\frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{ij} = 0.$$

□

N 点 DFT (离散傅里叶变换) 表示为乘法 $X = \mathbf{F}_n \mathbf{x}$, 其中 \mathbf{x} 是原始输入信号, \mathbf{F}_n 是 $N \times N$ 平方 DFT 矩阵, X 是信号的 DFT。

本节我们讨论了一些特殊正交矩阵, Haar 矩阵和 Hadamard 矩阵都是实数域上的矩阵。在复数域上, 类似正交矩阵, 我们定义了酉矩阵, 从而得到了傅里叶矩阵。这些矩阵都是信号处理中的常见矩阵。

3.5 阅读材料

本章我们介绍了线性代数的几何, 包括向量的范数和内积、矩阵的范数和内积、矩阵的四个基本空间、投影以及特殊的正交矩阵等。这些概念有助于我们从几何的角度来理解线性代数的基本概念以及在数据科学中的应用。

例如在数据科学, 机器学习和人工智能领域, 内积除了用于度量向量的相似性, 更重要的是可以用于很多分类、回归和降维的方法中, 如核方法 (Scholkopf and Smola, 2002)。核方法展示了这样一个事实上, 即许多线性算法可以纯粹的由内积计算来表示。而且, “核技巧”允许我们计算这些隐含在一个 (潜在无限维) 特征空间中的内积, 我们甚至不需要知道这个特征空间是什么。这允许我们对机器学习中使用的许多算法进行“非线性化”, 如用于降维的核用于降维。此外, 概率回归中的 Gaussian 过程也可归于核方法的范畴。关于核方法和内积的更多细节可参考 (Scholkopf and Smola, 2002) 和本书第 6 章。

范数和内积类似, 在数据科学和机器学习中主要用于度量预测值和真实值的误差, 因此其在机器学习的优化方法中会有重要的应用。除了 L_2 范数, 在过去十多年, L_1 范数在大规模的稀疏数据和信号处理领域大放异彩, 获得很大的成功。详细可参考 (Boyd, Stephen, and Vandenberghe, Lieven. 2004)。

投影经常用于计算机图形学, 例如, 生成阴影。在优化中, 正交投影经常用于 (迭代) 最小化残余误差。这在机器学习中也有应用, 例如, 在线性回归中, 我们希望找到一个 (线性) 函数,

该函数最小化残余误差，即数据到线性函数的正交投影的长度。PCA 也使用投影来对高维数据进行降维。更多的细节可以参考文献 (Bishop, 2006)。

子空间在数据科学，机器学习和信号处理领域有重要的应用，如在信号处理领域可以应用于多重信号分类，子空间白化和实时信号处理（投影逼近子空间跟踪）。

关于这些内容更详细的介绍，可以参考国内外优秀的教科书: Axler (2015) 和 Boyd and Vandenberghe (2018) 等。

习题

习题 3.1. 假设向量 β 可以经向量组 $\alpha_1, \alpha_2, \dots, \alpha_r$ 线性表出，证明：表示法是唯一的充分必要条件是 $\alpha_1, \alpha_2, \dots, \alpha_r$ 线性无关。

习题 3.2. 设 $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3$, 求方程式 $A\mathbf{x} = 12\mathbf{x}$ 所有的解，其中：

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

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

习题 3.3. 求出下列非齐次线性方程 $Ax=b$ 中所有解的集合 S ，其中 A 和 b 定义如下：

(1)

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

(2)

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

习题 3.4. 设 $A = \begin{pmatrix} 3 & 1 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 1 & -1 \\ 2 & -1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$

计算 $AB, AB - BA$

草稿请勿外传

习题 3.5. 计算:

$$(1) \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^n \quad (2) \quad \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix}^n \quad (3) \quad \begin{pmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{pmatrix}^n$$

习题 3.6. 求 A^{-1} , 设:

$$(1) \quad A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (2) \quad A = \begin{pmatrix} 2 & 2 & 3 \\ 1 & -1 & 0 \\ -1 & 2 & 1 \end{pmatrix}$$

习题 3.7. 证明 $\alpha_1, \alpha_2, \dots, \alpha_r$ (其中 $\alpha_1 \neq 0$) 线性相关的充分必要条件是至少有一 α_i ($1 < i \leq s$) 可被 $\alpha_1, \alpha_2, \dots, \alpha_{i-1}$ 线性表出。

习题 3.8. 设

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

将向量 $\mathbf{y} = \begin{bmatrix} 1 \\ -2 \\ 5 \end{bmatrix}$ 表示成 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 的线性组合。

习题 3.9. 判断下列向量是否线性无关。

(1)

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

(2)

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

习题 3.10. 把向量 β 表成向量 $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ 的线性组合:

(1) $\beta = (1, 2, 1, 1), \quad \alpha_1 = (1, 1, 1, 1), \quad \alpha_2 = (1, 1, -1, -1), \quad \alpha_3 = (1, -1, 1, -1),$
 $\alpha_4 = (1, -1, -1, 1);$

(2) $\beta = (0, 0, 0, 1), \quad \alpha_1 = (1, 1, 0, 1), \quad \alpha_2 = (2, 1, 3, 1), \quad \alpha_3 = (1, 1, 0, 1), \quad \alpha_4 = (0, 1, -1, -1);$

禁高请勿外传

习题 3.11. 设 $\alpha_1 = (1, -1, 2, 4)$, $\alpha_2 = (0, 3, 1, 2)$, $\alpha_3 = (3, 0, 7, 14)$, $\alpha_4 = (1, -1, 2, 0)$, $\alpha_5 = (2, 1, 5, 6)$.

(1) 证明: α_1, α_2 线性无关;

(2) 把 α_1, α_2 扩充成一极大线性无关组。

习题 3.12. 计算下列矩阵的秩:

$$(1) \begin{pmatrix} 0 & 1 & 1 & -1 & 2 \\ 0 & 2 & -2 & -2 & 0 \\ 0 & -1 & -1 & 1 & 1 \\ 1 & 1 & 0 & 1 & -1 \end{pmatrix}, \quad (2) \begin{pmatrix} 1 & -1 & 2 & 1 & 0 \\ 2 & -2 & 4 & -2 & 0 \\ 3 & 0 & 6 & -1 & 1 \\ 0 & 3 & 0 & 0 & 1 \end{pmatrix}, \quad (3) \begin{pmatrix} 14 & 12 & 6 & 8 & 2 \\ 6 & 104 & 21 & 9 & 17 \\ 7 & 6 & 3 & 4 & 1 \\ 35 & 30 & 15 & 20 & 5 \end{pmatrix}$$

习题 3.13. 判断下列映射是否是线性映射。

(1) $a, b \in \mathbb{R}$

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

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

其中 $L^1([a, b])$ 表示 $[a, b]$ 上的可积函数集。

(2)

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

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

其中 $k \geq 1, C^k$ 表示连续可微的 k 次的集合, C^0 表示连续函数集。

(3)

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

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

(4)

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

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

(5) $\theta \in [0, 2\pi]$.

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

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

习题 3.14. 已知 E 是一个向量空间, 令 f 和 g 是 E 上的自同态映射, 且 $f \circ g = \text{id}_E$ 。证明 $f = \ker(g \circ f)$, $\text{Im } g = \text{Im}(g \circ f)$ 和 $\ker(f) \cap \text{Im}(g) = \{\mathbf{0}_E\}$ 。

习题 3.15. 对于 $\Phi: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ 的变换矩阵是

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

(1) 求 $\ker(\Phi), \text{Im}(\Phi)$ 。

(2) 确定关于基 B 的变换矩阵 $\tilde{\mathbf{A}}_\Phi$ 。

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

习题 3.16. 已知 \mathbb{R}^3 标准基下向量 c_1, c_2, c_3

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

令 $C = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ 。

(1) 证明 C 是 \mathbb{R}^3 的基。

(2) $C' = (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ 是 \mathbb{R}^3 的标准基。计算从 C' 到 C 的过渡矩阵 P_2 。

习题 3.17. 考虑 \mathbb{R}^2 中的四个向量 $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}'_1, \mathbf{b}'_2$ 。令 $B = (\mathbf{b}_1, \mathbf{b}_2)$ 并且 $B' = (\mathbf{b}'_1, \mathbf{b}'_2)$

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

求 B' 到 B 的过渡矩阵。

习题 3.18. 判断如下的两个矩阵的正定性:

$$\mathbf{A}_1 = \begin{pmatrix} 9 & 6 \\ 6 & 5 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} 9 & 6 \\ 6 & 3 \end{pmatrix}$$

习题 3.19. 证明 $\mathbf{x}^\top \mathbf{A} \mathbf{x} = \text{Tr}(\mathbf{x}^\top \mathbf{A} \mathbf{x})$ 和 $\mathbf{x}^\top \mathbf{A} \mathbf{x} = \text{Tr}(\mathbf{A} \mathbf{x} \mathbf{x}^\top)$

习题 3.20. t 取什么值时, 下列二次型是正定的:

$$(1) \quad x_1^2 + x_2^2 + 5x_3^2 + 2tx_1x_2 - 2x_1x_3 + 4x_2x_3$$

$$(2) \quad x_1^2 + 4x_2^2 + x_3^2 + 2tx_1x_2 + 10x_1x_3 + 6x_2x_3$$

习题 3.21. 设 $\mathbf{A} = \begin{pmatrix} 1 & 4 & 2 \\ 0 & -3 & 4 \\ 0 & 4 & 3 \end{pmatrix}$ 求 \mathbf{A}^k

习题 3.22. 证明: 如果 \mathbf{A} 可逆, 证明: \mathbf{AB} 与 \mathbf{BA} 相似

习题 3.23. 设一个线性映射 $f : R^n \rightarrow R^m$, 如何计算(唯一)矩阵 A , 对每一个 $x \in R^n$ 都使 $f(x) = Ax$ 成立, 可以自己确定 f 在适当向量处的值表示。

习题 3.24. 已知线性映射

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

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

(1) 计算 A_Φ

(2) 计算 $\text{rank}(A_\Phi)$

(3) 计算 Φ 的核与像。核的维数 $\dim(\ker(\Phi))$ 和像的维数 $\dim(\text{Im}(\Phi))$ 是多少?

习题 3.25. 证明: 在 \mathbb{R}^n 上, 当且仅当对称矩阵 A 是正定矩阵时, 函数 $f(x) = (x^T A x)^{\frac{1}{2}}$ 是一个向量范数。

习题 3.26. 令 $A \in \mathbb{R}^{n \times n}$, $p(\lambda) \doteq \det(\lambda I_n - A) = \lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1\lambda + c_0$ 是 A 的特征多项式。

(1) 假设 A 是可对角化的。证明:

$$p(A) = A^n + c_{n-1}A^{n-1} + \cdots + c_1A + c_0I_n = 0$$

(2) 证明: 在一般情况下 $p(A) = 0$ 是成立的, 即对于不可对角方阵也是成立的。

习题 3.27. 斐波那契数列前两项为 1, 自第三项起为之前两项之和。 a_i 表示斐波那契数列的第 i 项。记向量

$$\alpha_i = \begin{bmatrix} a_i \\ a_{i+1} \end{bmatrix} \quad i = 1, 2, \dots$$

设 A 为 2×2 常量矩阵使得 $\alpha_{i+1} = A\alpha_i$:

(1) 写出矩阵 A

(2) 计算 A^n 并给出 a_n 的通项公式。

参考文献

- [1] Axler, Sheldon. 2015. Linear Algebra Done Right. third edn. Springer.
- [2] Boyd, Stephen, and Vandenberghe, Lieven. 2018. Introduction to Applied Linear Algebra. Cambridge University Press.
- [3] Giuseppe Calafiore and Laurent El Ghaoui. 2014. Optimization Models. Cambridge University Press.

- [4] Stoer, Josef, and Burlirsch, Roland. 2002. Introduction to Numerical Analysis. Springer.
- [5] Scholkopf, Bernhard, and Smola, Alexander J. 2002. Learning with Kernels—Support Vector Machines, Regularization, Optimization, and Beyond. Adaptive Computation and Machine Learning. Cambridge, MA, USA: The MIT Press.
- [6] Scholkopf, Bernhard, Smola, Alexander, and Müller, Klaus-Robert. 1997. Kernel principal component analysis. Pages 583-588 of: International Conference on Artificial Neural Networks. Springer.
- [7] Boyd, Stephen, and Vandenberghe, Lieven. 2004. Convex Optimization. Cambridge 6721 University Press.
- [8] Bishop, C. M. 2006. Pattern recognition and machine learning. Springer.

草稿请勿外传

第四章 矩阵分解

在第二章中，我们介绍了处理和测量向量的方法、投影和线性映射。线性映射和线性变换可以很方便地用矩阵进行描述。另外，数据科学中的数据通常也用矩阵形式进行表达，例如图片、关系网络等。在这一章节中，我们主要介绍有关矩阵的另一大内容——矩阵的分解。

矩阵，我们可以把它们看作存放了数据的表格，也可以看作是对向量进行线性变换。若将矩阵视为数据表格，可以将矩阵看作若干“简单”的数据表格的线性组合。每个简单表格的系数有时可以反映其在组合中的“重要程度”。则可以将矩阵看作若干个“简单”线性变换的乘积。在第二章中，我们介绍了矩阵的秩 1 分解，并且提到这样的分解是不唯一的。对什么样的数据表格是“简单”的，什么样的线性变换是“简单”的可以有不同的理解方式，从而得到不同的矩阵分解方式，这些分解有助于我们了解原本复杂的高维矩阵的某些性质。本章将一一介绍常用的矩阵分解方式，包括 LU 分解、正交三角（QR）分解、Cholesky 分解、谱分解和奇异值分解（SVD）。本章的内容概览图如下：

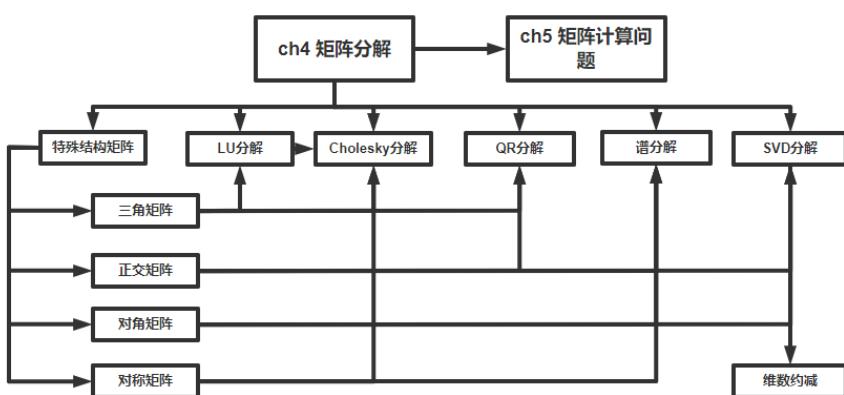


图 4.1: 本章内容概览

4.1 数学中常见的具有特殊结构的矩阵

方阵

首先我们来看看方阵。

若矩阵 $A \in \mathbb{R}^{n \times n}$, 即行数与列数都等于 n 的叫做 n 阶方阵。

方阵是非常特殊的矩阵。在大学线性代数当中大多数讨论都主要是针对方阵进行的。

比如

- 只有方阵才可以计算行列式。
- 只有方阵才可能有逆矩阵, 且方阵有逆矩阵当且仅当方阵满秩。
- 只有方阵才有伴随矩阵。
- 只有方阵才有特征值, 特征向量等概念。

对方阵中的元素做一些简单的约束便可得一类特殊一点的方阵, 也即对称矩阵。

对称矩阵和半正定矩阵

以主对角线为对称轴, 对应各元素相等的矩阵是对称矩阵。即矩阵 A 是对称矩阵, 当且仅

当

$$A^T = A$$

对对称矩阵我们定义一些约束, 则可得半正定矩阵和正定矩阵。如果对称矩阵 $A \in \mathbb{R}^{n \times n}$ 且对任意 $x \in \mathbb{R}^n$ 有

$$xA^Tx \geq 0$$

则称 A 为半正定矩阵, 记为 $A \succeq 0$ 。进一步, 若对任意的 $0 \neq x \in \mathbb{R}^n$ 有

$$xA^Tx > 0$$

则称 A 为正定矩阵, 记为 $A \succ 0$ 。

对称矩阵和正定矩阵在数据科学中是具有重要地位的, 它们是很多数据表示和建模的矩阵, 我们在之后的章节会介绍几个例子。

对角矩阵

上面介绍的几个矩阵是我们要分解的主要对象。接下来我们介绍几个用于表示分解的具有简单结构和性质的矩阵。首先我们来看看最简单的对角矩阵。非对角元素都为零元素的方阵叫做对角矩阵。

$n \times n$ 的对角矩阵可以记为 $A = \text{diag}(a) = \text{diag}(a_1, a_2, \dots, a_n)$ 。

这里 a 是 n 维向量, 包含了矩阵 A 的全部对角元素。

$$\mathbf{A} = \text{diag}(a_1, a_2, \dots, a_n) = \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{pmatrix}$$

容易验证，对于对角矩阵：

$$\mathbf{A}^k = \begin{pmatrix} a_1^k & & \\ & \ddots & \\ & & a_n^k \end{pmatrix}$$

如果 \mathbf{A} 是可逆矩阵，即矩阵 \mathbf{A} 的对角元都不为零，我们有：

$$\mathbf{A}^{-1} = \begin{pmatrix} \frac{1}{a_1} & & \\ & \ddots & \\ & & \frac{1}{a_n} \end{pmatrix}$$

三角矩阵

接下来我们看看三角矩阵。三角矩阵是对角元下方或对角元上方全是零的方阵。

定义 4.1.1. 若矩阵 \mathbf{A} 的所有元素满足 $i > j, a_{ij} = 0$ ，则称 \mathbf{A} 为上三角矩阵

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ & \ddots & \vdots \\ & & a_{nn} \end{bmatrix}$$

定义 4.1.2. 若矩阵 \mathbf{A} 的所有元素满足 $i < j, a_{ij} = 0$ ，则称 \mathbf{A} 为下三角矩阵

$$\mathbf{A} = \begin{bmatrix} a_{11} & & \\ \vdots & \ddots & \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$$

三角矩阵性质

- 三角矩阵主对角线上元素均非零 \iff 三角矩阵可逆
- 上三角矩阵的乘积还是上三角矩阵
- 若上三角矩阵可逆则其逆矩阵也是上三角矩阵
- 下三角矩阵的乘积还是下三角矩阵
- 若下三角矩阵可逆则其逆矩阵也是下三角矩阵
- 设矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 为三角矩阵， $k \in \mathbb{Z}$ ，那么矩阵 \mathbf{A}^k 主对角线上的元素 $(\mathbf{A}^k)_{ii} = (\mathbf{A}_{ii})^k, i = 1, 2, \dots, n$

正交矩阵

再接下来我们看看正交矩阵。正交矩阵因为是有列正交性，以其作为基底，则可以大大减少我们的计算量。**正交矩阵** (orthogonal matrix) 指行向量和列向量是分别标准正交的方阵，即

$$\mathbf{A}^T \mathbf{A} = \mathbf{A} \mathbf{A}^T = \mathbf{I}$$

从定义上可以看出

$$\mathbf{A}^{-1} = \mathbf{A}^T$$

正交矩阵求逆，只需对矩阵转置即求得矩阵的逆。

正交矩阵的性质

正交性 设 $\mathbf{A} = [a_1, \dots, a_n]$ ，并且 \mathbf{A} 是一个正交矩阵，那么

$$\mathbf{a}_i^T \mathbf{a}_j = \begin{cases} 1 & \text{如果 } i = j \\ 0 & \text{如果 } i \neq j \end{cases}$$

和范数有关的性质 如果矩阵 $\mathbf{U} \in \mathbb{R}^{m \times m}, \mathbf{V} \in \mathbb{R}^{n \times n}$ 是正交矩阵， $\mathbf{M} \in \mathbb{R}^{m \times n}, \mathbf{x} \in \mathbb{R}^m$

- $\|\mathbf{U}\|_2 = 1, \|\mathbf{U}\|_F = \sqrt{m}$
- $\|\mathbf{Ux}\|_2 = \|\mathbf{x}\|_2, \|\mathbf{Ux}\|_F = \|\mathbf{x}\|_F$
- $\|\mathbf{UMV}\|_2 = \|\mathbf{M}\|_2, \|\mathbf{UMV}\|_F = \|\mathbf{M}\|_F$

Dyads

Dyads (并向量或单纯矩阵或秩 1 矩阵)

定义 4.1.3. 矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 如果具有如下形式：

$$\mathbf{A} = \mathbf{u} \mathbf{v}^T$$

其中向量 $\mathbf{u} \in \mathbb{R}^m, \mathbf{v} \in \mathbb{R}^n$ ，则称其为 dyad，也称为并向量或单纯矩阵。如果 \mathbf{u} 和 \mathbf{v} 不为零，则我们称其为秩 1 矩阵。

如果 \mathbf{v}, \mathbf{u} 具有相同维度，则 dyad $\mathbf{A} = \mathbf{u} \mathbf{v}^T$ 就是一个方阵。

dyad 作为线性映射 一个 dyad $\mathbf{A} = \mathbf{u} \mathbf{v}^T$ 对于输入向量 $\mathbf{x} \in \mathbb{R}^n$ 有如下作用：

$$\mathbf{Ax} = (\mathbf{u} \mathbf{v}^T) \mathbf{x} = (\mathbf{v}^T \mathbf{x}) \mathbf{u}$$

- 因为 $A_{ij} = u_i v_j$ ；所以每一行（列）是对应的列（行）的缩放，其中“缩放”由向量 \mathbf{u} (\mathbf{v}) 给出。
- 对于一个给定的 $\mathbf{A} = \mathbf{u} \mathbf{v}^T$ ，由对应的线性映射 $\mathbf{x} \rightarrow \mathbf{Ax}$ 可知，无论输入 \mathbf{x} 是什么，输出向量方向始终与 \mathbf{u} 相同。因此，输出向量是 \mathbf{u} 的一个缩放，并且缩放量为 $\mathbf{v}^T \mathbf{x}$ ，故取决于向量 \mathbf{v} 。

- 对于一个 dyad $\mathbf{A} = \mathbf{u}\mathbf{v}^T$, 如果 \mathbf{u} 和 \mathbf{v} 不为零, 则其秩为 1, 因为它的像空间都是由 \mathbf{u} 生成的, 因此把 $\mathbf{A} = \mathbf{u}\mathbf{v}^T$ 称为秩 1 矩阵。

dyad 的特征值和特征向量 方的 dyad ($m = n$) 有唯一的非零特征值 $\lambda = \mathbf{v}^T \mathbf{u}$ 与对应的特征向量 \mathbf{u} 。

dyad 的正规化 对于一个 dyad $\mathbf{A} = \mathbf{u}\mathbf{v}^T$, 我们可以利用欧几里得范数单位化 \mathbf{u} 和 \mathbf{v} , 并且用一个系数来衡量 dyad 的大小, 以此来标准化 dyad, 也即任何 dyad 都可以写成如下正规化的形式:

$$\mathbf{A} = \mathbf{u}\mathbf{v}^T = (\|\mathbf{u}\|_2 \cdot \|\mathbf{v}\|_2) \frac{\mathbf{u}}{\|\mathbf{u}\|_2} \frac{\mathbf{v}^T}{\|\mathbf{v}\|_2} = \sigma \tilde{\mathbf{u}} \tilde{\mathbf{v}}^T$$

其中 $\sigma > 0$, 并且 $\|\tilde{\mathbf{u}}\| = \|\tilde{\mathbf{v}}\| = 1$ 。

分块矩阵

分块矩阵 任何矩阵都可以分成具有相容维的若干块或子矩阵的分块形式:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

所谓相容维就是指 $\mathbf{A}_{11}, \mathbf{A}_{12}$ 行数一样, $\mathbf{A}_{11}, \mathbf{A}_{21}$ 列数一样。当 \mathbf{A} 是方阵, 并且 $\mathbf{A}_{12} = \mathbf{O}, \mathbf{A}_{21} = \mathbf{O}$, 那么称 \mathbf{A} 为块对角矩阵:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{O} \\ \mathbf{O} & \mathbf{A}_{22} \end{bmatrix}$$

接下来我们看看分块对角矩阵特征值和块对角矩阵特征值的关系。若 \mathbf{A} 为块对角矩阵, 用 $\lambda(\mathbf{A})$ 表示 \mathbf{A} 的特征值集合, 显然, 它是 \mathbf{A}_{11} 和 \mathbf{A}_{22} 特征值集合的并集。

$$\mathbf{A} \text{ 为块对角矩阵} \implies \lambda(\mathbf{A}) = \lambda(\mathbf{A}_{11}) \cup \lambda(\mathbf{A}_{22})$$

一个块对角矩阵是可逆的, 当且仅当它的每个对角块是可逆的, 并且

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{O} \\ \mathbf{O} & \mathbf{A}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}_{11}^{-1} & \mathbf{O} \\ \mathbf{O} & \mathbf{A}_{22}^{-1} \end{bmatrix}$$

除块对角矩阵, 还有分块三角矩阵。分块方阵 \mathbf{A} , 如果 $\mathbf{A}_{21} = \mathbf{O}$, 称之为分块上三角矩阵; 如果 $\mathbf{A}_{12} = \mathbf{O}$, 称之为分块下三角矩阵。

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{O} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \quad \text{分块下三角矩阵}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{O} & \mathbf{A}_{22} \end{bmatrix} \quad \text{分块上三角矩阵}$$

若 \mathbf{A} 为分块三角矩阵, 用 $\lambda(\mathbf{A})$ 表示 \mathbf{A} 的特征值集合, 同样有:

$$\mathbf{A} \text{ 为分块(上或下)三角矩阵} \implies \lambda(\mathbf{A}) = \lambda(\mathbf{A}_{11}) \cup \lambda(\mathbf{A}_{22})$$

下面我们给出分块三角矩阵的逆和分块矩阵的逆。非退化的分块三角矩阵的逆可以表示为：

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{O} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}_{11}^{-1} & \mathbf{O} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{A}_{22}^{-1} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{O} & \mathbf{A}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}_{11}^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ \mathbf{O} & \mathbf{A}_{22}^{-1} \end{bmatrix}$$

这可以通过矩阵乘积来验证上述公式。当然也可以通过对下列分块矩阵的逆矩阵公式取特殊情形来得到。所以接下来就看一下分块矩阵的逆的求解。

考虑非退化分块矩阵

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}$$

其中 \mathbf{A}_{11} 和 \mathbf{A}_{22} 是方阵并且可逆。令 $\mathbf{S}_1 = \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$, $\mathbf{S}_2 = \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}$, 我们可以通过待定系数法来求解 \mathbf{A}^{-1} , 得到

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}^{-1} = \begin{pmatrix} \mathbf{S}_1^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{S}_2^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{S}_1^{-1} & \mathbf{S}_2^{-1} \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{S}_1^{-1} & -\mathbf{S}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{S}_2^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{S}_2^{-1} \end{pmatrix}$$

矩阵求逆引理 假设 \mathbf{A}_{11} 和 \mathbf{A}_{22} 分别是 $n_{\mathbf{A}_{11}} \times n_{\mathbf{A}_{11}}$ 和 $n_{\mathbf{A}_{22}} \times n_{\mathbf{A}_{22}}$ 阶方阵并且可逆, \mathbf{A}_{12} 和 \mathbf{A}_{21} 分别是 $n_{\mathbf{A}_{11}} \times n_{\mathbf{A}_{22}}$ 和 $n_{\mathbf{A}_{22}} \times n_{\mathbf{A}_{11}}$ 阶矩阵, 则如下等式成立:

$$(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21})^{-1} = \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1}\mathbf{A}_{12}(\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12})^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} \quad (4.1)$$

上式也即 \mathbf{S}_1 的逆的表达式。类似的也可得 \mathbf{S}_2 的逆的表达式。

令矩阵求逆引理中 $\mathbf{A}_{11} = \mathbf{E}, \mathbf{A}_{12} = -\mathbf{F}, \mathbf{A}_{22}^{-1} = \mathbf{G}, \mathbf{A}_{21} = \mathbf{H}$ 可得以下 Woodbury 公式。

Woodbury 公式 假设 \mathbf{E} 和 \mathbf{G} 分别是 $n_{\mathbf{E}} \times n_{\mathbf{E}}$ 和 $n_{\mathbf{G}} \times n_{\mathbf{G}}$ 阶方阵并且可逆, \mathbf{F} 和 \mathbf{H} 分别是 $n_{\mathbf{E}} \times n_{\mathbf{G}}$ 和 $n_{\mathbf{G}} \times n_{\mathbf{E}}$ 阶矩阵, 则如下等式成立:

$$(\mathbf{E} + \mathbf{FGH})^{-1} = \mathbf{E}^{-1} - \mathbf{E}^{-1}\mathbf{F}(\mathbf{G}^{-1} + \mathbf{HE}^{-1}\mathbf{F})^{-1}\mathbf{HE}^{-1}$$

矩阵求逆引例中的公式和 Woodbury 公式本质上是同一个公式, 如果我们对公式中的四个矩阵取特殊情形还可得一个著名的公式。

秩1扰动

Sherman-Morrison 公式 设矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ 如果我们令矩阵求逆引理公式(4.1)中

$$\mathbf{A}_{11} = \mathbf{A}, \mathbf{A}_{12} = \mathbf{u}, \mathbf{A}_{22} = -1, \mathbf{A}_{21} = \mathbf{u}^T$$

则我们可以得到如下等式:

$$(\mathbf{A} + \mathbf{uv}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{uv}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}$$

这个式子让我们能够计算矩阵 A 的秩 1 扰动的逆，并且计算仅仅依赖于 A 的逆。Sherman-Morrison 公式可用于拟牛顿迭代法的 BFGS 公式的推导。

一个更有趣的性质是矩阵 A 的秩 1 扰动和原矩阵的秩的变化不超过 1。这个事实不仅仅对方阵成立，对一般的矩阵也成立。我们有如下定理

定理 4.1.1. 令 $A \in \mathbb{R}^{n \times m}$, $q \in \mathbb{R}^m$, $p \in \mathbb{R}^n$ 则有

$$|\text{rank}(A) - \text{rank}(A + qp^T)| \leq 1$$

这个定理证明可利用线性代数基本定理来实现。这个定理在 Matrix Completion 类问题中有重要应用，比如欧几里得距离矩阵的完备化问题。

4.2 数据科学中常见的矩阵

现实世界对象，除了用数值型向量数据表示之外，也可以网络和图进行表达。事实上，网络和图是表示现实世界各种对象关系和相互作用过程表达的数据结构。比如社交网络、通信网络表达人与人间的相互关系，而蛋白质相互作用网络表达蛋白质间的相互作用关系，甚至病毒传播、单词共现、图像都可以看做一个网络。

图的矩阵

网络可以抽象出图结构。图结构可以说是无处不在。通过对它们的分析，我们可以深入了解社会结构、语言和不同的交流模式，因此图一直是学界研究的热点。图是点和边的集合，是网络表达的结构化和抽象化。现实世界的对象可以看成网络和图中的“点”，关系或相互作用可以通过点与点相连的“边”以及给边赋予“权重”和“方向”来进一步表达关系的“远近亲疏、重要程度和因果关系”。图一般可以按照边是否有向分为无向图和有向图；按照边是否有权重分为无权图和加权图；按照点与点的连接关系分为完全图，二分图等。

从数据科学的角度看，图分析任务包括节点分类、链接预测、聚类、降维或可视化等。实现任务的相应模型包括随机游走、相似性方法、最大似然和概率模型、属性基方法、嵌入方法等。我们以谱聚类方法为例，介绍图和矩阵的关系。

例 4.2.1. 设有数据集 $\mathbb{X} = \{(1, 3), (1, 4), (2, 4), (3, 2), (2, 1), (3, 1)\}$ ，如右图 (a) 所示。我们希望能够通过某一种方式将这 6 个点自动地分成两类，如右图 (b) 所示。我们可以将每一个顶点和它距离最近的 3 个顶点进行连接得到图 (c)。从而将问题转化为研究图上顶点聚类。

谱聚类的基本思想是：

1. 把所有的数据看做空间中的点，这些点之间可以用边连接起来，形成一个图，
2. 距离较远的两个点之间的边权重值较低，而距离较近的两个点之间的边权重值较高，

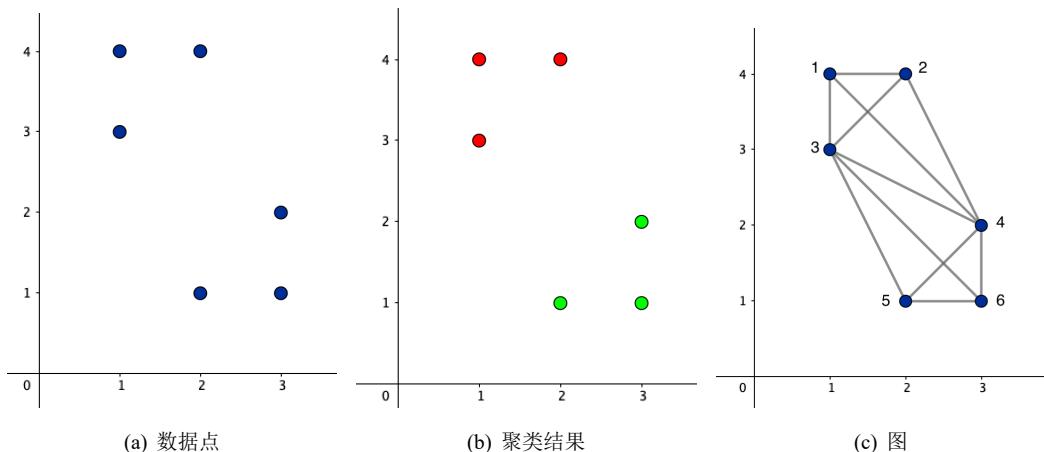


图 4.2: 谱聚类

3. 通过对所有数据点组成的图进行切图，让切图后不同的子图间边权重和尽可能的低，而子图内的边权重和尽可能的高，从而达到聚类的目的。

谱聚类解决如何发现并表示点与点之间，点与边之间关系的问题。

我们可以用以下术语描述单个点、边之间的关系：

- 关联 (incidence): 点与边的关系
- 邻接 (adjacent): 点与点的关系
- 度 (degree): 相邻节点的数量
- 路径、回路、连通
- ...

这样我们可以用以下矩阵描述多个点、边的关系：

- 关联矩阵
- 邻接矩阵
- 度矩阵
- 拉普拉斯矩阵
- ...

大部分图分析任务中的模型可以直接定义在原始图的邻接矩阵或由邻接矩阵和度矩阵导出的拉普拉斯矩阵上。

4.2.1 图的基本概念回顾

图是由一些节点和连接这些节点的边组成的离散结构。

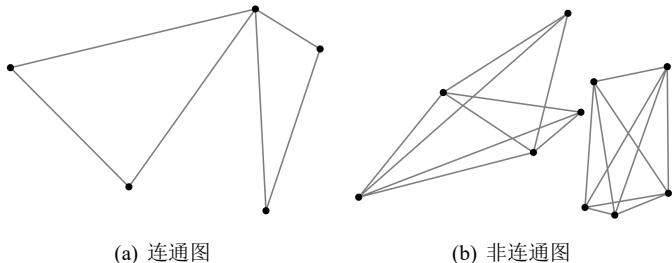
定义 4.2.1. 一张图 G 是一个二元组, $G = (\mathbb{V}, \mathbb{E})$ 是由节点集合 \mathbb{V} 和边集 \mathbb{E} 组成的。其中 \mathbb{E} 中的元素是一个二元对 $\{x, y\}$, 其中 $x, y \in \mathbb{V}$ 。

- 对于无向图而言, $\{x, y\}$ 是无序对, $\{x, y\}$ 和 $\{y, x\}$ 是 \mathbb{E} 中的同一个元素, 表示点 x 和 y 有一条边相连。
- 对于有向图, $\{x, y\}$ 表示有一条由 x 指向 y 的有向边, 和 $\{y, x\}$ 是 \mathbb{E} 中不同的元素。
- 如果图 $G = (\mathbb{V}, \mathbb{E})$ 中每一条边 $\{v_i, v_j\}$ 都被赋予一个权重 w_{ij} , 则称这样的图为加权图或赋权图。

在实际问题中, 权重 w_{ij} 通常是具有某种含义的数值, 比如在聚类中是衡量节点远近关系的距离度量数值。本节讨论的图都是简单图。

定义 4.2.2. 设 n 为正整数, $G = (\mathbb{V}, \mathbb{E})$ 为一简单图。我们称图中的一条长度为 n 的通路为 n 条边 e_1, e_2, \dots, e_n 的序列。其中 $e_1 = \{v_0, v_1\}, e_2 = \{v_1, v_2\}, \dots, e_n = \{v_{n-1}, v_n\}$ 我们可以用顶点序列 v_0, v_1, \dots, v_n 来表示这条通路。如果 $v_0 = v_n$ 我们则称这条通路为一条回路。如果通路 v_0, v_1, \dots, v_n 中 v_1, v_2, \dots, v_n 是互异的, 那么我们称这条通路为简单通路。

定义 4.2.3. 设 $G = (\mathbb{V}, \mathbb{E})$ 为一简单图。如果 $\forall u_1, u_2 \in \mathbb{V}$ 都存在一条通路 v_0, v_1, \dots, v_n 使得 $v_0 = u_1, v_n = u_2$ 。我们则称图 G 是连通的。

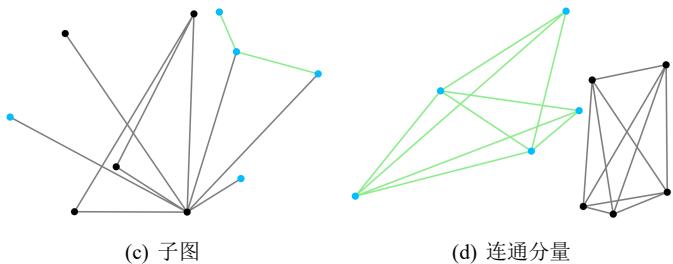


定义 4.2.4. 设图 $G = (\mathbb{V}_G, \mathbb{E}_G), H = (\mathbb{V}_H, \mathbb{E}_H)$ 如果 $\mathbb{V}_H \subseteq \mathbb{V}_G$ 且 $\mathbb{E}_H \subseteq \mathbb{E}_G$, 那么我们称图 H 为 G 的子图。

定义 4.2.5. 设图 $H = (\mathbb{V}_H, \mathbb{E}_H)$ 是图 $G = (\mathbb{V}_G, \mathbb{E}_G)$ 的子图。如果 $\forall v \in \mathbb{V}_H, u \in \mathbb{V}_G / \mathbb{V}_H$ 都满足 $\{v, u\} \notin \mathbb{E}_G$ 则称 H 是图 G 的一个连通分量。

定理 4.2.1. 如果图 $G = (\mathbb{V}, \mathbb{E})$ 是一连通图。那么图 G 有唯一的连通分量为自身。

本节如无特殊说明, 一般讨论的是连通图, 只有一个连通分量。



4.2.2 有向图相关的矩阵

有向图相关的矩阵：关联矩阵

定义 4.2.6. 设有向图 $G = \langle \mathbb{V}, \mathbb{E} \rangle$, 所有顶点的排列为 v_1, v_2, \dots, v_m , 其中 $m = |\mathbb{V}|$, 所有边的排列为 e_1, e_2, \dots, e_n , 其中 $n = |\mathbb{E}|$, 用 b_{ij} 表示顶点 v_i 与边 e_j 关联的次数, 其中 b_{ij} 定义为

$$b_{ij} = \begin{cases} 1 & v_i \text{ 是边 } e_j \text{ 的起点} \\ -1 & v_i \text{ 是边 } e_j \text{ 的终点} \\ 0 & \text{其他} \end{cases}$$

则称所得的矩阵 $B = (b_{ij})_{m \times n}$ 为有向图 G 的关联矩阵。

网络流和关联矩阵

例 4.2.2. 物品、交通、电荷和信息等网络可以表示成一个由 m 个顶点和 n 条有向边构成的有向图。我们可以通过顶点-边的 $m \times n$ 关联矩阵来描述这样的网络。图 4.3 是一个具有 4 个顶点和 4 条边的网络例子，其顶点-边的关联矩阵是：

$$\mathbf{B} = \begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & 0 & -1 & -1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

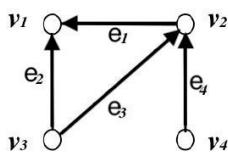


图 4.3: 一个有 4 个顶点的图

关联矩阵的四个基本子空间

性质 4.2.1. 设图 $G = \langle \mathbb{V}, \mathbb{E} \rangle$ 是一个具有 m 个顶点和 n 条边的连通图, 其对应的关联矩阵为 \mathbf{B} , 则关于 \mathbf{B} 的四个基本子空间具有以下性质:

- \mathbf{B} 的左零空间维数为 1, 即 $\text{Null}(\mathbf{B}^T) = 1$, 且 $\text{Null}(\mathbf{B}^T) = \text{span}\{\mathbf{1}\}$ 。
- \mathbf{B} 的行空间维数为 $m - 1$, 即 $\text{Col}(\mathbf{B}^T) = m - 1$, 且 $\text{Col}(\mathbf{B}^T)$ 可由 \mathbf{B}^T 任意 $m - 1$ 个列向量生成。
- \mathbf{B} 的列空间维数为 $m - 1$, 即 $\text{Col}(\mathbf{B}) = m - 1$, 且若 T 是图 G 的一棵生成树, 那么 $\text{Col}(\mathbf{B})$ 可由 T 的关联矩阵的 $m - 1$ 个列向量生成。
- \mathbf{B} 的零空间维数为 $n - m + 1$, 即 $\text{Null}(\mathbf{B}) = n - m + 1$, 这个数等于图 G 中小圈的个数。

4.2.3 无向图相关的矩阵

定义 4.2.7. 设图 $G = (\mathbb{V}, \mathbb{E})$, 我们把图 G 的顶点排列成 $v_1, v_2, \dots, v_n, n = |\mathbb{V}|$ 。用 a_{ij} 表示顶点 v_i 与顶点 v_j 之间的边数, 其中 a_{ij} 定义为

$$a_{ij} = \begin{cases} 1 & \{v_i, v_j\} \in \mathbb{E} \\ 0 & \{v_i, v_j\} \notin \mathbb{E} \end{cases}$$

则称所得的矩阵 $\mathbf{A} = (a_{ij})_{n \times n}$ 为无向图 G 的邻接矩阵。

我们把节点 v_i 相邻节点的数量称为 v_i 的度, 记为 $d(v_i)$, 则 $d(v_i) = \sum_j a_{ij}$, 图 G 的度矩阵 $\mathbf{D} = (d_{ij})_{n \times n}$ 定义为

$$d_{ij} = \begin{cases} d(v_i) & i = j \\ 0 & i \neq j \end{cases}$$

例 4.2.3. 例 4.2.1 中图 4.2.1(c) 所对应的邻接矩阵和度矩阵分别为

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

性质 4.2.2. 设无向图 $G = (\mathbb{V}, \mathbb{E})$ 对应于顶点排列 v_1, v_2, \dots, v_n 的邻接矩阵为 \mathbf{A} , 其中 $n = |\mathbb{V}|$, 则 \mathbf{A} 有以下性质:

- \mathbf{A} 是对称矩阵, 即 $\mathbf{A} = \mathbf{A}^T$ 。
- \mathbf{A} 有 n 个实特征值, 其中一定有最大特征值 λ_1 是单重特征值, 且满足 $\lambda_1 \leq \max_{v \in \mathbb{V}} d(v)$ 。

- 设 v'_1, v'_2, \dots, v'_n 图 G 节点另一种排列, 其对应的邻接矩阵为 A' , 则 A 与 A' 具有相同的特征值。

定义 4.2.8. 设图 $G = (\mathbb{V}, \mathbb{E})$ 的邻接矩阵为 A 。我们则称

- 矩阵 A 的特征值为图 G 的特征值。
- 矩阵 A 的谱为图 G 的谱。

例 4.2.4. 在例 4.2.3 中的邻接矩阵的特征值从小到大分别为

$$\lambda_1 = -1.82842712$$

$$\lambda_2 = \lambda_3 = \lambda_4 = -1$$

$$\lambda_5 = 1$$

$$\lambda_6 = 3.82842712$$

定义 4.2.9. 设无向图 $G = (\mathbb{V}, \mathbb{E})$ 的邻接矩阵和度矩阵分别为 A 和 D , 我们称矩阵 $L = D - A$ 为图 G 的拉普拉斯矩阵。

例 4.2.5. 例 4.2.1 中图 4.2.1(c) 对应的拉普拉斯矩阵为

$$L = \begin{pmatrix} 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ -1 & -1 & 5 & -1 & -1 & -1 \\ -1 & -1 & -1 & 5 & -1 & -1 \\ 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & -1 & 3 \end{pmatrix}$$

定义 4.2.10. 设无向图 $G = (\mathbb{V}, \mathbb{E})$ 的邻接矩阵和度矩阵分别为 A 和 D 。我们称矩阵

$$\tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

为图 G 的正规化的拉普拉斯矩阵。

4.2.4 加权图相关的矩阵

对于例 1 中的聚类例子, 我们的目的是要把它聚成两类。我们从例 5 的拉普拉斯矩阵

$$L = \begin{pmatrix} 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ -1 & -1 & 5 & -1 & -1 & -1 \\ -1 & -1 & -1 & 5 & -1 & -1 \\ 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & -1 & 3 \end{pmatrix}$$

学习外传
情稿草

中可以发现，第3个节点和第4个节点是对称的。也就是说，如果我们仅仅根据这个无向图的拉普拉斯矩阵，我们如果可以把数据聚成 $\{1, 2, 3\}, \{4, 5, 6\}$ 这两类，也就可以聚成 $\{1, 2, 4\}, \{3, 5, 6\}$ 这样两类。但是后者显然不是很合理。这主要因为我们前面定义的邻接矩阵并没有对连接两个顶点的边的长度进行区别考虑。所以，在实际的聚类中，我们要考虑对边进行赋权，构建权重相关的邻接矩阵和拉普拉斯矩阵。

定义 4.2.11. 设加权图 $G = \langle \mathbb{V}, \mathbb{E} \rangle$ ，我们把图 G 的顶点排列成 $v_1, v_2, \dots, v_n, n = |\mathbb{V}|$ ，我们将图 G 的邻接矩阵 $A = (a_{ij})_{n \times n}$ 定义为

$$a_{ij} = \begin{cases} w_{i,j} & \{v_i, v_j\} \in \mathbb{E} \\ 0 & \{v_i, v_j\} \notin \mathbb{E} \end{cases}$$

其中 $w_{i,j}$ 是边 $\{v_i, v_j\}, i, j = 1, \dots, n$ 上的权重。这样的矩阵称为加权图的邻接矩阵。

在实际问题中，权重的定义方式多种多样。在聚类中，一种较为常用的权重定义方式是使用高斯核

$$w_{ij} = e^{-\frac{\|v_i - v_j\|_2^2}{2\sigma^2}}$$

其中 $\|v_i - v_j\|_2$ 表示顶点 v_i 和 v_j 的欧氏距离， σ 是一参数，用于调节顶点间距离到权重的映射值。这样定义的好处是，权重介于0和1之间。

定义 4.2.12. 设加权图 $G = \langle \mathbb{V}, \mathbb{E} \rangle$ ，我们把图 G 的顶点排列成 $v_1, v_2, \dots, v_n, n = |\mathbb{V}|$ ，则顶点 v_i 的带权度数定义为 $d(v_i) = \sum_j w_{ij}$ ，其中 w_{ij} 是边 $\{v_i, v_j\}, i, j = 1, \dots, n$ 上的权重。图 G 的度矩阵 $D = (d_{ij})_{n \times n}$ 定义为

$$d_{ij} = \begin{cases} d(v_i) & i = j \\ 0 & i \neq j \end{cases}$$

定义 4.2.13. 设加权图 $G = \langle \mathbb{V}, \mathbb{E} \rangle$ 的邻接矩阵和度矩阵分别为 A 和 D ，我们称 $L = D - A$ 为图 G 的拉普拉斯矩阵。

定义 4.2.14. 设加权图 $G = \langle \mathbb{V}, \mathbb{E} \rangle$ 的邻接矩阵和度矩阵分别为 A 和 D ，我们称矩阵

$$\tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

为图 G 的正规化拉普拉斯矩阵。

对于一个加权图，如果我们令图上所有边的权重变为原来的 k 倍， $k \neq 0$ 。那么显然对于未正规化的拉普拉斯矩阵 L 将变为 kL 。而正规化的拉普拉斯矩阵 \tilde{L} 则不会发生变化。这是因为

$$\tilde{L}_{ij} = \frac{L_{ij}}{\sqrt{d(v_i)d(v_j)}} = \frac{kL_{ij}}{\sqrt{kd(v_i)kd(v_j)}}$$

因此正规化的拉普拉斯矩阵更为常用，它能够避免权重绝对值大小的影响。

例 4.2.6. 如果使用高斯核来给例 4.2.1 中图 4.2.1(c) 上的边进行赋权，则可得到如下拉普拉斯矩阵：

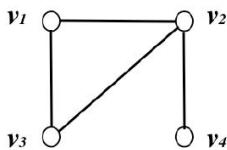
$$\mathbf{L} = \begin{pmatrix} 1.231 & -0.607 & -0.607 & -0.018 & 0 & 0 \\ -0.607 & 1.056 & -0.368 & -0.082 & 0 & 0 \\ -0.607 & -0.368 & 1.157 & -0.082 & -0.082 & -0.018 \\ -0.018 & -0.082 & -0.082 & 1.157 & -0.368 & -0.607 \\ 0 & 0 & -0.082 & -0.368 & 1.056 & -0.607 \\ 0 & 0 & -0.018 & -0.607 & -0.607 & 1.231 \end{pmatrix}$$

性质 4.2.3. 设有向无权图 G 的关联矩阵为 \mathbf{B} ，其对应的无向图的拉普拉斯矩阵为 \mathbf{L} ，则 \mathbf{L} 和 \mathbf{B} 满足以下关系：

$$\mathbf{L} = \mathbf{B}^T \mathbf{B}.$$

例 4.2.7. 在例 2 中图 4.3 对应的拉普拉斯矩阵和关联矩阵满足 $\mathbf{L} = \mathbf{B}\mathbf{B}^T$

$$\mathbf{L} = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & 0 & -1 & -1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



性质 4.2.4. 设正权图 $G = (\mathbb{V}, \mathbb{E})$ 对应于顶点排列 v_1, v_2, \dots, v_n 的拉普拉斯矩阵和正规化拉普拉斯矩阵分别为 \mathbf{L} 和 $\tilde{\mathbf{L}}$ ，其中 $n = |\mathbb{V}|$ ，则 \mathbf{L} 和 $\tilde{\mathbf{L}}$ 有以下性质：

1. \mathbf{L} 和 $\tilde{\mathbf{L}}$ 是对称矩阵，即有 $\mathbf{L} = \mathbf{L}^T$ 和 $\tilde{\mathbf{L}} = \tilde{\mathbf{L}}^T$ 。
2. 对任意的 n 维向量 \mathbf{x} ，有 $\mathbf{x}^T \mathbf{L} \mathbf{x} \geq 0$ 和 $\mathbf{x}^T \tilde{\mathbf{L}}^T \mathbf{x} \geq 0$ ，因而 \mathbf{L} 和 $\tilde{\mathbf{L}}$ 是半正定矩阵。
3. \mathbf{L} 和 $\tilde{\mathbf{L}}$ 的最小特征值为 0，且对应的特征向量分别为 $\mathbf{1}$ 和 $\mathbf{D}^{-\frac{1}{2}} \mathbf{1}$ 。

证明。为了证明第 2 条性质，我们首先证明等式

$$\sum_{\{v_i, v_j\} \in \mathbb{E}} w_{ij} (x_i - x_j)^2 = \mathbf{x}^T \mathbf{L} \mathbf{x}$$

利用拉普拉斯矩阵的定义有

$$\begin{aligned}
 \mathbf{x}^T \mathbf{L} \mathbf{x} &= \mathbf{x}^T \mathbf{D} \mathbf{x} - \mathbf{x}^T \mathbf{A} \mathbf{x} = \sum_{i=1}^n d_i x_i^2 - \sum_{i=1}^n w_{ij} x_i x_j \\
 &= \frac{1}{2} \left(\sum_{i,j=1}^n d_i x_i^2 - 2 \sum_{i,j=1}^n w_{ij} x_i x_j + \sum_{j=1}^n d_j x_j^2 \right) \\
 &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (x_i - x_j)^2 \\
 &= \sum_{\{v_i, v_j\} \in \mathbb{E}} w_{ij} (x_i - x_j)^2
 \end{aligned}$$

那么对于一个正权图来说，无论 \mathbf{x} 取什么， $\mathbf{x}^T \mathbf{L} \mathbf{x}$ 都是非负的。所以 \mathbf{L} 是半正定矩阵。

对于 $\tilde{\mathbf{L}}$ 和任意的 \mathbf{x} 有

$$\begin{aligned}
 &\mathbf{x}^T \tilde{\mathbf{L}} \mathbf{x} \\
 &= \mathbf{x}^T \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} \mathbf{x} \\
 &= (\mathbf{D}^{-\frac{1}{2}} \mathbf{x})^T \mathbf{L} (\mathbf{D}^{-\frac{1}{2}} \mathbf{x}) \geq 0
 \end{aligned}$$

所以 $\tilde{\mathbf{L}}$ 也是半正定矩阵。

对于第 3 条性质，我们只需要分别计算

$$\mathbf{L} \mathbf{1} = 0, \tilde{\mathbf{L}} \mathbf{D}^{-\frac{1}{2}} \mathbf{1} = 0$$

并且综合第 2 条性质 $\mathbf{L}, \tilde{\mathbf{L}}$ 是半正定的，我们可以知道 0 是 $\mathbf{L}, \tilde{\mathbf{L}}$ 最小的特征值，并且对应的特征向量分别为 $\mathbf{1}$ 和 $\mathbf{D}^{-\frac{1}{2}} \mathbf{1}$ 。□

对于谱聚类，我们最终可以将问题转化为求该图对应的拉普拉斯矩阵或正规化拉普拉斯矩阵次小特征值对应的特征向量问题。在得到特征向量后，对其分量进行聚类，聚类结果即为谱聚类的结果。

例 4.2.8. 在例 6 中，我们已经得到例 4.2.1 中图 4.2.1(c) 对应的拉普拉斯矩阵 \mathbf{L} 。可以计算得到它的次小特征值对应的特征向量为

$$\mathbf{x} = (-0.442, -0.421, -0.358, 0.358, 0.421, 0.442)^T$$

我们很容易就可以得到前 3 个节点作为一类，后 3 个节点作为一类。

4.2.5 稀疏矩阵

定义 4.2.15. 一个矩阵中，若数值为零的元素的数目远远多于非零元素的数目，称这样的矩阵为稀疏矩阵。

稀疏矩阵零元素分布常常是没有规律的。

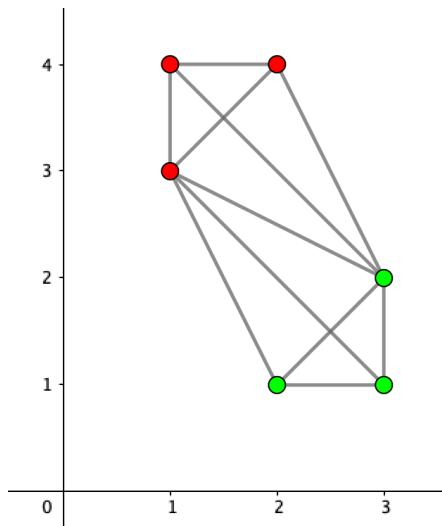


图 4.4: 谱聚类

传外勿请稿草

定义 4.2.16. 当一个矩阵的非零元素数目远远多于零元素数目时，称这样的矩阵为稠密矩阵。

例 4.2.9. 如下几个矩阵我们都可以认为是稀疏矩阵

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; B = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; C = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

邻接矩阵经常是稀疏矩阵。

例 4.2.10. 矩阵

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

是图4.5对应的邻接矩阵。

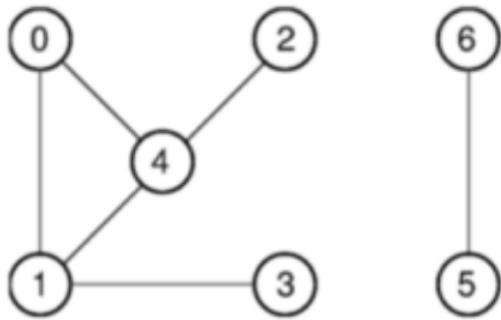


图 4.5: 图与邻接矩阵

有时我们会碰到一些特殊的图，如二分图，此时我们可以给出另外一种将图转换为矩阵的方式。比如图4.6对应的矩阵为

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

二分图有丰富的应用场景，比如

传
外
切
請
稿
草

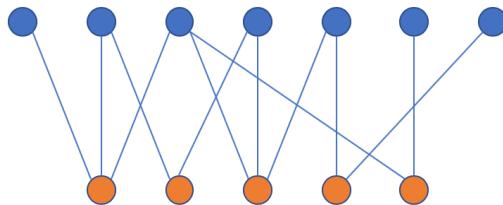


图 4.6: 二分图

- 电子商务：当我们在处理用户-网页、用户-服务、用户-产品等问题时，会遇到这样一个二分图，将这个图转换为矩阵也常常是一个稀疏的矩阵。
- 深度学习：一个多层感知机两层之间也是这样一个二分图。但对于大多数一般的深度神经网络，各层之间的连接矩阵不是一个稀疏矩阵而是一个稠密矩阵。
- 模型压缩：现在有一些剪枝方法，可以修剪掉一些不需要的边，这时便有可能得到一个稀疏矩阵。

在其他实际应用场景中，我们也会接触大量的稀疏矩阵，尤其是超大型的稀疏矩阵，例如：

- 推荐系统：用户只可能对有限商品进行过评价，对于大量的其他商品是没有过评价信息的，因此在用户-商品评价矩阵中有大量的零元素。
- 记数编码：当我们用词汇出现的频率表示文档时，在词汇表中有大量词汇没有在文档中出现过，使得文档矩阵出现很多零元素。
- 图像矩阵：以手写识别数据集为例，只有图像中间区域出现数字，表示该位置有像素点，其他背景都被标记为 0，使表示图像的矩阵有大量的零元素。

低秩矩阵

例 4.2.11. 考虑矩阵

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 & -1 \\ 1 & 3 & 1 & 3 & 1 \\ -1 & -2 & -1 & -2 & -1 \\ 2 & 1 & 2 & 1 & 2 \end{pmatrix}$$

尽管 \mathbf{A} 中没有一个零元素，但是矩阵 \mathbf{A} 的秩只有 2，考虑矩阵 \mathbf{A} 的行空间 $Col(\mathbf{A}^T) = \text{span}\{(1, 1, 1, 1, 1)^T, (-1, 1, -1, 1, -1)^T\}$ 。

在数据科学中，我们会碰到很多大规模但秩很低的稠密矩阵，我们将这样的矩阵称为低秩矩阵。

定义 4.2.17. 设矩阵 $A \in \mathbb{R}^{n \times m}$, 如果矩阵 A 的秩 $\text{rank}(A)$ 远小于 $\max\{n, m\}$, 那么我们称这样的矩阵为低秩矩阵。

之所以考虑的是 $\text{rank}(A)$ 和 $\max\{n, m\}$ 的关系, 是因为在很多时候, 我们都是在非方阵的情况下考虑低秩矩阵的, 并且常常有 n 远小于 m , 或者 m 远小于 n 的情况发生。

例 4.2.12. 考虑矩阵

$$A = \begin{pmatrix} 1 & 2 & 1 & 3 & 1 & 2 & 3 & 4 & 1 & 2 \\ 2 & 1 & 3 & 2 & 1 & 2 & 1 & 2 & 1 & 1 \end{pmatrix}$$

也是一个低秩矩阵。

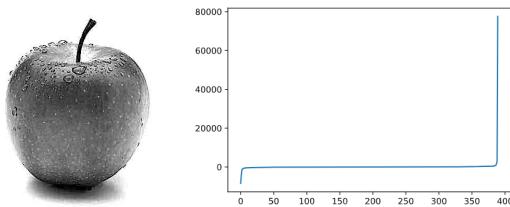


图 4.7: 图像以及图像的特征值

例 4.2.13. 左图是一个 390×390 的图像, 对应矩阵的特征值从小到大展示在右图上。可以注意到很多特征值都集中在 0 附近。记左图图像的矩阵为 A 。

前面我们提到若一个方阵 A 可对角化, 那么存在可逆矩阵 P 有

$$A = P \Sigma P^{-1}$$

其中 Σ 是对角矩阵, 且主对角线上元素是 A 从小到大的特征值。那么上面图像所对应的矩阵 A 就可以写出

$$A = P \Sigma P^{-1}$$

此时, 我们如果令那些绝对值小于 200 的特征值 (这些特征值的绝对值大小不到最大特征值绝对值大小的 0.3%) 都设为 0, 即求

$$A' = P \Sigma' P^{-1}$$

其中 $\Sigma'_{ii} = 0$, 若 $|\Sigma_{ii}| < 200$ 。此时 $\text{rank}(A') = 112$, 这说明我们实际上是可以把图像的矩阵看做是低秩矩阵。矩阵的特征值和特征向量存储着图像的信息, 尤其是特征值大的那些特征向量存储了更多的信息。

另外一方面, 把一个矩阵拆解成若干个矩阵相乘也是很有用的。关于这些内容, 我们将在下一节中详细学习。

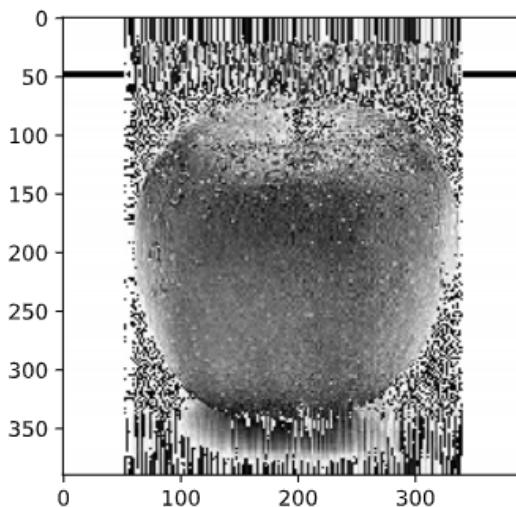


图 4.8: 新图像

低秩矩阵在很多领域都有用处。如图像恢复、图像校正、图像去噪、图像分割、图形化建模、组合系统辨识、视频监控、人脸识别、潜在语义检索、评分与协同筛选、矩阵填充、背景建模等。这些问题总体上可以分为三大问题

- 低秩矩阵恢复
- 低秩矩阵补全
- 低秩矩阵表示

低秩矩阵恢复 当低秩矩阵 \mathbf{A} 的观测或样本矩阵 $\mathbf{D} = \mathbf{A} + \mathbf{E}$ 的某些元素被严重损坏时。我们希望能够自动识别被损坏的元素，精确地恢复原低秩矩阵 \mathbf{A} 。

在工程和应用科学的许多领域（例如机器学习、控制、系统工程、信号处理、模式识别和计算机视觉）中，将一个数据矩阵分解为一个低秩矩阵与一个误差（或扰动）矩阵之和，旨在恢复低秩矩阵是远远不够的，而是需要将一个数据矩阵 \mathbf{D} 分解为一个低秩矩阵 \mathbf{A} 与一个稀疏矩阵 \mathbf{E} 之和 $\mathbf{D} = \mathbf{A} + \mathbf{E}$ ，并且希望同时恢复低秩矩阵与稀疏矩阵。矩阵的这类分解称为低秩与稀疏矩阵分解。通常这种问题，我们使用鲁棒 PCA 来求解。

鲁棒 PCA

$$\min_{\mathbf{A}, \mathbf{E}} \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_1$$

$$s.t. \mathbf{X} = \mathbf{A} + \mathbf{E}$$

其中 \mathbf{A} 代表了低秩结构信息， \mathbf{E} 是稀疏噪声。

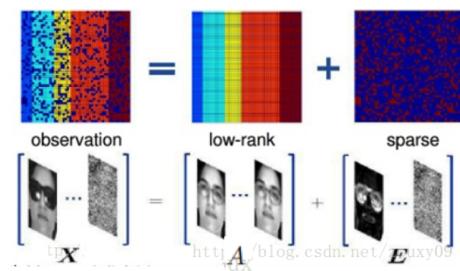


图 4.9: 低秩矩阵恢复

低秩矩阵补全 当数据矩阵 \mathbf{D} 含丢失元素时, 可根据矩阵的低秩结构来恢复的所有元素, 称此恢复过程为矩阵补全。

记 Ω 为集合 $\{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$ 的子集, 矩阵补全的原始模型可描述为如下的优化问题

$$\begin{aligned} & \min_{\mathbf{A}} \text{rank}(\mathbf{A}) \\ & \text{s.t. } P_{\Omega}(\mathbf{A}) = P_{\Omega}(\mathbf{D}) \end{aligned}$$

其中 $P_{\Omega} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$ 为一线性投影算子, 即

$$P_{\Omega}(\mathbf{D}_{ij}) = \begin{cases} D_{ij} & (i, j) \in \Omega \\ 0 & (i, j) \notin \Omega \end{cases}$$

为便于优化, 凸优化后转化为

$$\begin{aligned} & \min_{\mathbf{A}} \|\mathbf{A}\|_* \\ & \text{s.t. } P_{\Omega}(\mathbf{A}) = P_{\Omega}(\mathbf{D}) \end{aligned}$$

低秩矩阵表示 低秩矩阵表示是将数据集矩阵 \mathbf{D} 表示成字典矩阵 \mathbf{B} (也称为基矩阵) 下的线性组合, 即 $\mathbf{D} = \mathbf{BZ}$, 并希望线性组合系数矩阵 \mathbf{Z} 是低秩的。为此, 需要解下列优化问题

$$\begin{aligned} & \min_{\mathbf{Z}} \text{rank}(\mathbf{Z}) \\ & \text{s.t. } \mathbf{D} = \mathbf{BZ} \end{aligned}$$

为便于优化, 凸松弛后转化为

$$\begin{aligned} & \min_{\mathbf{Z}} \|\mathbf{Z}\|_* \\ & \text{s.t. } \mathbf{D} = \mathbf{BZ} \end{aligned}$$

若选取数据集 \mathbf{D} 本身作为字典, 则有

$$\begin{aligned} & \min_{\mathbf{Z}} \|\mathbf{Z}\|_* \\ & \text{s.t. } \mathbf{D} = \mathbf{DZ} \end{aligned}$$

那么我们可以直接写出它的解，但是需要用到 SVD 分解的知识。为了对噪声和离群点更加鲁棒，一个更合理的模型为

$$\min_{Z, E} \|Z\|_* + \lambda \|E\|_{2,1}$$

$$s.t. D = DZ + E$$

矩阵分解包括：LU 分解、QR 分解、谱分解和 Cholesky 分解、奇异值分解；将用于线性方程组求解、最小二乘问题和矩阵特征值的计算等，一起形成矩阵的四大基本计算问题。

4.3 LU 分解

4.3.1 LU 分解

LU 分解指将 $n \times n$ 的矩阵 A 分解成两个三角矩阵的乘积，形式如下：

$$A = LU = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ u_{22} & u_{23} & \cdots & u_{2n} \\ u_{33} & \cdots & u_{3n} \\ \ddots & & \vdots \\ u_{nn} \end{pmatrix}$$

其中， L 为 $n \times n$ 单位下三角矩阵（对角元素为 1）， U 是 $n \times n$ 上三角矩阵。从秩 1 分解的角度分析 $A = LU$ ，可以将 A 写成若干个秩 1 矩阵和的形式：

$$A = l_1 u_1 + l_2 u_2 + \cdots + l_r u_r = \sum_{i=1}^r l_i u_i$$

其中， r 为矩阵 A 的秩，若 A 是满秩，则 $r = n$ 。 l_i 是 L 的第 i 列， u_i 是 U 的第 i 行。 $l_i u_i$ 都是秩为 1 的矩阵，并且这个矩阵的前 $i - 1$ 行，前 $i - 1$ 列元素都是 0。

据此可以得到 u_1 是 A 的第 1 行， l_1 是 A 的第 1 列除以 u_{11} 。 u_k 是 $A - \sum_{i=1}^{k-1} l_i u_i$ 的第 k 行， l_k 是 $A - \sum_{i=1}^{k-1} l_i u_i$ 的第 k 列除以 u_{kk} 。

例 4.3.1. 求矩阵 A

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix}$$

的 LU 分解。

解. 令 u_1 是 A 的第 1 行， l_1 是 A 的第 1 列除以 u_{11} 。则

$$l_1 u_1 = \begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 8 & 2 \\ 7 & 14 & 21 \end{pmatrix}$$

$$\mathbf{A} - \mathbf{l}_1 \mathbf{u}_1 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix} - \begin{pmatrix} 1 & 2 & 3 \\ 4 & 8 & 2 \\ 7 & 14 & 21 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{pmatrix}$$

令 \mathbf{u}_2 是 $\mathbf{A} - \mathbf{l}_1 \mathbf{u}_1$ 的第 2 行, \mathbf{l}_2 是 $\mathbf{A} - \mathbf{l}_1 \mathbf{u}_1$ 的第 2 列除以 u_{22} 。则

$$\mathbf{l}_2 \mathbf{u}_2 = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \begin{pmatrix} 0 & -3 & -6 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -3 & -6 \\ 0 & -6 & -12 \end{pmatrix}$$

$$\mathbf{A} - \mathbf{l}_1 \mathbf{u}_1 - \mathbf{l}_2 \mathbf{u}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

令 \mathbf{u}_3 是 $\mathbf{A} - \mathbf{l}_1 \mathbf{u}_1 - \mathbf{l}_2 \mathbf{u}_2$ 的第 3 行, \mathbf{l}_3 是 $\mathbf{A} - \mathbf{l}_1 \mathbf{u}_1 - \mathbf{l}_2 \mathbf{u}_2$ 的第 3 列除以 u_{33} 。即

$$\mathbf{l}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \mathbf{u}_3 = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$$

所以

$$\mathbf{A} = \mathbf{LU} = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{pmatrix}$$

可以看出从 \mathbf{A} 得到 \mathbf{U} 的过程等价于对 \mathbf{A} 进行初等行变换。具体地说, \mathbf{u}_k 是通过将矩阵 \mathbf{A} 的第 k 行分别减去 \mathbf{A} 的前 $k-1$ 行的若干倍得到的。

因此, 我们可以利用初等行变化将矩阵进行 LU 分解。

步骤 1 利用初等行变换 (某一行加其它行的倍数) 化矩阵 \mathbf{A} 为阶梯型矩阵 \mathbf{U} , 即

$$\mathbf{A} = \mathbf{A}^{(0)} \xrightarrow{\mathbf{L}_1} [] \xrightarrow{\mathbf{L}_2} \cdots \xrightarrow{\mathbf{L}_{k-1}} [] \cdots \xrightarrow{\mathbf{L}_{n-1}} [] = \mathbf{U}$$

\mathbf{A} 经过 $\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_{k-1}$ 得到 $\mathbf{A}^{(k-1)}$, \mathbf{L}_k 将 $\mathbf{A}^{(k-1)}$ 的第 k 行的 $-l_{ik}$ 倍 ($i = k+1, \dots, n$), 分别加到第 i 行, 使得第 i 行的第 k 列元素都为 0。为了计算这样的 l_{ik} , 需要计算 $\frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}$ 。我们把其中 $\mathbf{A}^{(k-1)}$ 的第 k 行, 第 k 列的元素即 $a_{kk}^{(k-1)}$ 称为主元。

步骤 2 对单位矩阵执行与步骤 1 相应的初等行变换的逆变换, 得到单位下三角矩阵 \mathbf{L} , 即

$$\mathbf{I} \xrightarrow{\mathbf{L}_{n-1}^{-1}} [] \xrightarrow{\mathbf{L}_{n-2}^{-1}} \cdots \xrightarrow{\mathbf{L}_{k-1}^{-1}} [] \cdots \xrightarrow{\mathbf{L}_1^{-1}} [] = \mathbf{L}$$

输出 LU 分解由 $\mathbf{A} = \mathbf{LU}$ 给出。

例 4.3.2. 求矩阵 \mathbf{A} 的 LU 分解。

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix}$$

解. $A \rightarrow U$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix}_{A^{(0)}} \xrightarrow{\substack{R_2 - (\frac{4}{1})R_1 \\ R_3 - (\frac{7}{1})R_1}} \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{pmatrix}_{A^{(1)}} \xrightarrow{\substack{R_3 - (\frac{-6}{-3})R_2 \\ }} \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{pmatrix}_{A^{(2)}}$$

初等行变换 $\xrightarrow{\substack{R_2 - (\frac{4}{1})R_1 \\ R_3 - (\frac{7}{1})R_1}}$ 即对矩阵 $A^{(0)}$ 左乘一个初等矩阵

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -7 & 0 & 1 \end{pmatrix}$$

初等行变换 $\xrightarrow{R_3 - (\frac{-6}{-3})R_2}$ 即对 $A^{(0)}$ 左乘一个初等矩阵

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix}$$

即 $L_2 L_1 A = U$ 。所以 $A = L_1^{-1} L_2^{-1} U$, 显然

$$L_1^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 0 & 1 \end{pmatrix}, L_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}$$

可得

$$L = L_1^{-1} L_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{pmatrix}$$

观察以上的例子,

$$L_k = I - l_k e_k^T, \quad L_k^{-1} = I + l_k e_k^T$$

其中

$$l_k = (0, \dots, 0, l_{k+1,k}, \dots, l_{nk})^T, \quad l_{ik} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}, \quad i = k+1, \dots, n$$

则

$$\begin{aligned} L &= L_1^{-1} \cdots L_{n-1}^{-1} \\ &= (I + l_1 e_1^T) (I + l_2 e_2^T) \cdots (I + l_{n-1} e_{n-1}^T) \\ &= I + l_1 e_1^T + \cdots + l_{n-1} e_{n-1}^T \end{aligned}$$

草稿情切外传

即 \mathbf{L} 有如下形式

$$\mathbf{L} = \mathbf{I} + \begin{pmatrix} \mathbf{l}_1 & \mathbf{l}_2 & \cdots & \mathbf{l}_{n-1} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix}$$

我们把以上的过程归纳为算法1。

Algorithm 1 LU 分解

```

1:  $\mathbf{L} = \mathbf{I}, \mathbf{U} = \mathbf{O}$ 
2: for  $k = 1$  to  $n - 1$  do
3:   for  $i = k + 1$  to  $n$  do
4:      $l_{ik} = a_{ik}/a_{kk}$  % 更新  $L$  的第  $k$  列
5:   end for
6:   for  $j = k$  to  $n$  do
7:      $u_{kj} = a_{kj}$  % 更新  $U$  的第  $k$  行
8:   end for
9:   for  $i = k + 1$  to  $n$  do
10:    for  $j = k + 1$  to  $n$  do
11:       $a_{ij} = a_{ij} - l_{ik}u_{ik}$  % 更新矩阵  $\mathbf{A}(k + 1 : n, k + 1 : n)$ 
12:    end for
13:   end for
14: end for

```

通过以上的算法，我们会得到唯一形式的 LU 分解。我们可以证明如下定理。

定理 4.3.1. [LU 分解的唯一性] 如果 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 非奇异，并且其 LU 分解存在，则 \mathbf{A} 的 LU 分解是唯一的，且 $\det(\mathbf{A}) = u_{11}u_{22} \cdots u_{nn}$ 。

证明. 令 $\mathbf{A} = \mathbf{L}_1\mathbf{U}_1$ 和 $\mathbf{A} = \mathbf{L}_2\mathbf{U}_2$ 是非奇异矩阵 \mathbf{A} 的两个 LU 分解，则 $\mathbf{L}_1\mathbf{U}_1 = \mathbf{L}_2\mathbf{U}_2$ 。

由于 $\mathbf{L}_2^{-1}\mathbf{L}_1$ 是下三角矩阵，并且 $\mathbf{U}_2\mathbf{U}_1^{-1}$ 是上三角矩阵，所以这两个矩阵必定都等于单位矩阵，否则它们不可能相等。就是说， $\mathbf{L}_1 = \mathbf{L}_2, \mathbf{U}_1 = \mathbf{U}_2$ ，即 LU 分解是唯一的。

若 $\mathbf{A} = \mathbf{LU}$ ，则 $\det(\mathbf{A}) = \det(\mathbf{LU}) = \det(\mathbf{L})\det(\mathbf{U}) = \det(\mathbf{U}) = u_{11}u_{22} \cdots u_{nn}$ 。 □

然而，LU 分解并不一定总是存在的。我们来看一个例子。

4.3.2 选主元的 LU 分解

例 4.3.3. 求矩阵 A 的 LU 分解。

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 1 & 2 & 1 \\ 0 & 2 & 0 \end{pmatrix}$$

解. $A \rightarrow U$

$$\begin{pmatrix} 1 & 2 & 0 \\ 1 & 2 & 1 \\ 0 & 2 & 0 \end{pmatrix} \xrightarrow{R_2 - (\frac{1}{1})R_1} \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 2 & 0 \end{pmatrix}$$

由于第一次初等变换后得到矩阵 $L_1 A = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 2 & 0 \end{pmatrix}$ 的主元 $a_{22}^{(1)} = 0$, 无法进行下一步初等行变换, 也就无法继续进行 LU 分解。

我们自然地会提出疑问, 什么时候主元会为 0, 当主元为 0 时, 又该如何处理?

定理 4.3.2. 假设通过 LU 分解地过程能得到 $A^{(k-1)}$, 则主元 $a_{kk}^{(k-1)}$ 不为零的充分必要条件是 A 的 k 阶顺序主子式 $|A_k|$ 不为零。

证明. 这是显然成立的, 因为我们对矩阵 A 做初等行变换, 将矩阵的第 i 行的若干倍加到第 k 行 (其中 $k > i$), 这个变换并不改变矩阵的顺序主子式的值。也就是说 $|A_k| = \prod_{i=1}^k a_{ii}^{(i-1)}$ 。我们得到 $A^{(k-1)}$, 说明 $a_{ii}^{(i-1)} \neq 0, (i = 1, \dots, k-1)$, 因此 $a_{kk}^{(k-1)}$ 不为零, 等价于 A 的 k 阶顺序主子式 $|A_k|$ 不为零。

□

例 4.3.4. 在上例中, A 的 2 阶顺序主子式为 0, 因此 $a_{22}^{(1)} = 0$, 那么当主元为 0 时如何继续分解矩阵?

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 1 & 2 & 1 \\ 0 & 2 & 0 \end{pmatrix}$$

解. 对于出现主元为 0 的矩阵使用初等行变换中的行交换。

$A \rightarrow U$

$$\begin{pmatrix} 1 & 2 & 0 \\ 1 & 2 & 1 \\ 0 & 2 & 0 \end{pmatrix} \xrightarrow{R_2 - (\frac{1}{1})R_1} \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 2 & 0 \end{pmatrix} \xrightarrow{R_2 \leftrightarrow R_3} \begin{pmatrix} 1 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$A^{(0)}$ $A^{(1)}$ $A^{(2)}$

第一次初等变换 $\xrightarrow{R_2 - (\frac{1}{1})R_1}$ 即对矩阵 A^0 左乘 $L_1 = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, 第二次初等变换 $\xrightarrow{R_2 \leftrightarrow R_3}$

即对矩阵 $A^{(1)}$ 左乘 $P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$, 即 $PL_1A = U$ 。
所以 $A = (PL_1)^{-1}U = L_1^{-1}P^{-1}U$ 。

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \xrightarrow{R_2 \leftrightarrow R_3} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \xrightarrow{R_2 + (\frac{1}{1})R_1} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$A = (P_1L_1)^{-1}U = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

虽然 $(PL_1)^{-1}$ 不是一个下三角矩阵, 但是 $P(PL_1)^{-1}$ 是下三角矩阵。并且

$$PA = (P(PL_1)^{-1})U$$

这说明我们只需要对 A 的行重新排列, 就可以对重新排列后的矩阵进行 LU 分解。

为了避免在 LU 分解过程中主元为零, 在每次对 $A^{(i-1)}$ 做初等变换 L_i 前判断主元是否为零。若为零, 交换 $A^{(i-1)}$ 第 i 行与 $a_{ji}^{(i-1)} \neq 0$ 的第 $j(j \geq i)$ 行, 使 $a_{ji}^{(i-1)}$ 成为主元。记这个行交换的初等变换矩阵为 P_i (若不需要交换行则 $P_i = I$)。然后再做初等变换 L_i 得到 A^i , 重复上面的过程, 最终得到上三角矩阵 U 。即

$$L_nP_nL_{n-1}P_{n-1}\cdots L_2P_2L_1P_1A = U.$$

这样我们得到了 A 的分解:

$$A = (L_nP_nL_{n-1}P_{n-1}\cdots L_2P_2L_1P_1)^{-1}U.$$

虽然 $(L_nP_nL_{n-1}P_{n-1}\cdots L_2P_2L_1P_1)^{-1}$ 不是一个下三角阵, 但是如果我们将先对 A 的按如下方式先重新排列各行, 有

$$P_nP_{n-1}\cdots P_2P_1A = P_nP_{n-1}\cdots P_2P_1(L_nP_nL_{n-1}P_{n-1}\cdots L_2P_2L_1P_1)^{-1}U$$

可以证明

$$P_nP_{n-1}\cdots P_2P_1(L_nP_nL_{n-1}P_{n-1}\cdots L_2P_2L_1P_1)^{-1}$$

是一个下三角矩阵。

Algorithm 2 列主元 LU 分解

```
1:  $L = I, U = O$ 
2:  $p = [1 : n]$  % 记录行变换矩阵  $P$ 
3: for  $k = 1$  to  $n - 1$  do
4:   if  $a_{kk} = 0$  then
5:     for  $i = k + 1$  to  $n$  do
6:       if  $a_{ik} \neq 0$  then
7:         for  $j = 1$  to  $n$  do
8:            $tmp = a_{kj}, a_{kj} = a_{ij}, a_{ij} = tmp$  % 交换第  $k$  行与第  $i$  行
9:         end for
10:         $p_k = i$  % 更新行变换矩阵  $P$ 
11:      end if
12:    end for
13:  end if
14:  for  $i = k + 1$  to  $n$  do
15:     $l_{ik} = a_{ik}/a_{kk}$  % 更新  $L$  的第  $k$  列
16:  end for
17:  for  $j = k$  to  $n$  do
18:     $u_{kj} = a_{kj}$  % 更新  $U$  的第  $k$  行
19:  end for
20:  for  $i = k + 1$  to  $n$  do
21:    for  $j = k + 1$  to  $n$  do
22:       $a_{ij} = a_{ij} - a_{ik}a_{kj}$  % 更新矩阵  $A(k + 1 : n, k + 1 : n)$ 
23:    end for
24:  end for
25: end for
```

4.4 QR 分解

4.4.1 QR 分解

QR 分解指将矩阵 A 分解成列正交矩阵和上三角矩阵的乘积，形式如下：

$$A = QR = (q_1 \quad q_2 \quad q_3) \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} = q_1 r_1 + q_2 r_2 + \cdots + q_n r_n$$

其中 Q 是列正交矩阵， R 是上三角矩阵。 q_i 是 Q 的列向量， r_i 是 R 的行向量。且 $q_i r_i$ 的前 $i-1$ 列都为 0。

例 4.4.1. 考虑矩阵 A 如下：

$$A = \begin{pmatrix} 1 & 4 & 5 \\ 1 & -2 & 3 \\ 1 & 4 & 1 \\ 1 & -2 & -1 \end{pmatrix}$$

解。根据 Q 矩阵的正交性可得：

$$r_1 = q_1^T A$$

$$r_2 = q_2^T A$$

$$r_3 = q_3^T A$$

容易计算 q_1 ，它是单位化的 A 的第 1 列，即 $q_1 = \frac{a_1}{\|a_1\|} = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})^T$ ，此时可计算：

$$r_1 = q_1^T A = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \right) \begin{pmatrix} 1 & 4 & 5 \\ 1 & -2 & 3 \\ 1 & 4 & 1 \\ 1 & -2 & -1 \end{pmatrix} = (2, 2, 4)$$

$$A - q_1 r_1 = \begin{pmatrix} 1 & 4 & 5 \\ 1 & -2 & 3 \\ 1 & 4 & 1 \\ 1 & -2 & -1 \end{pmatrix} - \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} (2, 2, 4) = \begin{pmatrix} 0 & 3 & 3 \\ 0 & -3 & 1 \\ 0 & 3 & -1 \\ 0 & -3 & -3 \end{pmatrix}$$

容易计算 q_2 等于 $A - q_1 r_1$ 第二列的单位化。即 $q_2 = (\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, -\frac{1}{2})^T$ 。所以：

$$r_2 = q_2^T A = \left(\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, -\frac{1}{2} \right) \begin{pmatrix} 1 & 4 & 5 \\ 1 & -2 & 3 \\ 1 & 4 & 1 \\ 1 & -2 & -1 \end{pmatrix} = (0, 6, 2)$$

传外勿请稿草

$$A - q_1 r_1 - q_2 r_2 = \begin{pmatrix} 0 & 3 & 3 \\ 0 & -3 & 1 \\ 0 & 3 & -1 \\ 0 & -3 & -3 \end{pmatrix} - \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix} \begin{pmatrix} 0 & 6 & 2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & 2 \\ 0 & 0 & -2 \\ 0 & 0 & -2 \end{pmatrix}$$

所以 q_3 等于 $A - q_1 r_1 - q_2 r_2$ 第三列的单位化。即 $q_3 = (\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, -\frac{1}{2})^T$, 所以

$$r_3 = q_3^T A = \left(\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, -\frac{1}{2} \right) \begin{pmatrix} 1 & 4 & 5 \\ 1 & -2 & 3 \\ 1 & 4 & 1 \\ 1 & -2 & -1 \end{pmatrix} = (0, 0, 4)$$

$$A = QR = \begin{pmatrix} 1/2 & 1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 \\ 1/2 & 1/2 & -1/2 \\ 1/2 & -1/2 & -1/2 \end{pmatrix} \begin{pmatrix} 2 & 2 & 4 \\ 0 & 6 & 2 \\ 0 & 0 & 4 \end{pmatrix}$$

4.4.2 基于 Gram-Schmidt 正交化的 QR 分解

通过上面对矩阵 A 进行 QR 分解的过程，就是基于 Gram-Schmidt 正交化方法对 A 的各列进行正交化的过程。

定理 4.4.1. 若 $A \in \mathbb{R}^{m \times n}$, 且 $m \geq n$, 则存在列正交的矩阵 $Q \in \mathbb{R}^{m \times n}$ 和上三角矩阵 $R \in \mathbb{R}^{n \times n}$ 使得 $A = QR$ 。当 $m = n$ 时, Q 是正交矩阵。如果 A 是非奇异的 $n \times n$ 矩阵, 并且 R 的所有对角线元素均为正, 则在这种情况下 Q 和 R 二者是唯一的。若 A 是复矩阵, 则 Q 和 R 取复值。

证明. 先考虑 A 是非奇异的 $n \times n$ 矩阵, R 对角元都是正数的情况。

对 A 按列分块, 得到相互正交的向量 a_1, a_2, \dots, a_n , 则可以通过 Gram-Schmidt 正交化的方法求得 Q, R 。即通过对 Q 按列分块 q_1, q_2, \dots, q_m 有:

$$\begin{cases} a_1 = r_{11}q_1 \\ a_2 = r_{12}q_1 + r_{22}q_2 \\ \dots \\ a_n = r_{1n}q_1 + r_{2n}q_2 + \dots + r_{nn}q_n \end{cases}$$

由于 $\|q_1\|_2 = 1$, 令 $b_1 = r_{11}q_1 = a_1$, 所以

$$r_{11} = \|b_1\|_2, q_1 = b_1/r_{11}$$

由于 $\langle \mathbf{q}_1, \mathbf{q}_2 \rangle = 0$

$$\langle \mathbf{q}_1, \mathbf{a}_2 \rangle = r_{12} \langle \mathbf{q}_1, \mathbf{q}_1 \rangle + 0$$

$$r_{12} = \langle \mathbf{q}_1, \mathbf{a}_2 \rangle$$

令 $\mathbf{b}_2 = r_{22} \mathbf{q}_2 = \mathbf{a}_2 - \langle \mathbf{q}_1, \mathbf{a}_2 \rangle \mathbf{q}_1$, 有

$$r_{22} = \|\mathbf{b}_2\|_2, \mathbf{q}_2 = \mathbf{b}_2 / r_{22}$$

类似的, 对于

$$\mathbf{a}_i = r_{1i} \mathbf{q}_1 + r_{2i} \mathbf{q}_2 + \cdots + r_{ii} \mathbf{q}_i$$

有

$$r_{ji} = \langle \mathbf{q}_j, \mathbf{a}_i \rangle, \forall j < i$$

令 $\mathbf{b}_i = r_{ii} \mathbf{q}_i = \mathbf{a}_i - (r_{1i} \mathbf{q}_1 + r_{2i} \mathbf{q}_2 + \cdots + r_{i-1,i} \mathbf{q}_{i-1})$

$$r_{ii} = \|\mathbf{b}_i\|, \mathbf{q}_i = \mathbf{b}_i / r_{ii}$$

综上

$$\mathbf{A} = \mathbf{Q} \begin{pmatrix} \|\mathbf{b}_1\| & \langle \mathbf{q}_1, \mathbf{a}_2 \rangle & \langle \mathbf{q}_1, \mathbf{a}_3 \rangle & \cdots & \langle \mathbf{q}_1, \mathbf{a}_n \rangle \\ 0 & \|\mathbf{b}_2\| & \langle \mathbf{q}_2, \mathbf{a}_3 \rangle & \cdots & \langle \mathbf{q}_2, \mathbf{a}_n \rangle \\ 0 & 0 & \|\mathbf{b}_3\| & \cdots & \langle \mathbf{q}_3, \mathbf{a}_n \rangle \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \|\mathbf{b}_n\| \end{pmatrix}$$

即为所求的分解。

当 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 时, 对 \mathbf{R} 取前 n 列即可, 此时 \mathbf{Q} 不唯一。得到分解形式:

$$\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R}_{n \times n} \\ \mathbf{O}_{m-n} \end{pmatrix}$$

也可以写成

$$\mathbf{A} = \mathbf{Q}_{m \times n} \mathbf{R}_{n \times n}$$

$\mathbf{Q}_{m \times n}$ 为 \mathbf{Q} 的前 n 列。

□

注意到 $\mathbf{A}^T \mathbf{A} = (\mathbf{Q} \mathbf{R})^T (\mathbf{Q} \mathbf{R}) = \mathbf{R}^T \mathbf{R}$, 因此可以得出结论: $\mathbf{G} = \mathbf{R}^T$ 是 $\mathbf{A}^T \mathbf{A}$ 的下三角 Cholesky 因子。由于这个原因, 在关于估计的文献中, 矩阵 \mathbf{R} 常称为平方根滤波器(算子)。

下面的引理称为矩阵分解引理, 它在矩阵 QR 分解的应用中是一个有用的结果。

推论 4.4.1. 若 A 和 B 是两个任意 $m \times n$ 实矩阵，则

$$A^T A = B^T B \quad (4.2)$$

当且仅当存在一个 $m \times m$ 正交矩阵 Q ，使得

$$QA = B \quad (4.3)$$

矩阵 A 的 QR 分解可以通过 Gram-Schmidt 正交化、Household 变换、Givens 变换等方法实现。

在前一节，我们利用 Gram-Schmidt 正交化方法给出了构造出 QR 分解存在的证明，利用 Gram-Schmidt 正交化方法可以直接求得矩阵的 QR 分解。

例 4.4.2. 求下列矩阵的正交三角分解 (QR) 表达式：

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

解. 记 $\mathbf{a}_1 = (0, 1, 1)^T$, $\mathbf{a}_2 = (1, 1, 0)^T$, $\mathbf{a}_3 = (1, 0, 1)^T$, 由 Gram-Schmidt 正交化方法可得

$$\mathbf{b}_1 = \mathbf{a}_1 = (0, 1, 1)^T, |\mathbf{b}_1| = \sqrt{2}, \mathbf{q}_1 = \frac{1}{\sqrt{2}}(0, 1, 1)^T;$$

$$\langle \mathbf{q}_1, \mathbf{a}_2 \rangle = \frac{1}{\sqrt{2}}, \mathbf{b}_2 = \mathbf{a}_2 - \langle \mathbf{a}_2, \mathbf{q}_1 \rangle \mathbf{q}_1 = (1, \frac{1}{2}, -\frac{1}{2})^T, |\mathbf{b}_2| = \frac{\sqrt{6}}{2}, \mathbf{q}_2 = \frac{1}{\sqrt{6}}(2, 1, -1)^T;$$

$$\langle \mathbf{q}_1, \mathbf{a}_3 \rangle = \frac{1}{\sqrt{2}}, \langle \mathbf{q}_2, \mathbf{a}_3 \rangle = \frac{1}{6}, \mathbf{b}_3 = \mathbf{a}_3 - \langle \mathbf{a}_3, \mathbf{q}_1 \rangle \mathbf{q}_1 - \langle \mathbf{a}_3, \mathbf{q}_2 \rangle \mathbf{q}_2 = (\frac{2}{3}, -\frac{2}{3}, \frac{2}{3})^T,$$

$$|\mathbf{b}_3| = \frac{2}{\sqrt{3}}, \mathbf{q}_3 = \frac{1}{\sqrt{3}}(1, -1, 1)^T$$

于是取

$$Q = (\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3) = \begin{bmatrix} 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{bmatrix}$$

$$R = \begin{bmatrix} |\mathbf{b}_1| & \langle \mathbf{a}_2, \mathbf{q}_1 \rangle & \langle \mathbf{a}_3, \mathbf{q}_1 \rangle \\ 0 & |\mathbf{b}_2| & \langle \mathbf{a}_3, \mathbf{q}_2 \rangle \\ 0 & 0 & |\mathbf{b}_3| \end{bmatrix} = \begin{bmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{\sqrt{6}}{2} & \frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{2}{\sqrt{3}} \end{bmatrix}$$

那么 $A = QR$ 即为所求表达式。

在小节4.4.1和4.4.2中，我们知道，可以通过如算法3的标准的 Gram-Schmidt 正交化方法得到矩阵 A 的 QR 分解。

经典的 Gram-Schmidt 正交化方法将计算出的 Q 的每一列存储在 A 的对应列可以节省存储空间，其过程可以用图 4.10 表示

Algorithm 3 经典 Gram-Schmidt 正交化方法

Require: $A \in \mathbb{R}^{m \times n}$

- 1: $R = O_{n \times n}$
 - 2: $b_1 = a_1$
 - 3: $r_{11} = \|b_1\|$
 - 4: $q_1 = b_1 / r_{11}$
 - 5: **for** $i = 2 : n$ **do**
 - 6: $r_{ki} = \langle q_k, a_i \rangle, 1 \leq k \leq i - 1$
 - 7: $b_i = a_i - \sum_{k=1}^{i-1} r_{ki} q_k$
 - 8: $r_{ii} = \|b_i\|$
 - 9: $q_i = b_i / r_{ii}$
 - 10: **end for**
-

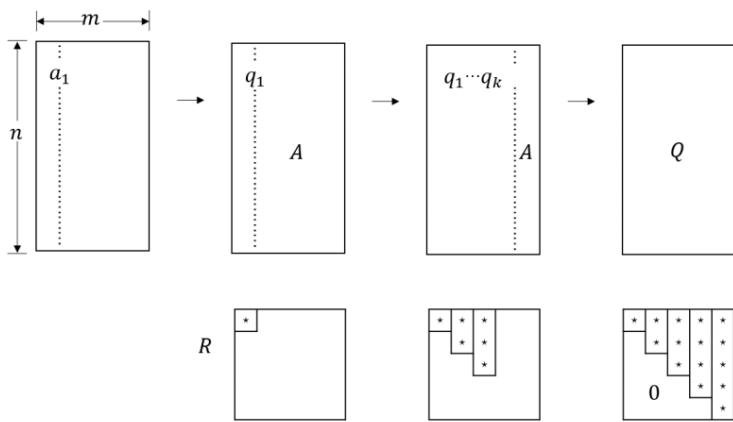


图 4.10: QR 分解的经典 Gram-Schmidt 正交化算法

Björck 发现，对经典 Gram-Schmidt 正交化法加以修正，使上三角形矩阵 R 的元素不是按列，而是按行计算时，舍入误差将变小。这样一种修正方法叫做修正经典 Gram-Schmidt 正交化算法。具体来说， a_1 的标准正交化结果取作 q_1 （这与经典 Gram-Schmidt 正交化法相同），但同时

需要从 $\mathbf{a}_2, \mathbf{a}_3, \dots, \mathbf{a}_n$ 预先减去与 \mathbf{a}_1 平行的分量，即

$$\left. \begin{array}{l} r_{11} = \|\mathbf{a}_1\|, \mathbf{q}_1 = \mathbf{a}_1 / r_{11} \\ r_{1j} = \langle \mathbf{q}_1, \mathbf{a}_j \rangle, \mathbf{a}_j^{(1)} = \mathbf{a}_j - \mathbf{q}_1 r_{1j}, \quad 2 \leq j \leq n \end{array} \right\} \quad (4.4)$$

经过以上运算后，向量 $\mathbf{a}_2^{(1)}, \mathbf{a}_3^{(1)}, \dots, \mathbf{a}_n^{(1)}$ 与 \mathbf{q}_1 正交。然后，将 $\mathbf{a}_2^{(1)}$ 标准正交化，并从 $\mathbf{a}_3^{(1)}, \mathbf{a}_4^{(1)}, \dots, \mathbf{a}_n^{(1)}$ 减去与 $\mathbf{a}_2^{(1)}$ 平行的分量，得

$$\left. \begin{array}{l} r_{22} = \|\mathbf{a}_2^{(1)}\|, \quad \mathbf{q}_2 = \mathbf{a}_2^{(1)} / r_{22} \\ r_{2j} = \mathbf{q}_2^T \mathbf{a}_j^{(1)}, \quad \mathbf{a}_j^{(2)} = \mathbf{a}_j^{(1)} - \mathbf{q}_2 r_{2j}, \quad 3 \leq j \leq n \end{array} \right\} \quad (4.5)$$

这样构造的向量 $\mathbf{a}_3^{(2)}, \mathbf{a}_4^{(2)}, \dots, \mathbf{a}_n^{(2)}$ 与 $\mathbf{q}_1, \mathbf{q}_2$ 均正交。重复这一过程，就可以将 A 和 Q 重写在同一矩阵。因此我们可以通过如算法 3 的修正 Gram-Schmidt 正交化方法得到矩阵 A 的 QR 分解。图 4.11 示出了修正 Gram-Schmidt 正交化算法的运算过程。显然，修正 Gram-Schmidt 正交化算法与经典 Gram-Schmidt 正交化算法的理论结果是一样的。但是，由于计算机在计算过程的每一步可能产生误差，修正方法能够减小误差。

Algorithm 4 修正 Gram-Schmidt 正交化方法

Require: $A \in \mathbb{R}^{m \times n}$

```

1: for  $i = 1, \dots, n$  do
2:    $\mathbf{b}^{(i)} = \mathbf{a}^{(i)}$ 
3:   end for
4:   for  $i = 1, \dots, n$  do
5:      $r_{ii} = \|\mathbf{b}^{(i)}\|$ 
6:      $\mathbf{q}^{(i)} = \mathbf{b}^{(i)} / r_{ii}$ 
7:     for  $j = i + 1, \dots, n$  do
8:        $r_{ij} = \mathbf{q}^{(i)T} \mathbf{b}^{(j)}, \mathbf{b}^{(j)} = \mathbf{b}^{(j)} - r_{ii} \mathbf{q}^{(i)}$ 
9:     end for
10:   end for

```

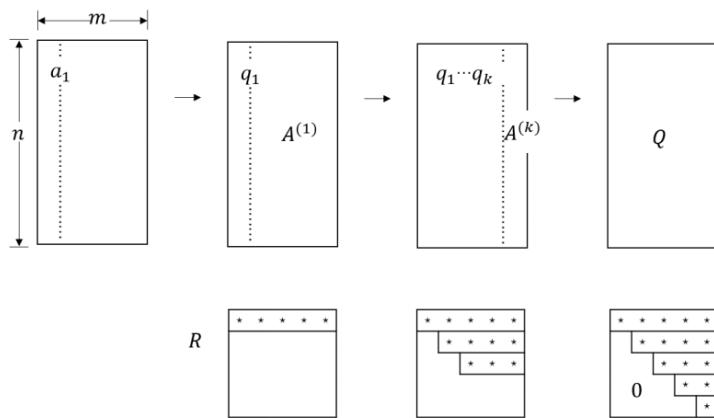


图 4.11: QR 分解的修正 Gram-Schmidt 正交化算法

为了分析误差的情况，假定 \mathbf{q}_2 的计算值 $\hat{\mathbf{q}}_2$ 包含有少量与 \mathbf{q}_1 平行的分量，即 $\hat{\mathbf{q}}_2 = \mathbf{q}_2 + \varepsilon \mathbf{q}_1$ 。

应用经典 Gram-Schmidt 法，求 r_{23} 的计算值 \hat{r}_{23} 时，具体计算如下：

$$\hat{r}_{23} = (\mathbf{q}_2^T + \varepsilon \mathbf{q}_1^T) \mathbf{a}_3 = \mathbf{q}_2^T \mathbf{a}_3 + \varepsilon \mathbf{q}_1^T \mathbf{a}_3 \quad (4.6)$$

而在修正 Gram-Schmidt 法中，求 r_{23} 时计算的是 $\hat{\mathbf{q}}_2$ 和 $\mathbf{a}_2^{(1)} = \mathbf{a}_3 - r_{13} \mathbf{q}_1$ 的内积，即

$$(\mathbf{q}_2^T + \varepsilon \mathbf{q}_1^T)(\mathbf{a}_3 - r_{13} \mathbf{q}_1) = \mathbf{q}_2^T \mathbf{a}_3 + \varepsilon \mathbf{q}_1^T \mathbf{a}_3 - \varepsilon r_{13} = \mathbf{q}_2^T \mathbf{a}_3 \quad (4.7)$$

比较(4.6)与(4.7)，可以看出，经典方法比修正方法多第二项误差。尤其是当 $|r_{23}| \ll |r_{13}|$ 时。修正 Gram-Schmidt 法减小误差的效果将更加明显。

4.4.3 基于 Householder 变换的 QR 分解

假设对任意一个向量 \mathbf{a} 我们都能找到一个和 \mathbf{a} 有关的正交矩阵 \mathbf{H} 使得 $\mathbf{H}\mathbf{a} = \alpha \mathbf{e}_1$ ，其中 \mathbf{e}_1 是只有第 1 个分量为 1，其余分量都为 0 的向量。关于如何由 \mathbf{a} 确定 \mathbf{H} 的方法我们稍后再讲。现在，按照我们的假设，可以通过如下方式得到 $A \in \mathbb{R}^{m \times n}$ 的 QR 分解：

$$A_{m \times n} = \begin{pmatrix} \otimes & \times & \times & \cdots & \times \\ \otimes & \times & \times & \cdots & \times \\ \otimes & \times & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \otimes & \times & \times & \cdots & \times \\ \otimes & \times & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \otimes & \times & \times & \cdots & \times \end{pmatrix} \quad (4.8)$$

传
习
草
稿
请

我们将 \mathbf{A} 的第一列 \mathbf{a}^0 用“ \otimes ”标注，其余元素用“ \times ”代表，按照假设，由这些元素构成的向量可以确定 m 阶正交矩阵 \mathbf{H}_1 ，使得：

$$\mathbf{H}_1 \begin{pmatrix} \otimes \\ \otimes \\ \otimes \\ \vdots \\ \otimes \\ \otimes \\ \otimes \\ \vdots \\ \otimes \end{pmatrix} = \begin{pmatrix} \times \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbf{A}_1 = \mathbf{H}_1 \mathbf{A}_{m \times n} = \begin{pmatrix} \times & \times & \times & \cdots & \times \\ 0 & \otimes & \times & \cdots & \times \\ 0 & \otimes & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \otimes & \times & \cdots & \times \\ 0 & \otimes & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \otimes & \times & \cdots & \times \end{pmatrix}$$

对新得到矩阵的第二列用“ \otimes ”标注的元素构成的向量 $\mathbf{a}^{(1)}$ 可以确定 $m - 1$ 阶正交矩阵 $\hat{\mathbf{H}}_2$ ，使得：

$$\hat{\mathbf{H}}_2 \begin{pmatrix} \otimes \\ \otimes \\ \vdots \\ \otimes \\ \otimes \\ \vdots \\ \otimes \end{pmatrix} = \begin{pmatrix} \times \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

令 $\mathbf{H}_2 = \text{diag}\{I_1, \hat{\mathbf{H}}_2\}$ ，这样保证 \mathbf{H}_2 是一个正交矩阵，并且以之左乘 \mathbf{A}_1 不会改变 \mathbf{A}_1 的第一行，容易得出：

$$\mathbf{A}_2 = \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} = \begin{pmatrix} \times & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ 0 & 0 & \otimes & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \otimes & \cdots & \times \\ 0 & 0 & \otimes & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \otimes & \cdots & \times \end{pmatrix}$$

以此方法，假设得到矩阵 $\mathbf{A}_k = \mathbf{H}_k \mathbf{H}_{k-1} \cdots \mathbf{H}_1 \mathbf{A}$ ，并且 \mathbf{A}_k 前 k 个对角元下方的元素全为 0。则可利用 \mathbf{A}_k 的第 $k + 1$ 个对角元及其下方的元素组成的向量 $\mathbf{a}^{(k)}$ 构造 $\hat{\mathbf{H}}_{k+1}$ 以及 $\mathbf{H}_{k+1} =$

暂稿请勿外传

$\text{diag}\{\mathbf{I}_k, \hat{\mathbf{H}}_{k+1}\}$ 使得 $\mathbf{A}_{k+1} = \mathbf{H}_{k+1}\mathbf{A}_k$ 满足 \mathbf{A}_{k+1} 的前 $k+1$ 个对角元下方的元素全为 0。不断进行下去，最终可以得到

$$\mathbf{A}_n = \mathbf{H}_n \mathbf{H}_{n-1} \cdots \mathbf{H}_1 \mathbf{A} = \begin{pmatrix} \times & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ 0 & 0 & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \times \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{R} \\ \mathbf{O} \end{pmatrix}$$

令 $\mathbf{Q} = (\mathbf{H}_n \mathbf{H}_{n-1} \cdots \mathbf{H}_1)^T$ 可得 \mathbf{A} 的 QR 分解：

$$\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R} \\ \mathbf{O} \end{pmatrix}$$

现在的问题是，到底能不能找到和 \mathbf{a} 有关的正交矩阵 \mathbf{H} 使得 $\mathbf{H}\mathbf{a} = \alpha\mathbf{e}_1$ 呢？答案自然是肯定的。Householder 变换就是我们所需要的矩阵。

Householder 变换能如 Gauss 变换一样，可以通过适当选取单位向量 \mathbf{w} ，把一个给定向量的若干个指定的分量变为零。

定理 4.4.2. $\mathbf{0} \neq \mathbf{x} \in \mathbb{R}^n$ ，则可构造单位向量 $\mathbf{w} \in \mathbb{R}^n$ ，使由(3.9)定义的 Householder 变换 \mathbf{H} 满足

$$\mathbf{H}\mathbf{x} = \alpha\mathbf{e}_1 \quad (4.9)$$

其中 $\alpha = \pm \|\mathbf{x}\|_2$ 。

证明。由于

$$\mathbf{H}\mathbf{x} = (\mathbf{I} - 2\mathbf{w}\mathbf{w}^T)\mathbf{x} = \mathbf{x} - 2(\mathbf{w}^T\mathbf{x})\mathbf{w}$$

$$2(\mathbf{w}^T\mathbf{x})\mathbf{w} = \mathbf{x} - \mathbf{H}\mathbf{x}$$

因此 \mathbf{w} 为与 $\mathbf{x} - \mathbf{H}\mathbf{x}$ 同方向的单位向量，故欲使 $\mathbf{H}\mathbf{x} = \alpha\mathbf{e}_1$ ，则 \mathbf{w} 应为

$$\mathbf{w} = \frac{\mathbf{x} - \alpha\mathbf{e}_1}{\|\mathbf{x} - \alpha\mathbf{e}_1\|_2}$$

又因 \mathbf{H} 是正交矩阵，必须有

$$\|\mathbf{x}\|_2 = \|\mathbf{H}\mathbf{x}\|_2 = \|\alpha\mathbf{e}_1\|_2 = |\alpha| \cdot \|\mathbf{e}_1\|_2 = |\alpha|$$

即 $\alpha = \pm \|\mathbf{x}\|_2$ 。容易验证，如上选取的 \mathbf{H} 确实满足式(4.9)。验证过程如下：

$$\begin{aligned}\mathbf{Hx} &= \mathbf{x} - 2 \frac{(\mathbf{x} - \alpha \mathbf{e}_1)(\mathbf{x} - \alpha \mathbf{e}_1)^T}{\|\mathbf{x} - \alpha \mathbf{e}_1\|_2^2} \mathbf{x} \\ &= \mathbf{x} - \frac{2(\mathbf{x} - \alpha \mathbf{e}_1)^T \mathbf{x}}{\|\mathbf{x} - \alpha \mathbf{e}_1\|_2^2} (\mathbf{x} - \alpha \mathbf{e}_1) \\ &= \mathbf{x} - \frac{2\|\mathbf{x}\|_2^2 - 2\alpha \mathbf{e}_1^T \mathbf{x}}{\|\mathbf{x}\|_2^2 - 2\alpha \mathbf{e}_1^T \mathbf{x} + \alpha^2} (\mathbf{x} - \alpha \mathbf{e}_1) \\ &= \mathbf{x} - \frac{2\alpha^2 - 2\alpha \mathbf{e}_1^T \mathbf{x}}{\alpha^2 - 2\alpha \mathbf{e}_1^T \mathbf{x} + \alpha^2} (\mathbf{x} - \alpha \mathbf{e}_1) \\ &= \mathbf{x} - (\mathbf{x} - \alpha \mathbf{e}_1) = \alpha \mathbf{e}_1\end{aligned}$$

□

定理4.4.2告诉我们，对任意的 $\mathbf{x} \in \mathbb{R}^n (\mathbf{x} \neq \mathbf{0})$ 都可构造出 Householder 矩阵 \mathbf{H} ，使 \mathbf{Hx} 的后 $n-1$ 分量为零。而且其证明亦告诉我们，可按如下的步骤来构造确定 \mathbf{H} 的单位向量 \mathbf{w} ：

- (1) 计算 $\mathbf{v} = \mathbf{x} \pm \|\mathbf{x}\|_2 \mathbf{e}_1$ ；
- (2) 计算 $\mathbf{w} = \mathbf{v} / \|\mathbf{v}\|_2$ 。

首先，一个自然的问题是，实际计算时， $\|\mathbf{x}\|_2$ 前的符号如何选取？最好为了使变换后得到的 α 为正数，则应取

$$\mathbf{v} = \mathbf{x} - \|\mathbf{x}\|_2 \mathbf{e}_1$$

但是这样选取就会出现一个问题，如果 \mathbf{x} 是一个很接近于 \mathbf{e}_1 的向量，计算

$$v_1 = x_1 - \|\mathbf{x}\|_2$$

时，就会出现两个相近的数相减，而导致严重地损失有效数字，这里 v_1 和 x_1 分别是向量 \mathbf{v} 和 \mathbf{x} 的第一个分量。不过幸运的是，只要对上式做一简单的等价变形，就可避免这一问题的出现。事实上，注意到

$$v_1 = x_1 - \|\mathbf{x}\|_2 = \frac{x_1^2 - \|\mathbf{x}\|_2^2}{x_1 + \|\mathbf{x}\|_2} = \frac{-(x_2^2 + \cdots + x_n^2)}{x_1 + \|\mathbf{x}\|_2}$$

只要在 $x_1 > 0$ 时使用上面式子来计算 v_1 ，就会避免出现两个相近的数相减的情形。

其次，注意到

$$\mathbf{H} = \mathbf{I} - 2\mathbf{w}\mathbf{w}^T = \mathbf{I} - \frac{2}{\mathbf{v}^T \mathbf{v}} \mathbf{v}\mathbf{v}^T = \mathbf{I} - \beta \mathbf{v}\mathbf{v}^T$$

其中 $\beta = 2/\mathbf{v}\mathbf{v}^T$ ，我们就没有必要非求出 \mathbf{w} 不可，而只需求出 β 和 \mathbf{v} 即可。然而在实际计算时，将 \mathbf{v} 规格化为第一个分量为 1 的向量是方便的，这是因为这样正好可以把 \mathbf{v} 的后 $n-1$ 个分量保存在 \mathbf{x} 的后 $n-1$ 个分量位置上，而 \mathbf{v} 的第一个分量 1 就无需保存了。

有了 Householder 变换，我们就可以完整的计算矩阵的 QR 分解了。

例 4.4.3. 已知矩阵 $\mathbf{A} = \begin{pmatrix} 0 & 3 & 1 \\ 0 & 4 & -2 \\ 2 & 1 & 1 \end{pmatrix}$ ，利用 Householder 变换求 \mathbf{A} 的 QR 分解。

解. 因为 $\alpha_1 = (0, 0, 2)^T$, 记 $a_1 = \|\alpha_1\|_2 = 2$, 令 $w_1 = \frac{\alpha_1 - a_1 e_1}{\|\alpha_1 - a_1 e_1\|_2} = \frac{1}{\sqrt{2}}(-1, 0, 1)^T$, 则

$$H_1 = I - 2w_1 w_1^H = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

从而

$$H_1 A = \begin{pmatrix} 2 & 1 & 2 \\ 0 & 4 & -2 \\ 0 & 3 & 1 \end{pmatrix}$$

记 $\beta = (4, 3)^T$, 则 $b_2 = \|\beta\|_2 = 5$, 令 $w_2 = \frac{\beta_2 - b_2 e_2}{\|\beta_2 - b_2 e_2\|_2} = \frac{1}{\sqrt{10}}(-1, 3)^T$

$$\tilde{H}_2 = I - 2w_2 w_2^H = \begin{pmatrix} \frac{4}{5} & \frac{3}{5} \\ \frac{3}{5} & -\frac{4}{5} \end{pmatrix}$$

记

$$H_2 = \begin{pmatrix} 1 & \mathbf{0}^T \\ 0 & \tilde{H}_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & \frac{3}{5} \\ 0 & \frac{3}{5} & -\frac{4}{5} \end{pmatrix}$$

, 则

$$H_2(H_1 A) = \begin{pmatrix} 2 & 1 & 2 \\ 0 & 5 & -1 \\ 0 & 0 & -2 \end{pmatrix} = R$$

取

$$Q = H_1 H_2 = \begin{pmatrix} 0 & \frac{3}{5} & -\frac{4}{5} \\ 0 & \frac{4}{5} & \frac{3}{5} \\ 1 & 0 & 0 \end{pmatrix}$$

则 $A = QR$ 。

4.4.4 基于 Givens 旋转的 QR 分解

Givens 变换

欲把一个向量中许多分量化为零, 可以用 Householder 变换, 例如前面所讲到的把一个向量中若干相邻分量化为零。如果只将其中一个分量化为零, 则应采用 Givens 变换。Givens 变换只改变向量的 i, k 个分量, 变换对应的矩阵如下:

定义 4.4.1.

$$\mathbf{G}(i, k, \theta) = \mathbf{I} + s(\mathbf{e}_i \mathbf{e}_k^T - \mathbf{e}_k \mathbf{e}_i^T) + (c - 1)(\mathbf{e}_i \mathbf{e}_i^T + \mathbf{e}_k \mathbf{e}_k^T)$$

$$= \begin{bmatrix} 1 & & \vdots & & \vdots & & \\ & \ddots & & \vdots & & \vdots & \\ \cdots & \cdots & c & \cdots & s & \cdots & \cdots & i \\ & & \vdots & & \vdots & & \\ \cdots & \cdots & -s & \cdots & c & \cdots & \cdots & k \\ & & \vdots & & \vdots & & \ddots \\ & & \vdots & & \vdots & & \\ & & & & & & 1 \end{bmatrix}$$

称为 *Givens* 变换矩阵, 即 *Givens* 旋转矩阵, 其中 $c = \cos \theta, s = \sin \theta$ 。

那么怎样选取 θ 或者 c 和 s , 以达到我们将向量的第 k 个分量变为 0 的目标呢?

设 $\mathbf{x} \in \mathbb{R}^n$, 令 $\mathbf{y} = \mathbf{G}(i, k, \theta)\mathbf{x}$, 则有

$$\begin{pmatrix} y_i \\ y_k \end{pmatrix} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} x_i \\ x_k \end{pmatrix} = \begin{pmatrix} x_k & x_i \\ -x_i & x_k \end{pmatrix} \begin{pmatrix} s \\ c \end{pmatrix}$$

当 $j \neq i, k$ 时, $y_j = x_j$

因此, 如要 $y_k = 0$, 利用正交变换不改变向量的二范数, 可知:

$$y_i = \sqrt{x_i^2 + x_k^2}, \quad y_k = 0$$

因此:

$$c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}}, \quad s = \frac{x_k}{\sqrt{x_i^2 + x_k^2}} \tag{4.10}$$

从几何上来看, $\mathbf{G}(i, k, \theta)\mathbf{x}$ 是在 (i, k) 坐标平面内将 \mathbf{x} 顺时针方向做了 θ 度的旋转, 所以 *Givens* 变换亦称为平面旋转变换。通过旋转使得 $x_k = 0$, 则此时旋转角度 $\theta = \arctan(x_k/x_i)$ 。

若利用(4.10)式计算 c 和 s , 在计算 $x_i^2 + x_k^2$ 时, 可能会发生溢出。为了避免这种情形发生, 对给定的实数 a 和 b , 实际上是按下列的方法计算 $c = \cos \theta$ 和 $s = \sin \theta$, 使得

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}$$

Algorithm 5 Givens 变换

```

1: if  $b = 0$  then
2:    $c = 1; s = 0$ 
3: else
4:   if  $|b| > |a|$  then
5:      $\tau = a/b; s = 1/\sqrt{1 + \tau^2}; c = s\tau$ 
6:   else
7:      $\tau = b/a; c = 1/\sqrt{1 + \tau^2}; s = c\tau$ 
8:   end if
9: end if

```

定理 4.4.3. 对于任意向量 $x \in \mathbb{R}^n$, 存在 Givens 变换 T_{kl} 使得 $T_{kl}x$ 的第 l 个分量为 0, 第 k 个分量为非负实数, 其余分量不变。

证明. 记 $x = (x_1, x_2, \dots, x_n)^T, T_{kl}x = (y_1, y_2, \dots, y_n)^T$

由 Givens 矩阵的定义可得

$$\begin{cases} y_k &= cx_k + sx_l \\ y_l &= -sx_k + cx_l \\ y_j &= x_j, (j \neq k, l) \end{cases}$$

(i) 当 $|x_k|^2 + |x_l|^2 = 0$ 时, 取 $c = 1, s = 0$, 则 $T_{kl} = I$, 此时

$$y_k = y_l = 0, y_j = x_j (j \neq k, l)$$

结论成立

(ii) 当 $|x_k|^2 + |x_l|^2 \neq 0$ 时, 取

$$c = \frac{x_k}{\sqrt{|x_k|^2 + |x_l|^2}}, s = \frac{x_l}{\sqrt{|x_k|^2 + |x_l|^2}},$$

$$\begin{cases} y_k &= \frac{x_k^2}{\sqrt{|x_k|^2 + |x_l|^2}} + \frac{x_l^2}{\sqrt{|x_k|^2 + |x_l|^2}} = \sqrt{|x_k|^2 + |x_l|^2} > 0 \\ y_l &= -\frac{x_k x_l}{\sqrt{|x_k|^2 + |x_l|^2}} + \frac{x_l x_k}{\sqrt{|x_k|^2 + |x_l|^2}} = 0 \\ y_j &= x_j, (j \neq k, l) \end{cases}$$

结论成立

□

推论 4.4.2. 给定一个向量 $x \in \mathbb{R}^n$, 则存在一组 Givens 矩阵 $T_{12}, T_{13}, \dots, T_{1n}$, 使得

$$T_{1n} \dots T_{13} T_{12} x = \|x\|_2 e_1,$$

称为用 Givens 变换化向量 $x \in \mathbb{R}^n$ 与第一自然基向量 e_1 共线。

证明. 设 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, 由定理 3 知存在 Givens 矩阵 \mathbf{T}_{12} , 使得

$$\mathbf{T}_{12}\mathbf{x} = (\sqrt{|x_1|^2 + |x_2|^2}, 0, x_3, \dots, x_n)^T$$

对于 $\mathbf{T}_{12}\mathbf{x}$ 又存在 Givens 矩阵 \mathbf{T}_{13} 使得

$$\mathbf{T}_{13}(\mathbf{T}_{12}\mathbf{x}) = (\sqrt{|x_1|^2 + |x_2|^2 + |x_3|^2}, 0, 0, x_4, \dots, x_n)^T$$

依此继续下去, 可以得出

$$\mathbf{T}_{1n} \dots \mathbf{T}_{13}\mathbf{T}_{12}\mathbf{x} = (\sqrt{|x_1|^2 + |x_2|^2 + \dots + |x_n|^2}, 0, 0, \dots, 0)^T = \|\mathbf{x}\|_2 \mathbf{e}_1$$

□

例 4.4.4. 用 Givens 变换化向量 $x = (1, 2, 2)$ 与第一自然基向量共线

解. 由于 $x_1 = 1, x_2 = 2, \sqrt{|x_1|^2 + |x_2|^2} = \sqrt{5}$ 取

$$c_1 = \frac{1}{\sqrt{5}}, s_1 = \frac{2}{\sqrt{5}}$$

构造 Givens 矩阵

$$\mathbf{T}_{12} = \begin{pmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 \\ -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{T}_{12}\mathbf{x} = \begin{pmatrix} \sqrt{5} \\ 0 \\ 2 \end{pmatrix}$$

对于 $\mathbf{T}_{12}\mathbf{x}$ 取

$$c_2 = \frac{\sqrt{5}}{3}, s_2 = \frac{2}{3}$$

$$\mathbf{T}_{13} = \begin{pmatrix} \frac{\sqrt{5}}{3} & 0 & \frac{2}{3} \\ 0 & 1 & 0 \\ -\frac{2}{3} & 0 & \frac{\sqrt{5}}{3} \end{pmatrix}, \mathbf{T}_{13}\mathbf{T}_{12}\mathbf{x} = 3\mathbf{e}_1$$

采用 Givens 旋转的 QR 分解

Givens 旋转也可以用来计算 QR 分解。显然, 如果用一个 Givens 变换左(或右)乘一个矩阵 $A \in \mathbb{R}^{n \times q}$, 则它改变 A 的第 i, k 行(列)的元素, 其余元素保持不变。对于 m 行 n 列的矩阵, 可以先对 $m-1, m$ 行做 Givens 变换使得第 1 列的第 m 行元素为 0, 再对 $m-2$ 和 $m-1$ 行做 Givens 变换, 使得第 1 列的第 $m-1$ 行元素为 0, 依次做下去, 就可以使得第 1 列除第 1 行外

的元素都变为 0。然后我们对第 2 列到第 n 列如法炮制，对第 i 列只使得其第 i 行以下的元素全部变为 0。这里以 4×3 矩阵为例，说明 Givens QR 分解的思想：

$$\begin{array}{c} \left[\begin{array}{ccc} \times & \times & \times \\ \times & \times & \times \\ \otimes & \times & \times \\ \otimes & \times & \times \end{array} \right] \xrightarrow{(3,4)} \left[\begin{array}{ccc} \times & \times & \times \\ \otimes & \times & \times \\ \otimes & \times & \times \\ 0 & \times & \times \end{array} \right] \xrightarrow{(2,3)} \left[\begin{array}{ccc} \otimes & \times & \times \\ \otimes & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{array} \right] \xrightarrow{(1,2)} \left[\begin{array}{ccc} \times & \times & \times \\ 0 & \times & \times \\ 0 & \otimes & \times \\ 0 & \otimes & \times \end{array} \right] \\ \xrightarrow{(3,4)} \left[\begin{array}{ccc} \times & \times & \times \\ 0 & \otimes & \times \\ 0 & \otimes & \times \\ 0 & 0 & \times \end{array} \right] \xrightarrow{(2,3)} \left[\begin{array}{ccc} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \otimes \\ 0 & 0 & \otimes \end{array} \right] \\ \xrightarrow{(3,4)} \left[\begin{array}{ccc} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \end{array} \right] \end{array}$$

其中， \otimes 表示用 Givens 旋转进行变换的元素。

从上述说明中易得出结论：如果令 \mathbf{G}_j 代表约化过程中的第 j 次 Givens 旋转，则 $\mathbf{Q}^T \mathbf{A} = \mathbf{R}$ 是上三角矩阵，其中， $\mathbf{Q} = \mathbf{G}_t \mathbf{G}_{t-1} \cdots \mathbf{G}_1$ ，而 t 是总的旋转次数。

例 4.4.5. 已知矩阵 $\mathbf{A} = \begin{pmatrix} 0 & 3 & 1 \\ 0 & 4 & -2 \\ 2 & 1 & 1 \end{pmatrix}$ ，利用 Givens 变换求 \mathbf{A} 的 QR 分解。

解. 因为 $a_{21} = 0, a_{31} = 2$ ，取 $c = \frac{0}{\sqrt{0^2+2^2}} = 0, s = \frac{2}{\sqrt{0^2+2^2}} = 1$ ，构造 $\mathbf{G}_{(2,3)}^{(1)} = \begin{pmatrix} 1 & & \\ & 0 & 1 \\ & -1 & 0 \end{pmatrix}$

$$\mathbf{A}^{(1)} = \mathbf{G}_{(2,3)}^{(1)} \mathbf{A} = \begin{pmatrix} 0 & 3 & 1 \\ 2 & 1 & 1 \\ 0 & 4 & -2 \end{pmatrix}$$

因为 $a_{11}^{(1)} = 0, a_{21}^{(1)} = 2$ ，取 $c = \frac{0}{\sqrt{0^2+2^2}} = 0, s = \frac{2}{\sqrt{0^2+2^2}} = 1$ ，构造 $\mathbf{G}_{(1,2)}^{(1)} = \begin{pmatrix} 0 & 1 & \\ -1 & 0 & \\ & & 1 \end{pmatrix}$

$$\mathbf{A}^{(2)} = \mathbf{G}_{(2,3)}^{(1)} \mathbf{A}^{(1)} = \begin{pmatrix} 2 & 1 & 1 \\ 0 & 3 & 1 \\ 0 & 4 & -2 \end{pmatrix}$$

因为 $a_{22}^{(2)} = 3, a_{32}^{(2)} = 4$, 取 $c = \frac{3}{\sqrt{3^2+4^2}} = \frac{3}{5}, s = \frac{4}{\sqrt{3^2+4^2}} = \frac{4}{5}$, 构造 $\mathbf{G}_{(2,3)}^{(2)} = \begin{pmatrix} 1 & & \\ & \frac{3}{5} & \frac{4}{5} \\ & -\frac{4}{5} & \frac{3}{5} \end{pmatrix}$

则

$$\mathbf{A}^{(3)} = \mathbf{G}_{(2,3)}^{(2)} \mathbf{A}^{(2)} = \begin{pmatrix} 2 & 1 & 2 \\ 0 & 5 & -1 \\ 0 & 0 & -2 \end{pmatrix} = \mathbf{R}$$

取

$$\mathbf{Q} = \mathbf{G}_{(2,3)}^{(1)} \mathbf{G}_{(1,2)}^{(1)} \mathbf{G}_{(2,3)}^{(2)} = \begin{pmatrix} 0 & \frac{3}{5} & -\frac{4}{5} \\ 0 & \frac{4}{5} & \frac{3}{5} \\ 1 & 0 & 0 \end{pmatrix}$$

则 $\mathbf{A} = \mathbf{QR}$ 。

设 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, 并且假定我们知道了 \mathbf{A} 的 QR 分解。

对 \mathbf{Q} 列分块:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ n & m-n \end{bmatrix}$$

并且令

$$\mathbf{Q}^T \mathbf{b} = \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix} \mathbf{b} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} \quad \begin{matrix} n \\ m-n \end{matrix}$$

那么

$$\|\mathbf{Ax} - \mathbf{b}\|_2^2 = \|\mathbf{Q}^T \mathbf{Ax} - \mathbf{Q}^T \mathbf{b}\|_2^2 = \|\mathbf{Rx} - \mathbf{c}_1\|_2^2 + \|\mathbf{c}_2\|_2^2$$

$\|\mathbf{c}_2\|_2^2$ 是和 \mathbf{x} 无关的。因此, $\mathbf{Rx} = \mathbf{c}_1$ 的解使得 $\|\mathbf{Ax} - \mathbf{b}\|_2^2$ 最小。而这个上三角方程组可以非常容易的利用回代法求解。

例 4.4.6. 已知线性方程组

$$x_1 + 2x_2 = 5$$

$$2x_1 + 3x_2 = 8$$

$$6x_1 + 7x_2 = 21$$

用 QR 分解求上述方程组的最小二乘解。

解: 记增广矩阵

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 5 \\ 2 & 3 & 8 \\ 6 & 7 & 21 \end{bmatrix}$$

对 A 做 QR 分解：

$$A = \begin{pmatrix} 0.1562 & 0.7711 & 0.6172 \\ 0.3123 & 0.5543 & -0.7715 \\ 0.9370 & -0.3133 & 0.1543 \end{pmatrix} \begin{pmatrix} 6.4031 & 7.8087 & 22.9575 \\ 0 & 1.0121 & 1.7110 \\ 0 & 0 & 0.1543 \end{pmatrix}$$

由此得线性方程组

$$6.4031x_1 + 7.8087x_2 = 22.9575$$

$$1.0121x_2 = 1.7110$$

解得

$$x_1 = 1.5238$$

$$x_2 = 1.6905$$

将这两个解代入原方程，得三个方程的拟合误差分别为 $e_1 = 0.0952$, $e_2 = 0.1191$ 和 $e_3 = 0.0237$ 。在这个任务中，我们也并不关心 Q 的结果。

4.5 谱分解与 Cholesky 分解

本节主要讨论两类特殊的矩阵：对称矩阵和半正定矩阵的分解。

- 对称矩阵的谱分解（特征分解）：可以把任意对称矩阵分解成三个矩阵的积，包括一个正交矩阵和一个实的对角矩阵。
- 正定矩阵的 Cholesky 分解：可以把任意对称正定矩阵分解成一个具有正的对角元的下三角矩阵和其转置的乘积。

特征值与物理或力学中振动的频谱相联系，所以特征分解也称为谱分解！

4.5.1 对称矩阵的谱分解

定理 4.5.1. 设 A 是实对称矩阵，则 A 的特征值皆为实数。

证明. 设 λ_0 是 A 的特征值，于是有非零向量

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

满足

$$A\mathbf{x} = \lambda_0 \mathbf{x}$$

令

$$\bar{\mathbf{x}} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \ddots \\ \bar{x}_n \end{pmatrix}$$

其中 \bar{x}_i 是 x_i 的共轭复数，则 $\bar{A}\bar{\mathbf{x}} = \bar{\lambda}_0\bar{\mathbf{x}}$ 。考察等式

$$\bar{\mathbf{x}}^T (\bar{A}\bar{\mathbf{x}}) = \bar{\mathbf{x}}^T \bar{A}^T \bar{\mathbf{x}} = (\bar{A}\bar{\mathbf{x}})^T \bar{\mathbf{x}} = (\bar{A}\bar{\mathbf{x}})^T \bar{\mathbf{x}}$$

其左边为 $\lambda_0 \bar{\mathbf{x}}^T \bar{\mathbf{x}}$ ，右边为 $\bar{\lambda}_0 \bar{\mathbf{x}}^T \bar{\mathbf{x}}$ 。故

$$\lambda_0 \bar{\mathbf{x}}^T \bar{\mathbf{x}} = \bar{\lambda}_0 \bar{\mathbf{x}}^T \bar{\mathbf{x}}$$

又因 \mathbf{x} 是非零向量

$$\bar{\mathbf{x}}^T \bar{\mathbf{x}} = \bar{x}_1 x_1 + \bar{x}_2 x_2 + \cdots + \bar{x}_n x_n \neq 0$$

故 $\lambda_0 = \bar{\lambda}_0$ ，即 λ_0 是一个实数。证毕。 \square

推论 4.5.1. 方阵 A 为正交矩阵的充分必要条件是 A 的列向量都是单位向量，且两两正交。

推论 4.5.2. 若 A 和 B 都是正交矩阵，则 AB 也是正交矩阵。

谱分解定理

定理 4.5.2. 设实矩阵 A 是 n 阶方阵，则下面 3 个命题等价：

1. $A = A^T$ 。
2. 存在一个正交矩阵 Q 使得 $Q^T A Q = \Lambda$ ，其中 Λ 是对角阵。
3. 存在 n 个 A 的特征向量构成 \mathbb{R}^n 的一个标准正交基。

证明. 1 \rightarrow 2:

利用数学归纳法，当 $n = 1$ 时，易知 1 推 2 成立。假设当 $k = n - 1$ 时结论成立。设 λ_1 是 $A \in \mathbb{R}^{n \times n}$ 的一个特征值，对应的特征向量为 \mathbf{q}_1 ，即 $A\mathbf{q}_1 = \lambda_1\mathbf{q}_1$ ，且我们令 $\|\mathbf{q}_1\| = 1$ 。我们可以将 (\mathbf{q}_1) 扩充成一个标准正交基 $(\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n)$ 从而得到了一个正交矩阵 Q ，又因为 $A\mathbf{q}_i \in \mathbb{R}^n$ 可以由 $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$ 线性表出，并且 $A\mathbf{q}_1 = \lambda_1\mathbf{q}_1$ ，那么

$$A\mathbf{Q} = A[\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n] = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n] \begin{pmatrix} \lambda_1 & \mathbf{a}^T \\ \mathbf{0} & C \end{pmatrix} = Q\Lambda_1$$

$$A\mathbf{Q} = A[\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n] = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n] \begin{pmatrix} \lambda_1 & \mathbf{a}^T \\ \mathbf{0} & C \end{pmatrix} = Q\Lambda_1$$

故 $A = Q\Lambda_1 Q^T$ ， $\Lambda_1 = Q^T A Q$ 。 A 对称，所以 Λ_1 对称，从而 $\mathbf{a}^T = \mathbf{0}^T, C$ 对称。

根据假设, 存在正交矩阵 \mathbf{Q}_1 使得 $\mathbf{Q}_1^T \mathbf{C} \mathbf{Q}_1 = \Lambda_1$, $\mathbf{C} = \mathbf{Q}_1 \Lambda_1 \mathbf{Q}_1^T$,

$$\mathbf{A}_1 = \begin{pmatrix} \lambda_1 & \\ & \mathbf{C} \end{pmatrix} = \begin{pmatrix} 1 & \\ & \mathbf{Q}_1 \end{pmatrix} \begin{pmatrix} \lambda_1 & \\ & \Lambda_1 \end{pmatrix} \begin{pmatrix} 1 & \\ & \mathbf{Q}_1^T \end{pmatrix}$$

即令 $\mathbf{Q}_2 = \mathbf{Q} \begin{pmatrix} 1 & \\ & \mathbf{Q}_1 \end{pmatrix}$, $\begin{pmatrix} \lambda_1 & \\ & \Lambda_1 \end{pmatrix} = \Lambda$ 即有 $\mathbf{A} = \mathbf{Q}_2 \Lambda \mathbf{Q}_2^T$, $\mathbf{Q}_2^T \mathbf{A} \mathbf{Q}_2 = \Lambda$ 。

2→3: 将 \mathbf{Q} 按列分块 $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$, 记 $\Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$

$$\mathbf{A}\mathbf{Q} = [\mathbf{A}\mathbf{q}_1, \mathbf{A}\mathbf{q}_2, \dots, \mathbf{A}\mathbf{q}_n] = \mathbf{Q}\Lambda = [\lambda_1\mathbf{q}_1, \lambda_2\mathbf{q}_2, \dots, \lambda_n\mathbf{q}_n]$$

故, $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$ 是矩阵 \mathbf{A} 的特征向量。

3→1: 令 $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$ 是矩阵 \mathbf{A} 的 n 个两两正交且模为 1 的特征向量。则 $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$ 是

一正交矩阵。记 $\Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$ 其中 λ_i 是 \mathbf{q}_i 对应的特征值。

$$\mathbf{A}\mathbf{Q} = [\mathbf{A}\mathbf{q}_1, \mathbf{A}\mathbf{q}_2, \dots, \mathbf{A}\mathbf{q}_n] = [\lambda_1\mathbf{q}_1, \lambda_2\mathbf{q}_2, \dots, \lambda_n\mathbf{q}_n] = \mathbf{Q}\Lambda$$

$$\text{故 } \mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^T \quad \mathbf{A}^T = \mathbf{Q}\Lambda^T\mathbf{Q}^T = \mathbf{Q}\Lambda\mathbf{Q}^T = \mathbf{A}.$$

□

谱分解定义

定义 4.5.1. 设对称矩阵 \mathbf{A} 为 n 阶方阵, 如果 \mathbf{A} 可以被分解为 $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^T$, 其中 $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$ 是由特征向量 $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$ 组成的 n 阶方阵, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ 是由特征值 $\lambda_1, \lambda_2, \dots, \lambda_n$ 组成的 n 阶对角矩阵, 则这种分解叫做对称矩阵的谱分解或者特征分解。

我们也可以将 $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^T$ 改写成秩-1 矩阵和的形式:

$$\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^T = \sum_{i=1}^n \lambda_i \mathbf{q}_i \mathbf{q}_i^T$$

求解对称方阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 的特征分解步骤

- 计算矩阵 \mathbf{A} 的特征值 $\lambda_1, \dots, \lambda_n$, 即求特征方程

$$|\mathbf{A} - \lambda \mathbf{I}| = 0$$

的 n 个根。

- 求特征值对应的 n 个相互正交的特征向量 $\mathbf{q}_1, \dots, \mathbf{q}_n$, 即求解方程组并单位化

$$\mathbf{A}\mathbf{q}_i = \lambda_i \mathbf{q}_i, i = 1, \dots, n$$

- 记矩阵 $\mathbf{Q} = (\mathbf{q}_1, \dots, \mathbf{q}_n)$ 。
- 最终得到矩阵 \mathbf{A} 的特征分解为

$$\mathbf{A} = \mathbf{Q} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \mathbf{Q}^T$$

例 4.5.1. 求实对称矩阵 $\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ 的特征分解.

(1) 计算特征值和正交单位特征向量.

解.

$$|\lambda\mathbf{I} - \mathbf{A}| = \begin{vmatrix} \lambda - 2 & -1 \\ -1 & \lambda - 2 \end{vmatrix} = 0$$

特征值

$$\lambda_1 = 3, \lambda_2 = 1$$

对应的特征向量通过求解

$$\mathbf{A}\mathbf{q}_1 = 3\mathbf{q}_1, \mathbf{A}\mathbf{q}_2 = \mathbf{q}_2,$$

并单位化得到, 所以有

$$\mathbf{q}_1 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)^T, \mathbf{q}_2 = \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)^T$$

(2) 写出左特征向量方阵 \mathbf{Q} 和特征值方阵 Λ .

解.

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2] = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \lambda_1 & \\ & \lambda_2 \end{bmatrix} = \begin{bmatrix} 3 & \\ & 1 \end{bmatrix}$$

又因为 \mathbf{A} 是实对称矩阵, 所以 $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 3 & \\ & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$.

(3) 我们也可以将其改写成秩-1矩阵和的形式:

$$\mathbf{A} = 3 \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} + 1 \cdot \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = 3 \cdot \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} + \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

瑞利商

定义 4.5.2. 设矩阵 $A \in \mathbb{R}^{n \times n}$ 为一对称阵, 比率式 $R(\mathbf{x}) = \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$, ($0 \neq \mathbf{x} \in \mathbb{R}^n$) 被称为瑞利商。

定理 4.5.3. 给定一个对称矩阵 $A \in \mathbb{R}^{n \times n}$, 瑞利商 $R(\mathbf{x})$ 有如下性质:

$$\lambda_{\min}(A) \leq R(\mathbf{x}) \leq \lambda_{\max}(A)$$

并且有

$$\lambda_{\max}(A) = \max_{\mathbf{x}: \|\mathbf{x}\|_2=1} \mathbf{x}^T A \mathbf{x}, \quad \lambda_{\min}(A) = \min_{\mathbf{x}: \|\mathbf{x}\|_2=1} \mathbf{x}^T A \mathbf{x},$$

当 $\mathbf{x} = \mathbf{u}_1$ 或 $\mathbf{x} = \mathbf{u}_n$ 时, 瑞利商取到最大值或最小值, 其中 $\mathbf{u}_1, \mathbf{u}_n$ 分别是最大特征值和最小特征值对应的特征向量。

证明. 矩阵 $A \in \mathbb{R}^{n \times n}$ 为一对称阵, 那么设它的 n 个特征值 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ 对应的标准正交的特征向量为 $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$ 。可以将 \mathbf{x} 表示为

$$\mathbf{x} = a_1 \mathbf{q}_1 + a_2 \mathbf{q}_2 + \dots + a_n \mathbf{q}_n$$

则有瑞利商分母

$$\mathbf{x}^T \mathbf{x} = \left(\sum_{i=0}^n a_i \mathbf{q}_i \right)^T \left(\sum_{j=0}^n a_j \mathbf{q}_j \right) = \sum_{i=0}^n a_i^2 \mathbf{q}_i^T \mathbf{q}_i = \sum_{i=0}^n a_i^2$$

而瑞利商分子

$$\begin{aligned} \mathbf{x}^T A \mathbf{x} &= \left(\sum_{i=0}^n a_i \mathbf{q}_i \right)^T A \left(\sum_{j=0}^n a_j \mathbf{q}_j \right) = \sum_{i=0}^n \sum_{j=0}^n a_i \mathbf{q}_i^T A a_j \mathbf{q}_j \\ &= \sum_{i=0}^n \sum_{j=0}^n a_i a_j \lambda_j \mathbf{q}_i^T \mathbf{q}_j = \sum_{i=0}^n a_i^2 \lambda_i \mathbf{q}_i^T \mathbf{q}_i = \sum_{i=0}^n a_i^2 \lambda_i \\ \mathbf{x}^T \mathbf{x} &= \sum_{i=0}^n a_i^2, \quad \mathbf{x}^T A \mathbf{x} = \sum_{i=0}^n a_i^2 \lambda_i \end{aligned}$$

又 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, 所以 $\lambda_n \mathbf{x}^T \mathbf{x} \leq \mathbf{x}^T A \mathbf{x} \leq \lambda_1 \mathbf{x}^T \mathbf{x}$,

$$\lambda_{\min}(A) \leq R(\mathbf{x}) \leq \lambda_{\max}(A)$$

当 $\mathbf{x} = \mathbf{u}_1 = a_1 \mathbf{q}_1$, ($a_1 \neq 0$) 时:

$$\mathbf{x}^T \mathbf{x} = a_1^2, \quad \mathbf{x}^T A \mathbf{x} = a_1^2 \lambda_1, \quad R(\mathbf{x}) = \lambda_{\max}(A)$$

当 $\mathbf{x} = \mathbf{u}_n = a_n \mathbf{q}_n$, ($a_n \neq 0$) 时:

$$\mathbf{x}^T \mathbf{x} = a_n^2, \quad \mathbf{x}^T A \mathbf{x} = a_n^2 \lambda_1, \quad R(\mathbf{x}) = \lambda_{\min}(A)$$

如果限制 $\|\mathbf{x}\| = 1$, 此时 $R(\mathbf{x}) = \mathbf{x}^T A \mathbf{x}$ 。当 $\mathbf{x} = \mathbf{q}_1$ 时, $\mathbf{x}^T A \mathbf{x}$ 取到极大值 λ_{\max} ; 当 $\mathbf{x} = \mathbf{q}_n$ 时, $\mathbf{x}^T A \mathbf{x}$ 取到极小值 λ_{\min} 。□

通过对瑞利商的讨论，我们也得到了如下结论：

$$\mathbf{A} \succeq 0 \iff \lambda_i(\mathbf{A}) \geq 0, i = 1, \dots, n$$

$$\mathbf{A} \succ 0 \iff \lambda_i(\mathbf{A}) > 0, i = 1, \dots, n$$

证明.

$$\mathbf{A} \succeq 0 \iff R(x) \geq 0 \iff \lambda_{\min}(\mathbf{A}) \geq 0$$

也就是说，如果 \mathbf{A} 是半正定矩阵，它的任何一个特征值都非负。

$$\mathbf{A} \succ 0 \iff R(x) > 0 \iff \lambda_{\min}(\mathbf{A}) > 0$$

也就是说，如果 \mathbf{A} 是正定矩阵，它的任何一个特征值都为正。 \square

- n 阶对称矩阵可以记为 S^n ,
- n 阶半正定矩阵可以记为 S_+^n ,
- n 阶正定矩阵可以记为 S_{++}^n 。

Poincare 不等式

定理 4.5.4. [Poincare 不等式] 令 $\mathbf{A} \in S^n$ ，并令 \mathbb{V} 是 \mathbb{R}^n 中的任意一个 k 维子空间，这里 $1 \leq k \leq n$ 。那么，存在单位向量 $\mathbf{x}, \mathbf{y} \in \mathbb{V}$, $\|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = 1$ ，使得

$$\mathbf{x}^\top \mathbf{A} \mathbf{x} \leq \lambda_k(\mathbf{A}), \quad \mathbf{y}^\top \mathbf{A} \mathbf{y} \geq \lambda_{n-k+1}(\mathbf{A})$$

证明. 令 $\mathbf{A} = \mathbf{U} \Lambda \mathbf{U}^\top$ 是 \mathbf{A} 的谱分解，记 $\mathbb{Q} = \text{Col}(\mathbf{U}_k)$ 是 $\mathbf{U}_k = [\mathbf{u}_1, \dots, \mathbf{u}_k]$ 张成的子空间。由于 \mathbb{Q} 是 $n - k + 1$ 维的， \mathbb{V} 维度为 k , $\mathbb{V} \cap \mathbb{Q}$ 一定是非空的。选取一个单位向量 $\mathbf{x} \in \mathbb{V} \cap \mathbb{Q}$ 。则存在 $\boldsymbol{\eta}, \|\boldsymbol{\eta}\| = 1$ 使得 $\mathbf{x} = \mathbf{U}_k \boldsymbol{\eta}$ ，那么

$$\begin{aligned} \mathbf{x}^\top \mathbf{A} \mathbf{x} &= \boldsymbol{\eta}^\top \mathbf{U}_k^\top \mathbf{U} \Lambda \mathbf{U}^\top \mathbf{U}_k \boldsymbol{\eta} = \sum_{i=k}^n \lambda_i(\mathbf{A}) \eta_i^2 \\ &\leq \lambda_k(\mathbf{A}) \sum_{i=k}^n \eta_i^2 = \lambda_k(\mathbf{A}) \end{aligned}$$

这就证明了命题中的第一个不等式。对于第二个不等式，我们可以对 $-\mathbf{A}$ 用同样的处理方式即可证明。 \square

极小极大准则 (Minimax principle)

推论 4.5.3. [极小极大准则] 令 $\mathbf{A} \in S^n$ ，并令 \mathbb{V} 是 \mathbb{R}^n 中的任意一个 k 维子空间，这里 $1 \leq k \leq n$ 。那么，对于 $k \in \{1, \dots, n\}$ ，有

$$\begin{aligned} \lambda_k(\mathbf{A}) &= \max_{\dim \mathbb{V}=k} \min_{\mathbf{x} \in \mathbb{V}, \|\mathbf{x}\|_2=1} \mathbf{x}^\top \mathbf{A} \mathbf{x} \\ &= \min_{\dim \mathbb{V}=n-k+1} \max_{\mathbf{x} \in \mathbb{V}, \|\mathbf{x}\|_2=1} \mathbf{x}^\top \mathbf{A} \mathbf{x} \end{aligned}$$

证明. 根据 Poincare 不等式, 如果 \mathbb{V} 是 \mathbb{R}^n 的 k 维子空间, 那么 $\min_{\mathbf{x} \in \mathbb{V}, \|\mathbf{x}\|_2=1} \mathbf{x}^\top \mathbf{A} \mathbf{x} \leq \lambda_k(\mathbf{A})$ 。如果我们令 $\mathbb{V} = \text{span}\{\mathbf{u}_1, \dots, \mathbf{u}_k\}$, 那么我们就得到了第一个等式。对 $-\mathbf{A}$ 用同样的处理方式, 我们会得到第二个等式。□

极小极大准则可以用于比较两个对称矩阵和的特征值和原矩阵特征值的大小关系。

推论 4.5.4. 令 $\mathbf{A}, \mathbf{B} \in S^n$, 对每个 $k = 1, \dots, n$, 有

$$\lambda_k(\mathbf{A}) + \lambda_{\min}(\mathbf{B}) \leq \lambda_k(\mathbf{A} + \mathbf{B}) \leq \lambda_k(\mathbf{A}) + \lambda_{\max}(\mathbf{B})$$

证明. 根据推论 1, 我们有

$$\begin{aligned} \lambda_k(\mathbf{A} + \mathbf{B}) &= \min_{\dim \mathbb{V}=n-k+1} \max_{\mathbf{x}: \|\mathbf{x}\|_2=1} (\mathbf{x}^\top \mathbf{A} \mathbf{x} + \mathbf{x}^\top \mathbf{B} \mathbf{x}) \\ &\geq \min_{\dim \mathbb{V}=n-k+1} \max_{\mathbf{x}: \|\mathbf{x}\|_2=1} \mathbf{x}^\top \mathbf{A} \mathbf{x} + \lambda_{\min}(\mathbf{B}) \\ &= \lambda_k(\mathbf{A}) + \lambda_k(\mathbf{B}) \end{aligned}$$

这就证明了命题中不等式的左半部分, 对于右半部分, 可以用类似的方法证明。□

4.5.2 正半定矩阵与 Cholesky 分解

LU 分解的本质是一种三角化分解, 即将矩阵分解为一个上三角矩阵和下三角矩阵的乘积, 而这一类分解中, 还有另一种分解: 设 $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n \times n}$ 是对称正定矩阵, $\mathbf{A} = \mathbf{G}\mathbf{G}^\top$ 称为矩阵 \mathbf{A} 的 Cholesky 分解, 其中, $\mathbf{G} \in \mathbb{R}^{n \times n}$ 是一个具有正的对角线元素的下三角矩阵, 即

$$\mathbf{G} = \begin{pmatrix} g_{11} & & & \\ g_{21} & g_{22} & & \\ \vdots & \vdots & \ddots & \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{pmatrix} \quad (4.11)$$

比较 $\mathbf{A} = \mathbf{G}\mathbf{G}^\top$ 两边, 易得

$$a_{ij} = \sum_{k=1}^j g_{jk} g_{ik}$$

从而有

$$g_{jj} g_{ij} = a_{ij} - \sum_{k=1}^{j-1} g_{jk} g_{ik} = v(i) \quad (4.12)$$

如果知道了 \mathbf{G} 的前 $j-1$ 列, 那么 $v(i)$ 就是可计算的。

在式 (4.12) 中令 $i = j$, 立即有 $g_{jj}^2 = v(j)$ 。然后, 由式 (4.12) 得

$$g_{ij} = v(i)/g_{jj} = v(i)/\sqrt{v(j)} \quad (4.13)$$

总结以上结论, 可得到计算 Cholesky 分解的下述 MATLAB 算法:

以上分析结果可以归纳为下面的定理。

Algorithm 6 Cholesky 分解

```

1: for  $j = 1 : n$  do
2:   for  $i = j : n$  do
3:      $v(i) = a_{ij};$ 
4:     for  $k = 1 : j - 1$  do
5:        $v(i) = v(i) - g_{jk}g_{ik};$ 
6:     end for
7:      $g_{ij} = v(i)/\sqrt{v(j)};$ 
8:   end for
9: end for

```

定理 4.5.5. [Cholesky 分解] 如果 $A \in \mathbf{R}^{n \times n}$ 是对称正定矩阵, 则 Cholesky 分解 $A = \mathbf{G}\mathbf{G}^T$ 是唯一的, 其中, 下三角矩阵 $\mathbf{G} \in \mathbf{R}^{n \times n}$ 的非零元素由式 (4.13) 决定。

例 4.5.2. 求矩阵 A 的 Cholesky 分解

$$A = \begin{bmatrix} 4 & -1 & -1 \\ -1 & 4.25 & 2.75 \\ 1 & 2.75 & 3.5 \end{bmatrix}$$

解. 显然 $A^T = A$, 特征值 $\lambda_1 = 1.15 > 0, \lambda_2 = 3.9 > 0, \lambda_3 = 6.7 > 0$, 因此, A 为对称正定矩阵。故存在 $A = \mathbf{G}\mathbf{G}^T$, 则有:

$$g_{11} = \sqrt{a_{11}} = 2, g_{21} = \frac{a_{21}}{g_{11}} = -0.5, g_{31} = \frac{a_{31}}{g_{11}} = 0.5,$$

$$g_{22} = \sqrt{a_{22} - g_{21}^2} = 2,$$

$$g_{32} = \frac{a_{32} - g_{31}g_{21}}{g_{22}} = 1.5$$

$$g_{33} = \sqrt{a_{33} - g_{31}^2 - g_{32}^2} = 1$$

可得:

$$\mathbf{G} = \begin{bmatrix} 2 & 0 & 0 \\ -0.5 & 2 & 0 \\ 0.5 & 1.5 & 1 \end{bmatrix}$$

4.5.3 改进的 Cholesky 分解

为了避免开方运算, 我们可以将 A 分解为: $A = \mathbf{L}\mathbf{D}\mathbf{L}^T$, 即

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \ddots & \vdots & \ddots & \ddots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \ddots & \ddots & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix} \begin{pmatrix} 1 & l_{21} & \cdots & l_{n1} \\ & 1 & \cdots & l_{n2} \\ & & \ddots & \ddots \\ & & & 1 \end{pmatrix}$$

使用待定系数法可得

$$a_{ij} = \sum_{k=1}^n l_{ik} d_k l_{jk} = d_j l_{ij} + \sum_{k=1}^{j-1} l_{ik} d_k l_{jk}, \quad i, j = 1, 2, \dots, n$$

基于以上分解来求解对称正定线性方程组的算法称为改进的 cholesky 法：

Algorithm 7 改进的 Cholesky 分解

```

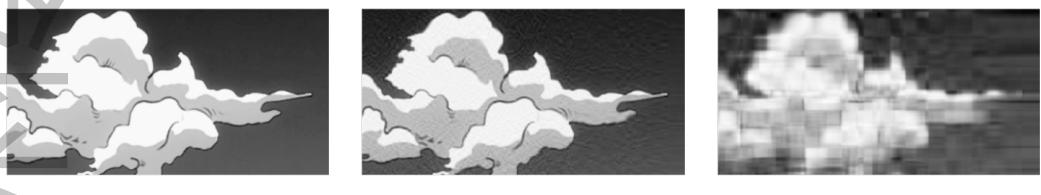
1: for  $j = 1 : n$  do
2:    $d_j = a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2 d_k$ 
3:   for  $i = j + 1 : n$  do
4:      $l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_k l_{jk}) / d_j;$ 
5:   end for
6: end for
7:  $y_1 = b_1$ 
8: for  $i = 2 : n$  do
9:    $y_i = b_i - \sum_{k=1}^{i-1} l_{ik} y_k$ 
10: end for
11:  $x_n = \frac{y_n}{d_n};$ 
12: for  $i = n - 1 : 1$  do
13:    $x_i = y_i / d_i - \sum_{k=i+1}^n l_{ki} x_k$ 
14: end for
```

4.6 奇异值分解

奇异值分解 (Singular Value Decomposition, SVD) 是线性代数和矩阵论中一种重要的矩阵分解技术。1873 年, Beltrami 给出实正方阵的奇异值分解, 1874 年, Jordan 也独立推导出实正方阵的奇异值分解, 1902 年, Autonne 把奇异值分解推广到复方阵, 1939 年, Eckhart 和 Young 进一步把它推广到复长方形矩阵。

4.6.1 引例

奇异值分解在数据分析、信号处理和模式识别等方面都具有广泛应用，比如在图像压缩领域，图像数据中通常存在冗余，包括：图像中相邻像素间的相关性引起的空冗余；图像序列中不同帧之间存在相关性引起的时间冗余；不同彩色平面或频谱带的相关性引起的频谱冗余。可以通过图像压缩处理来减少图像数据中的冗余信息从而用更加高效的格式存储和传输数据，其原理就是通过图像矩阵分解理论减少表示数字图像时需要的数据量，比如通过矩阵的特征分解，提取较大的特征值，舍弃比较小的特征值。还是因为特征值代表了信息量，所以保留比较大的特征值、舍弃比较小的特征值，从而达到图像矩阵压缩的目的。但是由于特征值分解压缩图片存在着不可靠性，所以通常会采用矩阵的奇异值分解，把获得的奇异值，取其中比较大的奇异值（类同特征值提取的压缩方法），舍去较小的奇异值，以达到数字图像压缩的目的。图像矩阵的奇异值及其特征空间反映了图像中的不同成分和特征。一般认为较大的奇异值及其对应的奇异向量表示图像信号，而噪声反映在较小的奇异值及其对应的奇异向量上，依据一定的准则选择门限，低于该门限的奇异值置零（截断），然后通过这些奇异值和其对应的奇异向量重构图像进行去噪。若考虑图像的局部平稳性，也可以对图像分块进行奇异值分解去噪，这样能在一定程度上保护图像的边缘细节。



(a) 原图

(b) 提取 50 个奇异值的图像

(c) 提取 10 个奇异值的图像

图 4.12: 使用 SVD 进行图像压缩

4.6.2 奇异值分解

定义 4.6.1. 矩阵的奇异值分解是指，将一个非零的 $m \times n$ 实矩阵 A , $A \in \mathbb{R}^{m \times n}$ 表示为以下三个实矩阵乘积的形式，即进行矩阵的因子分解：

$$A = U \Sigma V^T \quad (4.14)$$

其中 U 是 m 阶正交矩阵， V 是 n 阶正交矩阵， Σ 是由降序排列的非负的对角线元素组成的 $m \times n$ 矩形对角矩阵，满足

$$UU^T = I, \quad VV^T = I, \quad \Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_p), \quad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p, \quad p = \min(m, n)$$

$U\Sigma V^T$ 称为矩阵 A 的奇异值分解， σ_i 称为 A 的奇异值， U 的列向量称为左奇异向量， V 的列向量称为右奇异向量。

$$\begin{matrix} m \\ \textcolor{teal}{A} \\ n \end{matrix} = \begin{matrix} m \\ \textcolor{brown}{U} \\ m \end{matrix} \begin{matrix} m \\ \Sigma \\ n \end{matrix} \begin{matrix} n \\ \textcolor{brown}{V}^T \\ n \end{matrix}$$

图 4.13: 完全奇异值分解

例 4.6.1. 矩阵 $A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$ 的奇异值分解为：

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & \sqrt{0.2} & -\sqrt{0.8} & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \sqrt{0.8} & \sqrt{0.2} & 0 \end{pmatrix} \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & \sqrt{5} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

4.6.3 奇异值分解的几何解释

从线性变换的角度理解奇异值分解， $m \times n$ 矩阵 A 表示从 n 维空间 \mathbb{R}^n 到 m 维空间 \mathbb{R}^m 的一个线性变换，

$$\mathcal{T}: \mathbf{x} \mapsto A\mathbf{x}$$

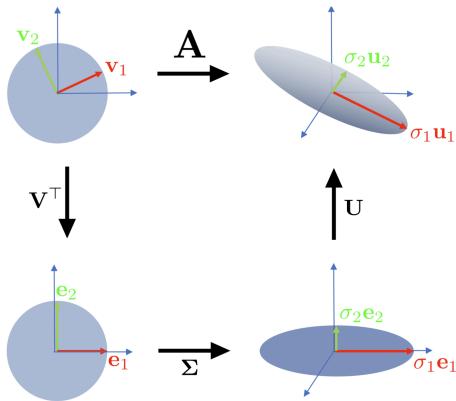
$\mathbf{x} \in \mathbb{R}^n, A\mathbf{x} \in \mathbb{R}^m$, \mathbf{x} 和 $A\mathbf{x}$ 分别是各自空间的向量。线性变换可以分解为三个简单的变换：

- 一个坐标系的旋转或者反射变换
- 一个坐标轴的缩放变换
- 另一个坐标系的旋转或者反射变换

奇异值定理保证这种分解一定存在。这就是奇异值分解的几何解释。

- 对矩阵 A 进行奇异值分解，得到 $A = U\Sigma V^T$, V 和 U 都是正交矩阵
- 所以 V 的列向量 v_1, v_2, \dots, v_n 构成 \mathbb{R}^n 空间的一组标准正交基，表示 \mathbb{R}^n 中的正交坐标系的旋转或反射变换
- U 的列向量 u_1, u_2, \dots, u_m 构成 \mathbb{R}^m 空间的一组标准正交基，表示 \mathbb{R}^m 中的正交坐标系的旋转或者反射变换
- Σ 的对角元素 $\sigma_1, \sigma_2, \dots, \sigma_n$ 是一组非负实数，表示 \mathbb{R}^n 中的原始正交坐标系坐标轴的 $\sigma_1, \sigma_2, \dots, \sigma_n$ 倍的缩放变换。

- 任意一个向量 $\mathbf{x} \in \mathbb{R}^n$, 经过基于 $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ 的线性变换, 等价于经过坐标系的旋转或者反射变换 \mathbf{V}^T , 坐标轴的缩放变换 Σ , 以及坐标系的旋转或者反射变换 \mathbf{U} 得到向量 $\mathbf{Ax} \in \mathbb{R}^m$ 。
- 对于 SVD 来说, 分别改变了 \mathbb{R}^n 和 \mathbb{R}^m 两个空间的基底。而特征分解仅仅是在同一个空间中做变换。



下面通过一个例子直观地说明奇异值分解的几何意义。

例 4.6.2. 给定一个 2 阶矩阵

$$\mathbf{A} = \begin{pmatrix} 3 & 1 \\ 2 & 1 \end{pmatrix}$$

其奇异值分解为

$$\mathbf{U} = \begin{pmatrix} 0.8174 & -0.5760 \\ 0.5760 & 0.8174 \end{pmatrix}, \Sigma = \begin{pmatrix} 3.8643 & 0 \\ 0 & 0.2588 \end{pmatrix}, \mathbf{V} = \begin{pmatrix} 0.9327 & 0.3606 \\ -0.3606 & 0.9327 \end{pmatrix}$$

观察基于矩阵 \mathbf{A} 的奇异值分解将 \mathbb{R}^2 的标准正交基

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

进行线性转换的情况。

首先, \mathbf{V}^T 表示一个旋转变换, 将标准正交基 $\mathbf{e}_1, \mathbf{e}_2$ 旋转, 得到向量

$$\mathbf{V}^T \mathbf{e}_1 = \begin{pmatrix} 0.9327 \\ -0.3606 \end{pmatrix}, \mathbf{V}^T \mathbf{e}_2 = \begin{pmatrix} 0.3606 \\ 0.9327 \end{pmatrix}$$

其次, Σ 表示一个缩放变换, 将向量 $\mathbf{V}^T \mathbf{e}_1, \mathbf{V}^T \mathbf{e}_2$ 在坐标轴方向缩放 σ_1 倍和 σ_2 倍, 得到向量

$$\Sigma \mathbf{V}^T \mathbf{e}_1 = \begin{pmatrix} 3.6042 \\ -0.0933 \end{pmatrix}, \Sigma \mathbf{V}^T \mathbf{e}_2 = \begin{pmatrix} 1.3935 \\ 0.2414 \end{pmatrix}$$

最后, \mathbf{U} 表示一个旋转变换, 再将向量 $\Sigma V^T e_1, \Sigma V^T e_2$ 旋转得到

$$Ae_1 = \mathbf{U} \Sigma V^T e_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, Ae_2 = \mathbf{U} \Sigma V^T e_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

综上, 矩阵的奇异值分解也可以看作是将其对应的线性变换分解为旋转变换、缩放变换以及旋转变换的组合。这一组合是一定存在的。

4.6.4 紧奇异值分解和截断奇异值分解

定义 4.6.2. 设有 $m \times n$ 实矩阵 A , 其秩 $\text{rank}(A) = r, r \leq \min(m, n)$, 则称 $\mathbf{U}_r \Sigma_r V_r^T$ 为 A 的紧奇异值分解, 即

$$A = \mathbf{U}_r \Sigma_r V_r^T$$

其中 $\mathbf{U}_r \in \mathbb{R}^{m \times r}, V_r \in \mathbb{R}^{n \times r}$, Σ_r 是 r 阶对角矩阵; 矩阵 \mathbf{U}_r 由完全奇异值分解中 \mathbf{U} 的前 r 列、矩阵 V_r 由 V 的前 r 列、矩阵 Σ_r 由 Σ 的前 r 个对角线元素得到。紧奇异值分解的对角矩阵 Σ_r 的秩与原始矩阵 A 的秩相等。

$$\begin{matrix} m & | & A & = & m & | & U & | & r & | & \Sigma & | & r & | & V^T & | & n \\ & r & & & r & & r & & r & & & & & & & & & n \end{matrix}$$

图 4.14: 紧 SVD

把完整 SVD 中的 \mathbf{U} 和 Σ 中多余的部分去掉。

例 4.6.3. 由例 4.6.1 给出的矩阵 $A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$ 的秩 $r = 3$, 其紧奇异值分解为:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & \sqrt{0.2} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sqrt{0.8} \end{pmatrix} \begin{pmatrix} 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & \sqrt{5} \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

定义 4.6.3. 设有 $m \times n$ 实矩阵 A , 其秩 $\text{rank}(A) = r$, 且 $0 < k < r$, 则称 $\mathbf{U}_k \Sigma_k V_k^T$ 为矩阵 A 的截断奇异值分解, 即

$$A \approx \mathbf{U}_k \Sigma_k V_k^T$$

其中 $\mathbf{U}_k \in \mathbb{R}^{m \times k}$, $\mathbf{V}_k \in \mathbb{R}^{n \times k}$, Σ_k 是 k 阶对角矩阵; 矩阵 \mathbf{U}_k 由完全奇异值分解中 \mathbf{U} 的前 k 列、矩阵 \mathbf{V}_k 由 \mathbf{V} 的前 k 列、矩阵 Σ_k 由 Σ 的前 k 个对角线元素得到。截断奇异值分解的对角矩阵 Σ_k 的秩比原始矩阵 \mathbf{A} 的秩低。

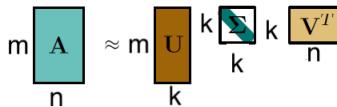


图 4.15: 截断 svd

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$$

例 4.6.4. 由例 4.6.1 给出的矩阵 $\mathbf{A} =$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} \approx \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 4 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

在实际应用中, 常常需要对矩阵的数据进行压缩, 将其近似表示, 奇异值分解提供了一种方法。后面将要叙述, 奇异值分解是在平方损失意义下对矩阵的最优近似。紧奇异值对应着无损压缩, 截断奇异值分解对应着有损压缩。

4.6.5 奇异值分解基本定理

对于对称矩阵来说, 奇异值分解总是存在的。因为, 我们知道如果 \mathbf{A} 是对称矩阵, 那么存在一个正交矩阵 \mathbf{P} 使得 \mathbf{A} 有特征分解

$$\mathbf{A} = \mathbf{P}\Sigma\mathbf{P}^T$$

此时我们令 $\mathbf{U} = \mathbf{P} = \mathbf{V}$ 那么就有

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$$

所以对称矩阵的奇异值分解就是他们的特征分解。

下面, 我们将给出一般矩阵奇异值分解的存在性证明, 并给出构造的一般方法。

定理 4.6.1. 奇异值分解基本定理 若 A 为一 $m \times n$ 实矩阵, $A \in \mathbb{R}^{m \times n}$, 则 A 的奇异值分解存在

$$A = U\Sigma V^T$$

其中 U 是 m 阶正交矩阵, V 是 n 阶正交矩阵, Σ 是 $m \times n$ 对角矩阵, 其前 r 个对角元素 $(\sigma_1, \dots, \sigma_r)$ 为正, 且按降序排列, 其余均为 0。

证明. 考虑矩阵 $A^T A$, 这个矩阵是对称半正定的, 所以我们可以对其进行谱分解

$$A^T A = V \Lambda_n V^T$$

其中 $V \in \mathbb{R}^{n \times n}$ 是正交矩阵, Λ_n 是对称矩阵, 并且对角线元素是 $A^T A$ 的特征值 $\lambda_i \geq 0, i = 1, \dots, n$, 并且是按降序排列的。因为 $\text{rank}(A) = \text{rank}(A^T A) = r$, 所以前 r 个特征值是正的。

注意到 AA^T 和 $A^T A$ 有相同的非零特征值, 因此他们的秩是相等的。我们定义

$$\sigma_i = \sqrt{\lambda_i} > 0, i = 1, \dots, r$$

记 v_1, \dots, v_r 是 V 的前 r 列, 它们同时也是 $A^T A$ 前 r 个特征值对应的特征向量。即有

$$A^T A v_i = \lambda_i v_i, i = 1, \dots, r$$

因此同时在两边左乘上 A 就有

$$(AA^T)Av_i = \lambda_i Av_i, i = 1, \dots, r$$

这就意味着 Av_i 是 AA^T 的特征向量, 因为 $v_i^T A^T A v_j = \lambda_j v_i^T v_j$ 所以这些特征向量也是正交的。所以将他们标准化则有

$$u_i = \frac{Av_i}{\sqrt{\lambda_i}} = \frac{Av_i}{\sigma_i}, i = 1, \dots, r$$

这些 u_1, \dots, u_r 是 r 个 AA^T 关于非零特征值 $\lambda_1, \dots, \lambda_r$ 的特征向量。

因此

$$u_i^T A v_j = \frac{1}{\sigma_i} v_i^T A^T A v_j = \frac{\lambda_j}{\sigma_i} v_i^T v_j = \begin{cases} \sigma_i & i = j \\ 0 & \text{otherwise} \end{cases}$$

以矩阵的方式重写即有

$$\begin{pmatrix} u_1^T \\ \vdots \\ u_r^T \end{pmatrix} A \begin{pmatrix} v_1, \dots, v_r \end{pmatrix} = \text{diag}(\sigma_1, \dots, \sigma_r) = \Sigma_r \quad (4.15)$$

至此就证明了紧 SVD。我们下面继续证明完全 SVD。注意到根据定义

$$A^T A v_i = 0, i = r+1, \dots, n$$

即有

$$Av_i = 0, i = r+1, \dots, n$$

为了说明上述等式成立, 我们假设 $\mathbf{A}^T \mathbf{A} \mathbf{v}_i = 0$ 且 $\mathbf{A} \mathbf{v}_i \neq 0$, 这意味着 $\mathbf{A} \mathbf{v}_i \in \text{Null}(\mathbf{A}^T) \equiv \text{Col}(\mathbf{A})^\perp$, 这与 $\mathbf{A} \mathbf{v}_i \in \text{Col}(\mathbf{A})$ 矛盾。所以 $\mathbf{A} \mathbf{v}_i = 0, i = r+1, \dots, n$ 。然后我们取相互正交的单位向量 $\mathbf{u}_{r+1}, \dots, \mathbf{u}_m$ 均与 $\mathbf{u}_1, \dots, \mathbf{u}_r$ 正交, 即有

$$\mathbf{u}_i^T \mathbf{A} \mathbf{v}_j = 0, i = 1, \dots, m; j = r+1, \dots, n$$

它们一起形成 \mathbb{R}^m 的一组标准正交基。因此, 扩展前述紧奇异值分解(4.15)有

$$\begin{pmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_m^T \end{pmatrix} \mathbf{A} \begin{pmatrix} \mathbf{v}_1, \dots, \mathbf{v}_n \end{pmatrix} = \begin{pmatrix} \Sigma_r & \mathbf{0}^T \\ \mathbf{0} & \mathbf{O} \end{pmatrix} = \Sigma$$

令 $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_m)$, $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_n)$ 就能得到 SVD 分解

$$\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T$$

至此就证明了矩阵 \mathbf{A} 存在奇异值分解。

□

4.6.6 奇异值分解的计算

奇异值分解定理的证明过程蕴含了奇异值分解的计算方法。矩阵 \mathbf{A} 的奇异值分解可以通过求对称矩阵 $\mathbf{A}^T \mathbf{A}$ 的特征值和特征向量得到。 $\mathbf{A}^T \mathbf{A}$ 的特征向量构成正交矩阵 \mathbf{V} 的列; $\mathbf{A}^T \mathbf{A}$ 的特征值 λ_j 的平方根为奇异值 σ_j , 即

$$\sigma_j = \sqrt{\lambda_j}, j = 1, 2, \dots, n$$

对其由大到小排列作为对角线元素, 构成对角矩阵 Σ ; 求正奇异值对应的左奇异向量, 再求扩充的 \mathbf{A}^T 的标准正交基, 构成正交矩阵 \mathbf{U} 的列。从而得到 \mathbf{A} 的奇异值分解 $\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T$ 。

给定 $m \times n$ 矩阵 \mathbf{A} , 可以根据上面的叙述写出奇异值分解的计算过程

- 首先求 $\mathbf{A}^T \mathbf{A}$ 的特征值和特征向量。计算对称矩阵 $\mathbf{W} = \mathbf{A}^T \mathbf{A}$,

求解特征方程

$$(\mathbf{W} - \lambda \mathbf{I}) \mathbf{x} = 0$$

得到特征值 λ_i , 并将特征值由大到小排列

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$$

将特征值 $\lambda_i (i = 1, 2, \dots, n)$ 代入特征方程求得对应的特征向量。

- 求 n 阶正交矩阵 \mathbf{V} 。将特征向量单位化得到 $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$, 构成 n 阶正交矩阵 \mathbf{V} 即

$$\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n)$$

- 求 $m \times n$ 对角矩阵 Σ 。计算 \mathbf{A} 的奇异值

$$\sigma_i = \sqrt{\lambda_i}, i = 1, 2, \dots, n$$

构造 $m \times n$ 矩形对角矩阵 Σ , 主对角线元素是奇异值, 其余元素是零

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$$

- 求 m 阶正交矩阵 \mathbf{U} , 对 \mathbf{A} 的前 r 个正奇异值, 令

$$\mathbf{u}_j = \frac{1}{\sigma_j} \mathbf{A} \mathbf{v}_j, j = 1, 2, \dots, r$$

得到

$$\mathbf{U}_1 = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r)$$

求 \mathbf{A}^T 的零空间的一组标准正交基 $\{\mathbf{u}_{r+1}, \dots, \mathbf{u}_m\}$, 令

$$\mathbf{U}_2 = (\mathbf{u}_{r+1}, \dots, \mathbf{u}_m)$$

并令

$$\mathbf{U} = (\mathbf{U}_1, \mathbf{U}_2)$$

- 得到奇异值分解

$$\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T$$

例 4.6.5. 试求矩阵 $\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 0 & 0 \end{pmatrix}$ 的奇异值分解

解. 求对称矩阵

$$\mathbf{A}^T \mathbf{A} = \begin{pmatrix} 5 & 5 \\ 5 & 5 \end{pmatrix}$$

求 $\mathbf{A}^T \mathbf{A}$ 的特征值与特征向量, 即求

$$\lambda^2 - 10\lambda = 0$$

所以特征值为 $\lambda_1 = 10, \lambda_2 = 0$ 从而得到 $\mathbf{v}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mathbf{v}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ 所以正交矩阵

$$\mathbf{V} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},$$

奇异值为 $\sigma_1 = \sqrt{10}, \sigma_2 = 0$ 所以对角矩阵为

$$\Sigma = \begin{pmatrix} \sqrt{10} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

再求正交矩阵 \mathbf{U} , 基于 \mathbf{A} 的正奇异值计算得到列向量

$$\mathbf{u}_1 = \frac{1}{\sigma_1} \mathbf{A} \mathbf{v}_1 = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$$

而列向量 $\mathbf{u}_2, \mathbf{u}_3$ 是 \mathbf{A}^T 零空间 $\text{Null}(\mathbf{A}^T)$ 的一组标准正交基, 所以

$$\mathbf{u}_2 = \frac{1}{\sqrt{5}} \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}, \mathbf{u}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

故正交矩阵 \mathbf{U} 为

$$\mathbf{U} = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & -2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & \sqrt{5} \end{pmatrix}$$

所以 \mathbf{A} 的奇异值分解为

$$\mathbf{A} = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & -2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & \sqrt{5} \end{pmatrix} \begin{pmatrix} \sqrt{10} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

4.6.7 奇异值分解和特征分解

性质 4.6.1. 设矩阵 \mathbf{A} 的奇异值分解为 $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$, 则以下关系成立:

$$\mathbf{A}^T\mathbf{A} = (\mathbf{U}\Sigma\mathbf{V}^T)^T(\mathbf{U}\Sigma\mathbf{V}^T) = \mathbf{V}(\Sigma^T\Sigma)\mathbf{V}^T$$

$$\mathbf{A}\mathbf{A}^T = (\mathbf{U}\Sigma\mathbf{V}^T)(\mathbf{U}\Sigma\mathbf{V}^T)^T = \mathbf{U}(\Sigma\Sigma^T)\mathbf{U}^T$$

也就是说, 矩阵 $\mathbf{A}^T\mathbf{A}, \mathbf{A}\mathbf{A}^T$ 的特征分解存在, 且可以由矩阵 \mathbf{A} 的奇异值分解的矩阵表示。 \mathbf{V} 的列向量是 $\mathbf{A}^T\mathbf{A}$ 的特征向量, \mathbf{U} 的列向量是 $\mathbf{A}\mathbf{A}^T$ 的特征向量, Σ 是奇异值是 $\mathbf{A}^T\mathbf{A}, \mathbf{A}\mathbf{A}^T$ 的特征值的平方根。

性质 4.6.2. 矩阵 \mathbf{A} 的奇异值分解中, 奇异值 $\sigma_1, \sigma_2, \dots, \sigma_n$ 是唯一的, 而矩阵 \mathbf{U}, \mathbf{V} 不是唯一的。

在矩阵 \mathbf{A} 的奇异值分解中, 奇异值、左奇异向量和右奇异向量之间存在对应关系。

性质 4.6.3. 设矩阵 \mathbf{A} 的奇异值分解为 $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$, 则以下关系成立:

$$\mathbf{A}\mathbf{v}_j = \sigma_j \mathbf{u}_j, j = 1, 2, \dots, n$$

$$\begin{cases} \mathbf{A}^T \mathbf{u}_j = \sigma_j \mathbf{v}_j & j = 1, 2, \dots, n \\ \mathbf{A}^T \mathbf{u}_j = 0 & j = n+1, n+2, \dots, m \end{cases}$$

证明. 由 $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$, 易知 $\mathbf{A}\mathbf{V} = \mathbf{U}\Sigma$ 。比较这一等式两端的第 j 列, 得到 $\mathbf{A}\mathbf{v}_j = \sigma_j \mathbf{u}_j, j = 1, 2, \dots, n$, 这是矩阵 \mathbf{A} 的右奇异向量和奇异值、左奇异向量的关系。

类似地, 我们可以得到另外一组关于矩阵 \mathbf{A} 的左奇异向量和奇异值、右奇异向量的关系。□

考虑矩阵 \mathbf{A} 的特征分解 $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ 和奇异值分解 $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ 。

- 对于任何矩阵 $A \in \mathbb{R}^{n \times m}$, SVD 始终存在。特征分解仅针对方阵 $A \in \mathbb{R}^{n \times n}$ 定义的，并且只有在我们可以找到 n 个相互独立的特征向量时才存在。
- 特征分解矩阵中的向量不一定是正交的，因此对基的改变并不是简单的旋转和缩放。另一方面，SVD 中矩阵 U 和 V 是正交矩阵，因此它们可以表示旋转或反射。
- 特征分解和 SVD 都是三个线性映射的组合：

 - 改变空间的基底
 - 在每个新基底方向上进行独立缩放并且从一个空间映射到另外一个空间。
 - 改变另外一个空间的基底

- 特征分解和 SVD 之间的主要区别在于，在 SVD 中，上述两个空间可以是不同维的向量空间。
- 在 SVD 中，左右奇异向量矩阵 U 和 V 通常不是互为逆矩阵。在特征分解中，特征向量矩阵 P 和 P^{-1} 是互为逆矩阵。
- 在 SVD 中，对角矩阵 Σ 中的项都是实数且非负，对于特征分解中的对角矩阵来说通常不成立。
- SVD 和特征分解通过他们的投影被紧密联系
 - A 的左奇异向量是 AA^T 的特征向量
 - A 的右奇异向量是 A^TA 的特征向量
 - A 非零奇异值是 A^TA 非零特征值的开方，同时也是 AA^T 非零特征值的开方
- 对于对称矩阵的特征分解和 SVD 是相同的。

4.6.8 基于奇异值分解的矩阵性质

本小节，我们将利用矩阵 $A \in \mathbb{R}^{m \times n}$ 的完全奇异值分解

$$A = U\Sigma V^T$$

或紧奇异值分解

$$A = U_r \Sigma_r V_r^T$$

来重新探讨关于矩阵 A 的一些性质：

- 矩阵 A 的秩、零空间和列空间
- 矩阵范数
- 矩阵广义逆
- 正交投影

性质 4.6.4. 设矩阵 $A \in \mathbb{R}^{m \times n}$ ，其奇异值分解为 $A = U\Sigma V^T$ ，则矩阵 A 的秩和对角矩阵 Σ 的秩相等，等于正奇异值 σ_i 的个数 r （包含重复的奇异值）。

同样, 由于在实际中 Σ 上对角元可能很小, 但不为零 (例如由于数值误差), 因此可以在给定的误差 $\epsilon \geq 0$ 的范围内给出一个更加可靠的数值秩:

$$r = \max_{\sigma_k > \epsilon \sigma_1} k$$

性质 4.6.5. 设矩阵 $A \in \mathbb{R}^{m \times n}$ 的紧奇异值分解为 $A = U_r \Sigma_r V_r^T$, 其秩 $\text{rank}(A) = r$, 则有 $\text{Null}(A) = n - r$ 且生成 $\text{Null}(A)$ 的一组正交基底由 V 的最后 $n - r$ 列给出, 也即

$$\text{Null}(A) = \text{Col}(V_{nr}), \quad V_{nr} = (\mathbf{v}_{r+1}, \dots, \mathbf{v}_n)$$

证明. 根据线性代数的基本定理, 有 $\text{Null}(A) = n - r$. 因为 $V = (V_r V_{nr})$ 是正交矩阵, 所以 $\{\mathbf{v}_{r+1}, \dots, \mathbf{v}_n\}$ 是正交向量组, 并且 $V_r^T V_{nr} = 0$. 因此对于 V_{nr} 列空间中任意的向量 $\eta = V_{nr} z$, 由矩阵 $A \in \mathbb{R}^{m \times n}$ 的紧奇异值分解有

$$A\eta = U_r \Sigma_r V_r^T \eta = U_r \Sigma_r V_r^T V_{nr} z = 0$$

所以

$$\text{Null}(A) = \text{Col}(V_{nr})$$

□

性质 4.6.6. 设矩阵 $A \in \mathbb{R}^{m \times n}$ 的紧奇异值分解为 $A = U_r \Sigma_r V_r^T$, 其秩 $\text{rank}(A) = r$, 则 A 的列空间由 U 的前 r 个列向量生成, 即

$$\text{Col}(A) = \text{Col}(U_r), \quad U_r = (\mathbf{u}_1, \dots, \mathbf{u}_r)$$

证明. 首先, 因为 $\Sigma_r V_r^T \in \mathbb{R}^{r \times n} \square r \leq n$, 是一个行满秩矩阵, 则当 x 张成整个 \mathbb{R}^n 时, $z = \Sigma_r V_r^T x$ 张成整个 \mathbb{R}^r . 因此

$$\begin{aligned} \text{Col}(A) &= \{y | y = Ax, x \in \mathbb{R}^n\} \\ &= \{y | y = U_r \Sigma_r V_r^T x, x \in \mathbb{R}^n\} \\ &= \{y | y = U_r z, z \in \mathbb{R}^r\} \\ &= \text{Col}(U_r) \end{aligned}$$

□

例 4.6.6. 矩阵 $A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$ 的奇异值分解为

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & \sqrt{0.2} & -\sqrt{0.8} & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \sqrt{0.8} & \sqrt{0.2} & 0 \end{pmatrix} \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & \sqrt{5} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

草稿
请勿外传

$$\text{所以 } \text{Col}(\mathbf{A}) = \text{Col} \begin{pmatrix} 0 & 0 & \sqrt{0.2} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sqrt{0.8} \end{pmatrix}, \text{Null}(\mathbf{A}) = \text{Col} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

矩阵范数

- 矩阵的 F 范数满足以下等式

$$\|\mathbf{A}\|_F^2 = \text{Tr}(\mathbf{A}^T \mathbf{A}) = \sum_{i=1}^n \lambda_i(\mathbf{A}^T \mathbf{A}) = \sum_{i=1}^n \sigma_i^2$$

其中 σ_i 是矩阵 \mathbf{A} 的奇异值。因此 F 范数的平方实际上就是奇异值的平方和。

- 矩阵 2 范数的平方是 $\mathbf{A}^T \mathbf{A}$ 的最大特征值，所以

$$\|\mathbf{A}\|_2^2 = \sigma_1^2$$

即 \mathbf{A} 的 2 范数就是 \mathbf{A} 的最大的奇异值

- 对于矩阵核范数，我们有

$$\|\mathbf{A}\|_* = \sum_{i=1}^r \sigma_i, r = \text{rank}(\mathbf{A})$$

核范数常常出现在低秩矩阵补全和秩最小化问题中，这是因为 $\|\mathbf{A}\|_*$ 表示在谱范数以 1 为界的矩阵集合中 $\text{rank}(\mathbf{A})$ 的最大可能凸下界。

定义 4.6.4. 令 \mathbf{A} 是一个 $m \times n$ 矩阵，若存在一个的 $n \times m$ 矩阵 \mathbf{G} ，使得下列条件满足：

$$(\mathbf{AG})^T = \mathbf{AG}$$

$$(\mathbf{GA})^T = \mathbf{GA}$$

$$\mathbf{GAG} = \mathbf{G}$$

$$\mathbf{AGA} = \mathbf{A}$$

则称 \mathbf{G} 是 \mathbf{A} 的广义逆或 Moore-Penrose 逆或伪逆。

我们还可以定义其他广义逆，比如在上面四条中去掉一条到三条就可以定义另外 14 种广义逆。但是只有 Moore-Penrose 逆有下列性质。

性质 4.6.7. 设矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ ，如果 \mathbf{G} 是 \mathbf{A} 的 Moore-Penrose 逆，那么 \mathbf{G} 是 \mathbf{A} 唯一的 Moore-Penrose 逆。

在后面的内容中，我们不关心其他的广义逆，所以默认这里的广义逆均指的是 Moore-Penrose 逆。

利用奇异值分解求解广义逆 若矩阵 \mathbf{M} 的奇异值分解为 $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^T$, 那么 \mathbf{M} 的伪逆为

$$\mathbf{M}^\dagger = \mathbf{V}\Sigma^\dagger\mathbf{U}^T$$

其中 Σ^\dagger 是 Σ 的伪逆, 是将 Σ 主对角线上每个非零元素都求倒数之后再转置得到的。

例 4.6.7. 求矩阵 $A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$ 的广义逆

解. 首先我们对 A 进行奇异值分解得

$$A = \mathbf{U}\Sigma\mathbf{V}^T = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{6}} & 0 & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

根据公式我们有

$$A^\dagger = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} -1 & 1 & 2 \\ 2 & 1 & -1 \end{bmatrix}$$

对于一些特殊的矩阵的逆, 我们可以使用奇异值分解推导出更便于计算的公式。

性质 4.6.8. 如果 $A \in \mathbb{R}^{n \times n}$ 可逆, 那么

$$A^\dagger = A^{-1}$$

例 4.6.8. 矩阵 $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ 的广义逆矩阵为

$$A^\dagger = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$$

容易验证

$$AA^\dagger = A^\dagger A = I$$

性质 4.6.9. 当矩阵 $A \in \mathbb{R}^{m \times n}$ 为列满秩矩阵时有

$$A^\dagger = (A^T A)^{-1} A^T$$

证明. 如果 $A \in \mathbb{R}^{m \times n}$ 是一个列满秩矩阵, 因此 $r = n \leq m$, 故有

$$A^\dagger A = V_r V_r^T = I_n$$

所以 A^\dagger 是矩阵 A 的左逆 (即 $A^\dagger A = I_n$)。注意到 $A^T A$ 是可逆的, 所以

$$(A^T A)^{-1} A^T = (V \Sigma^{-2} V^T) V \Sigma^T U^T = V \Sigma^{-1} U^T = A^\dagger$$

□

\mathbf{A} 所有的左逆都可以表示为 $\mathbf{A}^{li} = \mathbf{A}^\dagger + \mathbf{Q}^T$ 其中 \mathbf{Q} 满足 $\mathbf{A}^T \mathbf{Q} = 0$

性质 4.6.10. 当矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 为行满秩矩阵时有

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

证明. 如果 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 是一个行满秩矩阵, 因此 $r = m \leq n$, 故有

$$\mathbf{A}\mathbf{A}^\dagger = \mathbf{U}_r \mathbf{U}_r^T = \mathbf{I}_m$$

所以 \mathbf{A}^\dagger 是矩阵 \mathbf{A} 的右逆 (即 $\mathbf{A}\mathbf{A}^\dagger = \mathbf{I}_m$)。注意到 $\mathbf{A}\mathbf{A}^T$ 是可逆的, 所以

$$\mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} = \mathbf{V}\Sigma^T \mathbf{U}^T (\mathbf{U}\Sigma^{-2}\mathbf{U}^T) = \mathbf{V}\Sigma^{-1}\mathbf{U}^T = \mathbf{A}^\dagger$$

□

\mathbf{A} 所有的右逆都可以表示为 $\mathbf{A}^{ri} = \mathbf{A}^\dagger + \mathbf{Q}$ 其中 \mathbf{Q} 满足 $\mathbf{A}\mathbf{Q} = 0$ 。

例 4.6.9. 求矩阵 $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$ 的广义逆

解. 显然这是一个列满秩的矩阵, 我们利用列满秩矩阵的公式

$$\begin{aligned} \mathbf{A}^\dagger &= (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T = \left(\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \\ &= \frac{1}{3} \begin{bmatrix} -1 & 1 & 2 \\ 2 & 1 & -1 \end{bmatrix} \end{aligned}$$

正交投影 我们知道任何一个矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 是一个从输入空间 \mathbb{R}^n 到输出空间 \mathbb{R}^m 的线性映射, 并且根据线性代数基本定理, 我们可以将 $\mathbb{R}^n, \mathbb{R}^m$ 分解成如下正交子空间的直和:

$$\mathbb{R}^n = \text{Null}(\mathbf{A}) \oplus \text{Null}(\mathbf{A})^\perp = \text{Null}(\mathbf{A}) \oplus \text{Col}(\mathbf{A}^T)$$

$$\mathbb{R}^m = \text{Col}(\mathbf{A}) \oplus \text{Col}(\mathbf{A})^\perp = \text{Col}(\mathbf{A}) \oplus \text{Null}(\mathbf{A}^T)$$

正如前面讨论的, 矩阵 \mathbf{A} 的奇异值分解 $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ 给四个基本子空间提供了正交基底, 我们令

$$\mathbf{U} = (\mathbf{U}_r, \mathbf{U}_{nr}), \mathbf{V} = (\mathbf{V}_r, \mathbf{V}_{nr})$$

其中 $r = \text{rank}(\mathbf{A})$, 我们就有

$$\text{Null}(\mathbf{A}) = \text{Col}(\mathbf{V}_{nr}), \text{Col}(\mathbf{A}^T) = \text{Col}(\mathbf{V}_r)$$

$$\text{Col}(\mathbf{A}) = \text{Col}(\mathbf{U}_r), \text{Null}(\mathbf{A}^T) = \text{Col}(\mathbf{U}_{nr})$$

接下来, 我们讨论如何将一个向量 $\mathbf{x} \in \mathbb{R}^n$ 投影到 $\text{Null}(\mathbf{A}), \text{Col}(\mathbf{A}^T)$ 中, 以及把一个向量 $\mathbf{y} \in \mathbb{R}^m$ 投影到 $\text{Null}(\mathbf{A}^T), \text{Col}(\mathbf{A})$ 中。

如果给定一个向量 $\mathbf{x} \in \mathbb{R}^n$ 和 d 个线性无关的向量 $\mathbf{b}_1, \dots, \mathbf{b}_d \in \mathbb{R}^n$, 那么 \mathbf{x} 到子空间 $\text{span}\{\mathbf{b}_1, \dots, \mathbf{b}_d\}$ 的正交投影就是向量

$$\mathbf{x}^* = \mathbf{B}\boldsymbol{\alpha}$$

其中 $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_d), \boldsymbol{\alpha} \in \mathbb{R}^d$, 并且我们需要解方程

$$\mathbf{B}^T \mathbf{B} \boldsymbol{\alpha} = \mathbf{B}^T \mathbf{x}$$

来得到 $\boldsymbol{\alpha}$ 。注意到, 如果 \mathbf{B} 的列向量是正交的, 则有 $\mathbf{B}^T \mathbf{B} = \mathbf{I}_d$, 因此

$$\boldsymbol{\alpha} = \mathbf{B}^T \mathbf{x}$$

故可得投影

$$\mathbf{x}^* = \mathbf{B} \mathbf{B}^T \mathbf{x}$$

因此如果我们将一个向量 $\mathbf{x} \in \mathbb{R}^n$ 投影到 $\text{Null}(\mathbf{A})$ 上, 投影 $\pi_{\text{Null}(\mathbf{A})}(\mathbf{x})$ 则可以通过以下等式算出

$$\pi_{\text{Null}(\mathbf{A})}(\mathbf{x}) = (\mathbf{V}_{nr} \mathbf{V}_{nr}^T) \mathbf{x}$$

我们又知道

$$\mathbf{I} = \mathbf{V} \mathbf{V}^T = \mathbf{V}_r \mathbf{V}_r^T + \mathbf{V}_{nr} \mathbf{V}_{nr}^T$$

因此由广义逆的定义, 我们可知投影矩阵

$$\mathbf{P}_{\text{Null}(\mathbf{A})} = (\mathbf{V}_{nr} \mathbf{V}_{nr}^T) = \mathbf{I}_n - \mathbf{V}_r \mathbf{V}_r^T = \mathbf{I}_n - \mathbf{A}^\dagger \mathbf{A}$$

在 \mathbf{A} 行满秩的情况下, 由性质 10 可知 $\mathbf{A}^\dagger = \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1}$, 所以有

$$\mathbf{P}_{\text{Null}(\mathbf{A})} = \mathbf{I}_n - \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{A}$$

矩阵 $\mathbf{P}_{\text{Null}(\mathbf{A})}$ 称为子空间 $\text{Null}(\mathbf{A})$ 上的正交投影。

用同样的方式, 我们可以得到 \mathbf{x} 在 $\text{Col}(\mathbf{A}^T)$ 上的投影为 $\pi_{\text{Col}(\mathbf{A}^T)}(\mathbf{x}) = (\mathbf{V}_r \mathbf{V}_r^T) \mathbf{x} = \mathbf{A}^\dagger \mathbf{A} \mathbf{x}$

而当 \mathbf{A} 行满秩的时, 有 $\pi_{\text{Col}(\mathbf{A}^T)}(\mathbf{x}) = \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{A} \mathbf{x}$.

类似的, 我们可以得到 \mathbf{y} 在 $\text{Col}(\mathbf{A})$ 上的投影为 $\pi_{\text{Col}(\mathbf{A})}(\mathbf{y}) = (\mathbf{U}_r \mathbf{U}_r^T) \mathbf{y} = \mathbf{A} \mathbf{A}^\dagger \mathbf{y}$.

而当 \mathbf{A} 列满秩的时, 有 $\pi_{\text{Col}(\mathbf{A})}(\mathbf{y}) = \mathbf{A} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}$.

向量 \mathbf{y} 在 $\text{Null}(\mathbf{A}^T)$ 上的投影为 $\pi_{\text{Null}(\mathbf{A}^T)}(\mathbf{y}) = (\mathbf{U}_{nr} \mathbf{U}_{nr}^T) \mathbf{y} = (\mathbf{I}_m - \mathbf{A} \mathbf{A}^\dagger) \mathbf{y}$.

并且当 \mathbf{A} 列满秩时, 有 $\pi_{\text{Null}(\mathbf{A}^T)}(\mathbf{y}) = (\mathbf{I}_m - \mathbf{A} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T) \mathbf{y}$.

4.6.9 奇异值分解与低秩表示

考虑通过奇异值分解把矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 表示为一些更低秩矩阵的和。

- 假设矩阵 \mathbf{A} 的奇异值分解为 $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$, 其中 $\mathbf{U} \in \mathbb{R}^{m \times n}$ 和 $\mathbf{V} \in \mathbb{R}^{n \times n}$ 都是正交矩阵, $\Sigma \in \mathbb{R}^{n \times n}$ 是对角矩阵。我们把 \mathbf{A} 的奇异值分解看成矩阵 $\mathbf{U}\Sigma$ 和 \mathbf{V}^T 的乘积, 将 $\mathbf{U}\Sigma$ 按列分块, 将 \mathbf{V}^T 按行分块, 即

$$\mathbf{U}\Sigma = (\sigma_1 \mathbf{u}_1, \sigma_2 \mathbf{u}_2, \dots, \sigma_n \mathbf{u}_n); \mathbf{V}^T = \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_n^T \end{pmatrix}$$

则有

$$\mathbf{A} = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \dots + \sigma_n \mathbf{u}_n \mathbf{v}_n^T = \sum_{i=1}^n \sigma_i \mathbf{A}_i$$

称该式子为矩阵 \mathbf{A} 的外积展开式, 其中 $\mathbf{A}_i = \mathbf{u}_i \mathbf{v}_i^T$ 为 $m \times n$ 的秩 1 矩阵, 是列向量 \mathbf{u}_i 和行向量 \mathbf{v}_i^T 的外积, 其第 k 行第 j 列元素为 \mathbf{u}_i 的第 k 个元素与 \mathbf{v}_i^T 的第 j 个元素的乘积。

- 如果矩阵 \mathbf{A} 的秩为 r , 则对于任意 $i > r$ 的项, 因为奇异值为 0, 所以可以将该矩阵分解为 r 个秩为 1 矩阵 \mathbf{A}_i 之和:

$$\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{A}_i$$

其中外积矩阵 \mathbf{A}_i 前面的系数是矩阵 \mathbf{A} 第 i 个非零奇异值 σ_i 。

- 更进一步, 如果把上述 \mathbf{A}_i 从 1 到 r 求和替换成从 1 到 k ($k < r$) 求和, 则我们可以获得矩阵 \mathbf{A} 的近似

$$\hat{\mathbf{A}}(k) = \sum_{i=1}^k \sigma_i \mathbf{A}_i$$

其中 $\text{rank}(\hat{\mathbf{A}}) = k$, 称为矩阵 \mathbf{A} 的秩 k 近似。

那么如何度量矩阵 \mathbf{A} 和它的秩 k 近似 $\hat{\mathbf{A}}(k)$ 之间的差异大小或近似程度呢? 能否找到矩阵 \mathbf{A} 的一个最优低秩矩阵近似呢?

低秩矩阵近似 给定一个秩为 r 的矩阵 \mathbf{A} , 欲求其最优的秩 k 近似矩阵 $\hat{\mathbf{A}}(k)$, 其中 $k \leq r$, 该问题可形式化为求

$$\begin{aligned} \min_{\hat{\mathbf{A}}(k) \in \mathbb{R}^{m,n}} \|\mathbf{A} - \hat{\mathbf{A}}(k)\|_F, \\ \text{s.t.: } \text{rank}(\hat{\mathbf{A}}(k)) = k \end{aligned}$$

定理 4.6.2. 设矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$, 矩阵的秩 $\text{rank}(\mathbf{A}) = r$, 并设 \mathbb{M} 为 $\mathbb{R}^{m \times n}$ 中所有秩不超过 k 的矩阵集合 $0 < k < r$, 则存在一个秩为 k 的矩阵 $\mathbf{X} \in \mathbb{M}$, 使得

$$\|\mathbf{A} - \mathbf{X}\|_F = \min_{\mathbf{S} \in \mathbb{M}} \|\mathbf{A} - \mathbf{S}\|_F$$

称矩阵 \mathbf{X} 为矩阵 \mathbf{A} 在 F 范数下的最优近似。

定理 4.6.3. 设矩阵 $A \in \mathbb{R}^{m \times n}$, 矩阵的秩 $\text{rank}(A) = r$, 有奇异值分解 $A = U\Sigma V^T$, 并设 \mathbb{M} 为 $\mathbb{R}^{m \times n}$ 中所有秩不超过 k 的矩阵集合 $0 < k < r$, 若秩为 k 的矩阵 $X \in \mathbb{M}$, 满足

$$\|A - X\|_F = \min_{S \in \mathbb{M}} \|A - S\|_F$$

则 $\|A - X\|_F = (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{\frac{1}{2}}$ 。特别地, 若 $A' = U\Sigma'V^T$, 其中

$$\Sigma' = \begin{pmatrix} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_k & \\ & & & 0 \\ 0 & & & & \ddots \\ & & & & & 0 \end{pmatrix} = \begin{pmatrix} \Sigma_k & 0 \\ 0 & 0 \end{pmatrix}$$

则 $\|A - A'\|_F = (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{\frac{1}{2}} = \min_{S \in \mathbb{M}} \|A - S\|_F$ 。

证明. 若秩为 k 的矩阵 $X \in \mathbb{M}$, 满足 $\|A - X\|_F = \min_{S \in \mathbb{M}} \|A - S\|_F$ 则

$$\|A - X\|_F \leq \|A - A'\|_F = (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{\frac{1}{2}}$$

下面证明 $\|A - X\|_F \geq (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{\frac{1}{2}}$ 。设 X 的奇异值分解为 $Q\Omega P^T$, 其中

$$\Omega = \begin{pmatrix} \omega_1 & & & \\ & \ddots & & 0 \\ & & \omega_k & \\ & & & 0 \\ 0 & & & & \ddots \\ & & & & & 0 \end{pmatrix} = \begin{pmatrix} \Omega_k & 0 \\ 0 & 0 \end{pmatrix}$$

若令矩阵 $B = Q^T AP$, 则 $A = QBP^T$, 由此得到 $\|A - X\|_F = \|Q(B - \Omega)P^T\|_F = \|B - \Omega\|_F$ 。

用 Ω 分块方法对 B 分块

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

其中 $B_{11} \in \mathbb{R}^{k \times k}$, $B_{12} \in \mathbb{R}^{k \times (n-k)}$, $B_{21} \in \mathbb{R}^{(n-k) \times k}$, $B_{22} \in \mathbb{R}^{(n-k) \times (n-k)}$ 可得

$$\|A - X\|_F^2 = \|B - \Omega\|_F^2 = \|B_{11} - \Omega_k\|_F^2 + \|B_{12}\|_F^2 + \|B_{21}\|_F^2 + \|B_{22}\|_F^2$$

现证 $B_{12} = 0$, $B_{21} = 0$ 用反证法。若 $B_{12} \neq 0$, 令

$$Y = Q \begin{pmatrix} B_{11} & B_{12} \\ 0 & 0 \end{pmatrix} P^T$$

则 $Y \in \mathbb{M}$ 且 $\|A - Y\|_F^2 = \|B_{21}\|_F^2 + \|B_{22}\|_F^2 < \|A - X\|_F^2$, 这与 X 的定义 $\|A - X\|_F = \min_{S \in \mathbb{M}} \|A - S\|_F$ 矛盾, 因此 $B_{12} = 0$ 。

同理可证 $\mathbf{B}_{21} = 0$ 。于是

$$\|\mathbf{A} - \mathbf{X}\|_F^2 = \|\mathbf{B}_{11} - \boldsymbol{\Omega}_k\|_F^2 + \|\mathbf{B}_{22}\|_F^2$$

再证 $\mathbf{B}_{11} = \boldsymbol{\Omega}_k$, 为此令

$$\mathbf{Z} = \mathbf{Q} \begin{pmatrix} \mathbf{B}_{11} & 0 \\ 0 & 0 \end{pmatrix} \mathbf{P}^T$$

则 $\mathbf{Z} \in \mathbb{M}$, 且

$$\|\mathbf{A} - \mathbf{Z}\|_F^2 = \|\mathbf{B}_{22}\|_F^2 \leq \|\mathbf{B}_{11} - \boldsymbol{\Omega}_k\|_F^2 + \|\mathbf{B}_{22}\|_F^2 = \|\mathbf{A} - \mathbf{X}\|_F^2$$

由 \mathbf{X} 的定义

$$\|\mathbf{A} - \mathbf{X}\|_F = \min_{\mathbf{S} \in \mathbb{M}} \|\mathbf{A} - \mathbf{S}\|_F$$

知

$$\|\mathbf{B}_{11} - \boldsymbol{\Omega}_k\|_F^2 = 0$$

即 $\mathbf{B}_{11} = \boldsymbol{\Omega}_k$ 。

最后看 \mathbf{B}_{22} , 设 \mathbf{B}_{22} 有奇异值分解为 $\mathbf{U}_1 \boldsymbol{\Lambda} \mathbf{V}_1^T$, 则 $\|\mathbf{A} - \mathbf{X}\|_F = \|\mathbf{B}_{22}\|_F = \|\boldsymbol{\Lambda}\|_F$ 。下面证明 $\boldsymbol{\Lambda}$ 的对角线元素为 \mathbf{A} 的奇异值。为此令

$$\mathbf{U}_2 = \begin{pmatrix} \mathbf{I}_k & 0 \\ 0 & \mathbf{U}_1 \end{pmatrix}, \mathbf{V}_2 = \begin{pmatrix} \mathbf{I}_k & 0 \\ 0 & \mathbf{V}_1 \end{pmatrix}$$

其中 \mathbf{I}_k 是 k 阶单位矩阵, $\mathbf{U}_2, \mathbf{V}_2$ 的分块与 \mathbf{B} 的分块一致。注意到 \mathbf{B} 以及 \mathbf{B}_{22} 的奇异值分解, 即得

$$\mathbf{U}_2^T \mathbf{Q}^T \mathbf{A} \mathbf{P} \mathbf{V}_2 = \begin{pmatrix} \boldsymbol{\Omega}_k & \\ & \boldsymbol{\Lambda} \end{pmatrix}, \mathbf{A} = (\mathbf{Q} \mathbf{U}_2) \begin{pmatrix} \boldsymbol{\Omega}_k & \\ & \boldsymbol{\Lambda} \end{pmatrix} (\mathbf{P} \mathbf{V}_2)^T$$

由此可知 $\boldsymbol{\Lambda}$ 的对角线元素为 \mathbf{A} 的奇异值故有

$$\|\mathbf{A} - \mathbf{X}\|_F = \|\boldsymbol{\Lambda}\|_F \geq (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{\frac{1}{2}}$$

可证 $\|\mathbf{A} - \mathbf{X}\|_F = (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{\frac{1}{2}} = \|\mathbf{A} - \mathbf{A}'\|_F$

□

- 在秩不超过 k 的 $m \times n$ 矩阵的集合中, 存在矩阵 \mathbf{A} 的 F 范数意义下的最优近似矩阵 \mathbf{X}
- $\mathbf{A}' = \mathbf{U} \boldsymbol{\Sigma}' \mathbf{V}^T$ 是达到最优值的一个矩阵
- 紧奇异值分解是在 F 范数意义下的无损压缩
- 截断奇异值分解是有损压缩
- 截断奇异值分解得到的矩阵的秩为 k , 通常远小于原始矩阵的秩 r , 所以是由低秩矩阵实现了对原始矩阵的压缩。

定理4.6.3中若把 F 范数改为谱范数, 则有

$$\|\mathbf{A} - \mathbf{X}\|_2 = \sigma_{k+1} = \min_{\mathbf{S} \in \mathbb{M}} \|\mathbf{A} - \mathbf{S}\|_2,$$

成立。定理4.6.3也被称为 Eckhart-Young 或 Eckhart-Young-Mirsky 定理。

例 4.6.10. 求例 4.6.1 中矩阵 $A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$ 秩为 2 的最优近似。

解. 先对 A 进行奇异值分解得

$$\begin{pmatrix} 0 & 0 & -\frac{\sqrt{5}}{5} & 0 & -\frac{2\sqrt{5}}{5} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{2\sqrt{5}}{5} & 0 & \frac{\sqrt{5}}{5} \end{pmatrix} \begin{pmatrix} 4 & & & \\ & 3 & & \\ & & \sqrt{5} & \\ & & & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

然后令

$$A' = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 4 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

即为矩阵 A 秩 2 的最优近似。

基于奇异值分解的图像压缩 假定一幅图像有 $m \times n$ 个像素, 如果将这 mn 个数据一起传送, 往往会显得数据量太大。因此, 我们希望能够改为传送另外一些比较少的数据, 并且在接收端还能够利用这些传送的数据重构原图像。用 $m \times n$ 矩阵 A 表示要传送的原 $m \times n$ 个像素。

假定对矩阵 A 进行奇异值分解, 便得到 $A = U\Sigma V^T$, 其中, 奇异值按照从大到小的顺序排列。如果从中选择 k 个大奇异值以及与这些奇异值对应的左和右奇异向量逼近原图像, 便可以其使用 $k(n+m+1)$ 个数值代替原来的 $m \times n$ 个图像数据。这 $k(n+m+1)$ 个被选择的新数据是矩阵 A 的前 k 个奇异值、 $m \times m$ 左奇异向量矩阵 U 的前 k 列和 $n \times n$ 右奇异向量矩阵 V 的前 k 列的元素。

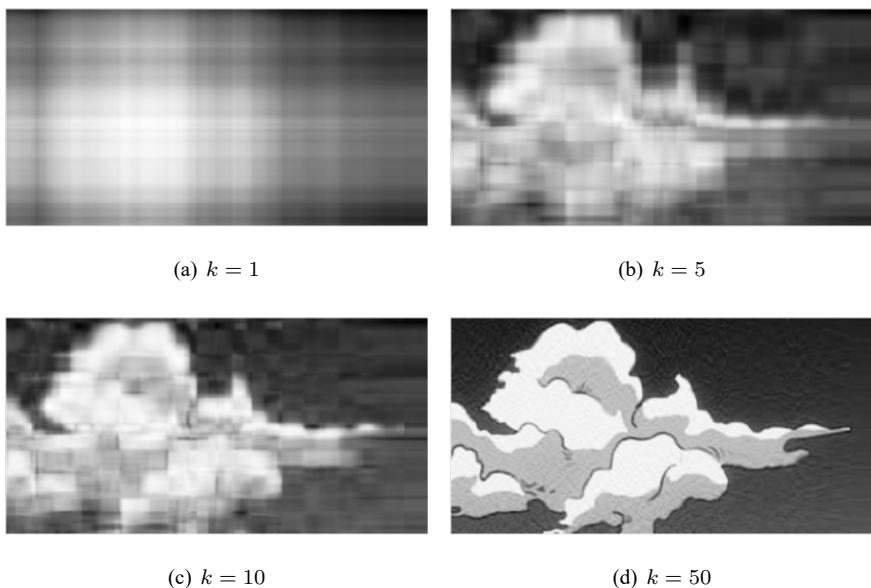
把比率

$$\rho = \frac{nm}{k(n+m+1)} \tag{4.16}$$

称为图像的压缩比。显然, 被选择的大奇异值的个数 k 应该满足条件 $k(n+m+1) < nm$, 即 $k < \frac{nm}{n+m+1}$ 。

右边四张图在视觉上展示了取不同数量的奇异值的效果:

- 当 $k = 5$ 时, 我们已经可以看出图像上是什么了。

图 4.16: 不同 k 值对于压缩图像的影响

- 当 $k = 10$ 时, 我们获得了更多的细节。但是仍然有一些模糊。
- 当 $k = 50$ 时, 我们获得了一个相当不错的图像, 只有非常细微的地方有一些模糊。整体上和原图相差无几。

原图是一张 1328×680 的图像, 要传输这样一张图像需要发送 $1328 \times 680 = 903040$ 个数值。而如果使用 $k = 50$ 时的截断 SVD, 那么只需要传送 $50 \times (1328 + 680 + 1) = 100450$ 个数值。也就是说压缩比达到了 8.9899。

因此, 我们在传送图像的过程中, 就无须传送 $m \times n$ 个原始数据, 而只需要传送 $k(n+m+1)$ 个有关奇异值和奇异向量的数据即可。在接收端, 在接收到奇异值 $\sigma_1, \sigma_2, \dots, \sigma_k$ 以及左奇异向量 $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$ 和右奇异向量 $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ 后, 即可通过截尾的奇异值分解公式

$$\hat{\mathbf{A}} = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad (4.17)$$

重构出原图像。

一个容易理解的事实是: 若 k 值偏小, 即压缩比 p 偏大, 则重构的图像的质量有可能不能令人满意。反之, 过大的 k 值又会导致压缩比过小, 从而降低图像压缩和传送的效率。因此, 需要根据不同种类的图像, 选择合适的压缩比, 以兼顾图像传送效率和重构质量。

本节介绍了各种奇异值分解的形式, 包括完全奇异值分解, 紧奇异值分解, 截断奇异值分解等。也介绍了矩阵性质与奇异值分解的关系。如矩阵范数, 矩阵的广义逆, 最优低秩矩阵近

似等。其中，矩阵的低秩近似问题实际上是一个优化问题。它表明数据科学与机器学习中某些优化问题可以方便的通过奇异值分解进行求解，这些优化问题还包括 PCA、正交的 Procrustean 变换等！

4.7 阅读材料

本章介绍了五种常用的矩阵分解方法，包括 LU（三角）分解、QR（正交）三角分解、谱（特征）分解、Cholesky 分解和奇异值分解等。线性代数包含很多有趣的矩阵，如：对角阵、三角矩阵、正交矩阵、对称矩阵、置换矩阵、投影矩阵和关联矩阵等等。在这些矩阵当中对称正定矩阵是核心，因为数据科学与机器学习中大部分矩阵都是非方阵，而非方阵总是可以通过与其自身的转置相乘得到对称正（半）定矩阵。对称正（半）定矩阵有正（非负）的特征值，并且有正交的特征向量，它也可以表示成一些秩 1 矩阵的线性组合，因此可以方便的用于做低秩近似计算。在机器学习中，我们主要处理的是这些大规模的对称正定矩阵或复杂的非方阵矩阵，需要借助矩阵分解的技术，特别是奇异值分解，把它表示为对角阵、三角阵和正交矩阵的乘积等等，然后利用这些特殊的简单的矩阵实现复杂矩阵的特征值等矩阵基本特征的快速计算，并用于数据压缩、数据降维，矩阵低秩近似问题的求解等等，这对帮助理解原本复杂的高维数据矩阵的结构和性质具有重要的作用。

例如，当我们必须计算或模拟随机事件时，经常会用到基于 Cholesky 分解的矩阵分解 (Rubinstein 和 Kroese, 2016)。关于稀疏的 Cholesky 因式分解包含于 (George 和 Liu, 1981), (Duff, Erisman 和 Reid, 2017)，这些文献中讨论了稀疏的 LU 和 LDL^T 因式分解。特征分解是使我们能够提取表征线性映射的有意义且可解释的信息的基础。因此，特征分解是称为谱方法的一类机器学习算法的基础，这些算法中通常会进行正定核的特征分解。基于特征分解的统计数据分析中的经典方法包括：主成分分析 (PCA (Pearson, 1901a))，其中寻找解释数据中内蕴不变结构的低维子空间；Fisher 判别分析，旨在确定用于数据分类的分离超平面 (Mika 等, 1999)；多维尺度分析 (MDS) (Carroll 和 Chang, 1970)。这些方法的计算通常是通过找到对称的半正定矩阵的最佳秩 k 近似得到的。SVD 允许我们发现一些与特征分解相同的信息。然而 SVD 更一般地适用于非方形矩阵，例如当我们想要对数据做压缩时，只要我们想要识别数据中的异质性，基于 SVD 的矩阵因子分解方法就变得非常相关。由于计算效率的原因，SVD 低秩近似经常用于机器学习，这是因为它减少了我们需要对非常大的数据矩阵执行的非零乘法的存储和操作 (Trefethen 和 Bau III, 1997)。此外，低秩近似还可用于对可能包含缺失值的矩阵进行处理，以及用于有损压缩和降维等 (Moonen 和 De Moor, 1995; Markovsky, 2011)。

习题

习题 4.1. 判定矩阵 $C = \begin{bmatrix} 3 & 2 & -1 \\ -1 & 0 & 0 \\ -1 & 3 & 0 \end{bmatrix}$ 和 $B = \begin{bmatrix} 0 & 2 & -1 \\ -1 & 4 & -1 \\ 1 & 3 & -5 \end{bmatrix}$ 能否进行 LU 分解, 为什么?

如果能分解, 试分解之。

习题 4.2. 对下列矩阵进行 LU 分解:

$$(1) A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 2 & 2 \end{bmatrix}; (2) B = \begin{bmatrix} 12 & -3 & 3 \\ -18 & 3 & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

习题 4.3. 求矩阵 $A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ 的 LU 分解。

习题 4.4. 求对称正定矩阵

$$A = \begin{bmatrix} 5 & 2 & -4 \\ 2 & 1 & -2 \\ -4 & -2 & 5 \end{bmatrix}$$

的不带平方根的 Cholesky 分解。

习题 4.5. 对 A 进行 LU 分解

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

习题 4.6. 对 A 进行 Cholesky 分解

$$\begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix}$$

习题 4.7. 求下列矩阵的正交三角分解 (UR) 表达式:

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

传
外
勿
清
稿
草

习题 4.8. 求矩阵

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

的 QR 分解。

习题 4.9. 求矩阵 $\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ 的奇异值分解。

习题 4.10. 求 $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$ 的奇异值分解。

习题 4.11. 设 $\mathbf{A} = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \end{bmatrix}$, 求 \mathbf{A} 的奇异值分解。

习题 4.12. 已知

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

求 \mathbf{A} 的奇异分解表达式。

习题 4.13. 已知 $\mathbf{A} \in \mathcal{C}_r^{m \times n}$ (秩为 $r > 0$) 的奇异值分解表达式为

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \Delta & 0 \\ 0 & 0 \end{bmatrix} \mathbf{V}^H$$

试求矩阵 $\mathbf{B} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A} \end{bmatrix}$ 的奇异值分解表达式。

习题 4.14. 已知矩阵

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 4 \\ \frac{1}{2} & 0 & 2 \\ \frac{1}{4} & \frac{1}{2} & 0 \end{bmatrix}$$

验证 \mathbf{A} 是可对角化矩阵，并求 \mathbf{A} 的谱分解表达式。

习题 4.15. 在对 PCA 是最佳的 d -维仿射变化拟合时,

$$\begin{aligned}\nabla_{\boldsymbol{\mu}} \sum_{i=1}^n \|\mathbf{x}_i - (\boldsymbol{\mu}_n + \mathbf{V}\boldsymbol{\beta}_i)\|_2^2 = 0 &\Leftrightarrow \sum_{i=1}^n (\mathbf{x}_i - (\boldsymbol{\mu} + \mathbf{V}\boldsymbol{\beta}_i)) = 0 \\ &\Leftrightarrow \left(\sum_{i=1}^n \mathbf{x}_i\right) - n\boldsymbol{\mu} - \mathbf{V}\left(\sum_{i=1}^n \boldsymbol{\beta}_i\right) = 0\end{aligned}$$

有不失一般性假设 $\sum_{i=1}^n \boldsymbol{\beta}_i = \mathbf{0}$ 。

证明, 对任意的 \mathbf{b} , 假设 $\sum_{i=1}^n \boldsymbol{\beta}_i = \mathbf{b}$ 。最终得到 \mathbf{x}_i 的拟合值 $\boldsymbol{\mu} + \mathbf{V}\boldsymbol{\beta}_i$ 是相等的。

习题 4.16. 由 $\|\mathbf{x}\|_2^2 = \langle \mathbf{x}, \mathbf{x} \rangle$ 和 $\mathbf{V}^T \mathbf{V} = \mathbf{I}$, 证明:

$$\|(\mathbf{x}_i - \boldsymbol{\mu}_n) - \mathbf{V}\mathbf{V}^T(\mathbf{x}_i - \boldsymbol{\mu}_n)\|_2^2 = (\mathbf{x}_i - \boldsymbol{\mu}_n)^T(\mathbf{x}_i - \boldsymbol{\mu}_n) - (\mathbf{x}_i - \boldsymbol{\mu}_n)^T\mathbf{V}\mathbf{V}^T(\mathbf{x}_i - \boldsymbol{\mu}_n)$$

习题 4.17. 利用矩阵迹的性质, 证明:

$$\sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu}_n)^T \mathbf{V} \mathbf{V}^T (\mathbf{x}_i - \boldsymbol{\mu}_n) = (n-1) \operatorname{Tr}(\mathbf{V}^T \boldsymbol{\Sigma}_n \mathbf{V})$$

其中 $\boldsymbol{\Sigma}_n = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu}_i)(\mathbf{x}_i - \boldsymbol{\mu}_i)^T$

参考文献

- [1] Press, William H., Teukolsky, Saul A., Vetterling, William T., and Flannery, Brian P. 2007. Numerical Recipes: The Art of Scientific Computing. third edn. Cambridge University Press.
- [2] Pearson, Karl. 1901a. LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2(11), 559–572.
- [3] Rubinstein, Reuven Y, and Kroese, Dirk P. 2016. Simulation and the Monte Carlo method. Vol. 10. John Wiley & Sons.
- [4] Mika, Sebastian, Ratsch, Gunnar, Weston, Jason, Scholkopf, Bernhard, and Muller, Klaus-Robert. 1999. Fisher discriminant analysis with kernels. Pages 41–48 of: Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop. Ieee.
- [5] Carroll, J Douglas, and Chang, Jih-Jie. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition. Psychometrika, 35(3), 283–319.
- [6] Carroll, J Douglas, and Chang, Jih-Jie. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition. Psychometrika, 35(3), 283–319.
- [7] Kolda, Tamara G, and Bader, Brett W. 2009. Tensor decompositions and applications. SIAM review, 51(3), 455–500.

- [8] Markovsky, Ivan. 2011. Low rank approximation: algorithms, implementation, applications. Springer Science & Business Media.
- [9] Moonen, Marc, and De Moor, Bart. 1995. SVD and Signal Processing, III: Algorithms, Architectures and Applications. Elsevier.
- [10] Ormoneit, Dirk, Sidenbladh, Hedvig, Black, Michael J, and Hastie, Trevor. 2001. Learning and tracking cyclic human motion. Pages 894 – 900 of: Advances in Neural Information Processing Systems.
- [11] Trefethen, Lloyd N, and Bau III, David. 1997. Numerical Linear Algebra. Vol. 50. Siam. Tucker, Ledyard R. 1966. Some mathematical notes on three-mode factor analysis. Psychometrika, 31(3), 279–311.
- [12] Duff I S, Erisman A M, Reid J K. Direct methods for sparse matrices[M]. Oxford University Press, 2017.

传外勿请稿草

第五章 矩阵计算问题

在众多科学与工程学科，如物理、化学工程、统计学、经济学、生物学、信号处理、自动控制、系统理论、医学和军事工程等中，许多问题都可用数学建模成矩阵方程 $\mathbf{Ax} = \mathbf{b}$ 。根据数据向量 $\mathbf{b} \in \mathbb{R}^{m \times 1}$ 和数据矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 的不同，矩阵方程主要有以下三种类型：

(1) **适定方程组**：方程的个数与未知量的个数相等即 $m = n$ ，并且 \mathbf{A} 满秩可逆，此时 \mathbf{x} 有唯一的解。

(2) **超定方程组**：当上述 $m > n$ 时，并且数据矩阵 \mathbf{A} 和数据向量 \mathbf{b} 均已知，其中之一或者二者可能存在误差或者干扰。

(3) **欠定方程组**：当上述 $m < n$ 时，数据矩阵 \mathbf{A} 和数据向量 \mathbf{b} 均已知，但未知向量 \mathbf{x} 可能要求为稀疏向量。

我们这里引进线性方程并给出它的标准形式 $\mathbf{Ax} = \mathbf{y}$ ，其中 $\mathbf{x} \in \mathbb{R}^n$ 是未知变量， $\mathbf{A} \in \mathbb{R}^{m \times n}$ 是参数矩阵， $\mathbf{y} \in \mathbb{R}^m$ 是已知向量。线性方程构成了数值线性代数的基础，它们的解法是许多优化方法的关键。事实上，解线性方程组问题 $\mathbf{Ax} = \mathbf{y}$ 可以被看成优化问题，即关于 \mathbf{x} ，最小化 $\|\mathbf{Ax} - \mathbf{y}\|^2$ 。我们描述线性方程组解得集合并且当线性方程组正确解不存在的情况下，讨论求解线性方程组近似解的方法。随后引出最小二乘问题以及它的变体、解的数值敏感性及其解决方法，它们与矩阵分解的关系（例如 QR 分解和 SVD）也将被介绍。

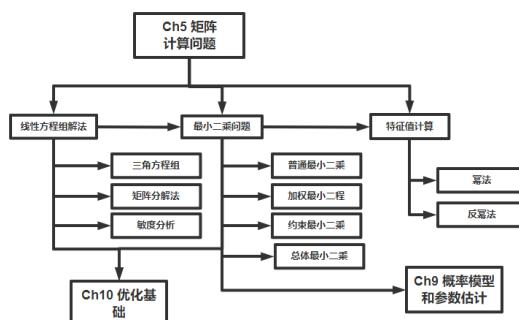


图 5.1: 本章导图

5.1 线性方程组的直接解法

5.1.1 线性方程组问题

在工程问题中，线性方程组描述了变量之间最基本的关系。线性方程在各个科学分支中无处不在，例如弹性力学、电阻网络、曲线拟合等。线性方程构成了线性代数的核心并时常作为优化问题的约束条件。由于许多优化算法的迭代过程非常依赖线性方程组的解，所以它也是许多优化算法的基础。下面我们以一个例子来说明，线性方程组如何解决上面的问题。

例 5.1.1. (三点测距问题) 三角测量是一种确定点位置的方法，给定距离到已知控制点(锚点)，三边测量可以应用于许多不同的领域，如地理测绘、地震学、导航(例如 GPS 系统)等。在图 5.2 中，三个测距点 $a_1, a_2, a_3 \in \mathbb{R}^2$ 的坐标是已知的，并且从点 $x = (x_1, x_2)^T$ 到测距点的距离为 d_1, d_2, d_3 ， x 的未知坐标与距离测量有关，可以由下面非线性方程组描述

$$\|x - a_1\|_2^2 = d_1^2, \quad \|x - a_2\|_2^2 = d_2^2, \quad \|x - a_3\|_2^2 = d_3^2 \quad (5.1)$$

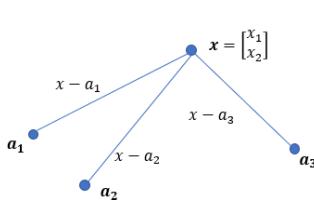


图 5.2: 三点测量位置图例。点 x 处，我们测量距三个测距点 a_1, a_2, a_3 的距离，以便确定 x 的坐标。

通过第一个方程减去另外两个方程，我们获得了两个 x 的线性方程组。

$$2(a_2 - a_1)^T x = d_1^2 - d_2^2 + \|a_2\|_2^2 - \|a_1\|_2^2$$

$$2(a_3 - a_1)^T x = d_1^2 - d_3^2 + \|a_3\|_2^2 - \|a_1\|_2^2$$

也就是说，原始非线性方程组(5.1)的每个解也可以看作线性方程组的解。使用方程组标准形式 $Ax = y$ (标准形式的定义在下一小节给出)可以描述为：

$$A = \begin{bmatrix} 2(a_2 - a_1)^T \\ 2(a_3 - a_1)^T \end{bmatrix}, \quad y = \begin{bmatrix} d_1^2 - d_2^2 + \|a_2\|_2^2 - \|a_1\|_2^2 \\ d_1^2 - d_3^2 + \|a_3\|_2^2 - \|a_1\|_2^2 \end{bmatrix} \quad (5.2)$$

上述问题的解，将在后面详细讨论。

我们先回顾线性方程组中的一般概念。

含 n 个未知量 x_1, x_2, \dots, x_n , m 个方程的线性方程组的一般形式为

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m. \end{array} \right. \quad (5.3)$$

若记

$$\mathbf{A} = (a_{ij})^{m \times n} \quad \mathbf{x} = (x_1, x_2, \dots, x_n)^T \quad \mathbf{b} = (b_1, b_2, \dots, b_m)^T, \quad (5.4)$$

则方程组(5.3)可表为如下的矩阵形式:

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \quad (5.5)$$

当 $\mathbf{b} = \mathbf{0}$ 时, 方程组(5.5)所对应的齐次线性方程组为

$$\mathbf{A}\mathbf{x} = \mathbf{0}. \quad (5.6)$$

定义 5.1.1. 矩阵 $\mathbf{A} = (a_{ij})^{m \times n}$, 称为方程组(5.5)的系数矩阵, 而矩阵 $\tilde{\mathbf{A}} = (\mathbf{A}, \mathbf{b})$ 称为它的增广矩阵。方程组(5.6)称为方程组(5.5)的导出组。

定义 5.1.2. 给定方程组

$$\bar{\mathbf{A}}\bar{\mathbf{x}} = \bar{\mathbf{b}}. \quad (5.7)$$

其中 $\bar{\mathbf{A}} = (\bar{a}_{ij})^{m \times n}$, $\bar{\mathbf{x}} = (x_1, x_2, \dots, x_n)^T$, $\bar{\mathbf{b}} = (\bar{b}_1, \bar{b}_2, \dots, \bar{b}_m)^T$ 。当 $\bar{\mathbf{x}}^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$ 是方程组(5.7)的解向量时, 若它也是方程组(5.5)的解向量, 则称方程组(5.5)与方程组(5.7)是同解方程组。

定理 5.1.1. 对方程组(5.5)的系数矩阵 \mathbf{A} 及右端作相同的行初等变换, 所得到的新方程组与原方程组同解。

定义 5.1.3. 设 $\eta_1, \eta_2, \dots, \eta_t$ 是齐次线性方程组(5.6)的解向量组, 如果 $\eta_1, \eta_2, \dots, \eta_t$ 线性无关, 且方程组(5.6)的任意解向量 η 都可由 $\eta_1, \eta_2, \dots, \eta_t$ 线性表出, 则称解向量组 $\eta_1, \eta_2, \dots, \eta_t$ 为方程组(5.6)的一个基础解系。

定理 5.1.2. 设齐次线性方程组(5.6)的系数矩阵 \mathbf{A} 的秩为 r , 此时

- (1) 方程组(5.6)有非零解的必要充分条件是 $r < n$ 。
- (2) 若 $r < n$, 则方程组(5.6)一定有基础解系。基础解系不是唯一的, 但任两个基础解系必等价, 且每一个基础解系所含解向量的个数都等于 $n - r$ 。
- (3) 若 $r < n$, 设 $\eta_1, \eta_2, \dots, \eta_{n-r}$ 是方程组(5.6)的一个基础解系, 则它的一般解为

$$\eta = \lambda_1\eta_1 + \lambda_2\eta_2 + \cdots + \lambda_{n-r}\eta_{n-r}, \quad (5.8)$$

其中 $\lambda_i (i = 1, 2, \dots, n - r)$ 是数域 \mathbb{K} 中的任意常数。

定理 5.1.3. 方程组(5.5)有解的必要充分条件是: $\text{rank}(\mathbf{A}) = \text{rank}(\tilde{\mathbf{A}})$. 矩阵 $\tilde{\mathbf{A}}$ 为它的增广矩阵.

定理 5.1.4. 设 $\text{rank}(\mathbf{A}) = \text{rank}(\tilde{\mathbf{A}}) = r, \gamma_0$ 是非齐次方程组(5.5)的一个解向量 (常称为 特解), $\eta_1, \eta_2, \dots, \eta_{n-r}$ 是其导出组(5.6)的一个基础解系, 则方程组(5.5)的解向量均可表为:

$$\gamma = \gamma_0 + \eta = \gamma_0 + \lambda_1 \eta_1 + \lambda_2 \eta_2 + \dots + \lambda_{n-r} \eta_{n-r},$$

其中 $\lambda_i (i = 1, 2, \dots, n-r)$ 是数域 \mathbb{K} 中的任意常数 (这种形式的解向量常称为一般解).

对增广矩阵 $\tilde{\mathbf{A}}$ 进行初等行变换将其化为阶梯型矩阵, 写出相应的阶梯型方程组.

(1) 若 $r = n$, 则阶梯型方程组形如:

$$\left\{ \begin{array}{l} c_{11}x_1 + c_{12}x_2 + \dots + c_{1n}x_n = d_1, \\ c_{22}x_2 + \dots + c_{2n}x_n = d_2, \\ \dots \dots \dots \\ c_{nn}x_n = d_n. \end{array} \right. \quad (5.9)$$

其中 $c_{ii} \neq 0 (i = 1, 2, \dots, n)$. 依次由第 n 个, 第 $n-1$ 个, \dots, 第一个方程组可解得 x_n, x_{n-1}, \dots, x_1 , 由此即得方程组(5.3)的唯一解 x_1, x_2, \dots, x_n .

(2) 若 $r < n$, 则阶梯型方程组可表为:

$$\left\{ \begin{array}{l} c_{11}x_1 + c_{12}x_2 + \dots + c_{1r}x_r = d_1 - c_{1,r+1}x_{r+1} - \dots - c_{1n}x_n, \\ c_{22}x_2 + \dots + c_{2r}x_r = d_2 - c_{2,r+1}x_{r+1} - \dots - c_{2n}x_n, \\ \dots \dots \dots \\ c_{rr}x_r = d_r - c_{r,r+1}x_{r+1} - \dots - c_{rn}x_n. \end{array} \right. \quad (5.10)$$

其中 $c_{ii} \neq 0 (i = 1, 2, \dots, r)$. 此时方程组(5.3)有无穷多组解. 若令 $x_{r+1} = x_{r+2} = \dots = x_n = 0$, 则可由方程组(5.10)求得一个特解 $\gamma_0 = (\Delta_1, \Delta_2, \dots, \Delta_r, 0, 0, \dots, 0)$, 再由其导出组的一个基础解系 $\eta_1, \eta_2, \dots, \eta_{n-r}$, 可得方程组(5.3)的一般解.

线性方程组的解集被定义为:

$$S \doteq \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{y}\} \quad (5.11)$$

用 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n \in \mathbb{R}^n$ 表示矩阵 \mathbf{A} 的列, 即 $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$. \mathbf{Ax} 仅仅表示矩阵 \mathbf{A} 的列与向量 \mathbf{x} 中各个元素的加权和:

$$\mathbf{Ax} = x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \dots + x_n \mathbf{a}_n \quad (5.12)$$

通过定义, 我们能够看出, 无论 \mathbf{x} 的值是什么, \mathbf{Ax} 生成了由矩阵 \mathbf{A} 的列张成的子空间. 向量 $\mathbf{Ax} \in \text{Range}(\mathbf{A})$. 若 $\mathbf{y} \notin \text{Range}(\mathbf{A})$, 则线性方程组没有解. 因此解集 S 为空. 等价地, 线性方程组有解当且仅当 $\mathbf{y} \in \text{Range}(\mathbf{A})$.

回顾定义, 设 $\mathbf{A} \in \mathbb{R}^{m \times n}$. \mathbf{A} 的值域定义为

$$\text{Range}(\mathbf{A}) = \{\mathbf{y} \in \mathbb{R}^m : \mathbf{y} = \mathbf{Ax}, \mathbf{x} \in \mathbb{R}^n\} \quad (5.13)$$

易证 $\text{Range}(\mathbf{A}) = \text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_n\}$, 其中 \mathbf{a}_i 为 \mathbf{A} 的列向量. \mathbf{A} 的零空间定义为:

$$\text{Null}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = 0\} \quad (5.14)$$

它的维数记为 $\text{nullity}(\mathbf{A})$

一个子空间 $\mathbb{S} \subset \mathbb{R}^n$ 的正交补定义为:

$$\mathbb{S}^\perp = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y}^\top \mathbf{x} = 0, \forall \mathbf{x} \in \mathbb{S}\} \quad (5.15)$$

从矩阵值域的角度, 给出定理5.1.3的证明:

定理 5.1.5. 方程组的解(5.5)存在的充分必要条件是 $\text{rank}(\mathbf{A}) = \text{rank}([\mathbf{A}, \mathbf{b}])$.

证明. 必要性: 设存在 \mathbf{x} 使 $\mathbf{Ax} = \mathbf{b}$, 则 \mathbf{b} 是 \mathbf{A} 的列向量的线性组合, 即 $\mathbf{b} \in \mathcal{R}(\mathbf{A})$. 这说明 $\mathcal{R}([\mathbf{A}, \mathbf{b}]) = \mathcal{R}(\mathbf{A})$, 所以有 $\text{rank}(\mathbf{A}) = \text{rank}([\mathbf{A}, \mathbf{b}])$.

充分性: 若 $\text{rank}(\mathbf{A}) = \text{rank}([\mathbf{A}, \mathbf{b}])$ 成立, 则 $\mathbf{b} \in \mathcal{R}(\mathbf{A})$, 即 \mathbf{b} 可表示为 $\mathbf{b} = \sum_{i=1}^n \mathbf{x}_i \mathbf{a}_i$, 这里 $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n]$, 所以令 $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$, 即有 $\mathbf{Ax} = \mathbf{b}$. \square

定理 5.1.6. 假定方程组(5.5)的解存在, 并且假定 \mathbf{x} 是其任一给定的解, 则(5.5)全部解的集合是

$$\mathbf{x} + \text{Null}(\mathbf{A}) \quad (5.16)$$

证明. 如果 \mathbf{y} 满足(5.5), 则 $\mathbf{A}(\mathbf{y} - \mathbf{x}) = 0$, 即 $(\mathbf{y} - \mathbf{x}) \in \text{Null}(\mathbf{A})$, 于是有 $\mathbf{y} = \mathbf{x} + (\mathbf{y} - \mathbf{x}) \in \mathbf{x} + \text{Null}(\mathbf{A})$. 反之, 如果 $\mathbf{y} \in \mathbf{x} + \text{Null}(\mathbf{A})$, 则存在 $\mathbf{z} \in \text{Null}(\mathbf{A})$, 使 $\mathbf{y} = \mathbf{x} + \mathbf{z}$, 从而有 $\mathbf{Ay} = \mathbf{Ax} + \mathbf{Az} = \mathbf{Ax} = \mathbf{b}$. \square

定理5.1.6告诉我们, 只要知道了方程组(5.5)的一个解, 便可以用它及 $\text{Null}(\mathbf{A})$ 中向量的和得到(5.5)的全部解. 由此可知, 方程组(5.5)的解唯一, 只有当 $\text{Null}(\mathbf{A})$ 中仅有零向量才行.

推论 5.1.1. 方程组(5.5)的解唯一的充分必要条件是 $\text{nullity}(\mathbf{A}) = 0$.

当线性方程组的系数矩阵 \mathbf{A} 不为可逆方阵时, 矩阵的广义逆是表示一般线性方程组通解的强大有力工具。

定理 5.1.7. 设 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, 则线性方程组(??)有解的充要条件是

$$\mathbf{AA}^\dagger \mathbf{b} = \mathbf{b}$$

并且在有解时, 其通解为

$$\mathbf{x} = \mathbf{A}^\dagger \mathbf{b} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \mathbf{z}$$

其中 $\mathbf{z} \in \mathbb{R}^n$

注: 上述结论对于复矩阵和复向量也成立。

证明. 若 $\mathbf{A}\mathbf{A}^\dagger \mathbf{b} = \mathbf{b}$, 则方程组 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 显然有解 $\mathbf{x} = \mathbf{A}^\dagger \mathbf{b}$ 。反之若 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 则

$$\mathbf{A}\mathbf{A}^\dagger \mathbf{b} = \mathbf{A}\mathbf{A}^\dagger \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x} = \mathbf{b}$$

下面证明通解为

$$\mathbf{x} = \mathbf{A}^\dagger \mathbf{b} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A})\mathbf{z}$$

事实上, 考虑矩阵 $\text{Null}(\mathbf{A})$ 上的投影矩阵

$$\mathbf{P}_{\text{Null}(\mathbf{A})} = \mathbf{I} - \mathbf{A}^\dagger \mathbf{A}$$

故

$$\text{Null}(\mathbf{A}) = \{(\mathbf{A} - \mathbf{A}^\dagger \mathbf{A})\mathbf{z} \mid \mathbf{z} \in \mathbb{R}^n\}$$

综上, 线性方程组 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 通解为

$$\mathbf{x} = \mathbf{A}^\dagger \mathbf{b} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A})\mathbf{z}$$

其中 $\mathbf{z} \in \mathbb{R}^n$

□

对于不同类型的方程组, 解的数量情况有各自的特点。

超定方程组

线性方程组 $\mathbf{A}\mathbf{x} = \mathbf{y}$ 中线性方程的个数大于未知变量的个数时, 我们说 $\mathbf{A}\mathbf{x} = \mathbf{y}$ 是超定的. 或者说矩阵 $\mathbf{A}^{m \times n}$ 的行数大于列数: $m > n$. 假设 \mathbf{A} 是一个列满秩矩阵, 也就是说 $\text{rank}(\mathbf{A}) = n$, 则我们可以得出 $\text{Null}(\mathbf{A}) = 0$.

因此线性方程组的解要么没有解, 要么有唯一解. 在超定方程组中, $\mathbf{y} \notin \text{Range}(\mathbf{A})$ 是很常见的, 因此引入近似解的概念, 近似解使得 $\mathbf{A}\mathbf{x}$ 与 \mathbf{y} 在合适的度量下距离最小.

欠定方程组

线性方程组 $\mathbf{A}\mathbf{x} = \mathbf{y}$ 中未知变量的个数小于方程组的个数时, 我们说线性方程组是欠定的. 或者说 $\mathbf{A}^{m \times n}$ 的列数大于行数: $m < n$. 假设 \mathbf{A} 是一个行满秩矩阵, 也就是说 $\text{rank}(\mathbf{A}) = m$, $\text{Range}(\mathbf{A}) = \mathbb{R}^m$. 则根据定理 5.1.6:

$$\text{rank}(\mathbf{A}) + \dim(\text{Null}(\mathbf{A})) = n \tag{5.17}$$

因此 $\dim(\text{Null}(\mathbf{A})) = n - m > 0$. 此时线性方程组有解且有无限多个解, 并且解集的维度是 $n - m$. 在所有可能的解中, 我们总是对具有最小范数的解很感兴趣.(后面详细讨论).

适定方程组

线性方程组 $\mathbf{Ax} = \mathbf{y}$ 中线性方程的个数等于未知变量的个数时，我们说 $\mathbf{Ax} = \mathbf{y}$ 是适定方程组。或者说 $A^{m \times n}$ 的列数等于行数: $m = n$. 如果系数矩阵是满秩的即 A 可逆, A^{-1} 唯一且有 $A^{-1}A = I$. 在这种情况下，线性方程组的解是唯一的:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y} \quad (5.18)$$

注意，实际中我们几乎不会通过先求 \mathbf{A}^{-1} 再乘以向量 \mathbf{y} 的方式求解 \mathbf{x} . 而是通过数值方法(比如之前学过的 LU 分解, Cholesky 分解)来计算线性方程组非奇异方程组的解.

5.1.2 容易求解的线性方程组

基于浮点运算次数的复杂性分析

数值线性代数算法的成本经常表示为完成算法所需的浮点运算次数关于各种问题维度的函数。

定义 5.1.4. 两个浮点数做一次相加、相减、相乘或相除称为一次浮点运算。

为了顾及一个算法的复杂性，我们计算总的浮点运算次数，将其表示为所涉及的矩阵或向量的维数的函数(通常是多项式)，并通过只保留主导(即最高次数或占优势)项的方式来简化所得到的表达式。

例 5.1.2. 假设一个具体的算法需要总数为

$$m^3 + 3m^2n + mn + 4mn^2 + 5m + 22$$

次浮点运算，其中 m, n 是问题的维数。正常情况下，我们将其简化为

$$m^3 + 3m^2n + 4mn^2$$

次浮点运算，因为这些是问题维数 m, n 的主导项。如果此外又假设 m 远小于 n ，我们将进一步将浮点运算次数简化为 $4mn^2$ 。

为了完成两个向量 $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ 的内积运算 $\mathbf{x}^T\mathbf{y}$ ，我们先要计算乘积 $x_i y_i$ ，然后将它们相加，这需要 n 次乘法和 $n - 1$ 次加法，或者为 $2n - 1$ 次浮点运算。只保留主导项，称内积运算需要 $2n$ 次浮点运算，甚至更近似地说，需要次数为 n 的浮点运算。

例 5.1.3. 矩阵与向量相乘 $\mathbf{y} = \mathbf{Ax}$ ，其中 $\mathbf{A} \in \mathbb{R}^{m \times n}$ ，成本为 $2mn$ 次浮点运算：我们必须计算 \mathbf{y} 的 m 个分量，每一个分量是 \mathbf{A} 的行向量和 \mathbf{x} 的内积。

例 5.1.4. 矩阵与矩阵相乘 $\mathbf{C} = \mathbf{AB}$ ，其中 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times p}$ ，需要 $2mnp$ 次浮点运算，因为我们需要计算 \mathbf{C} 的 mp 个元素，而每一个元素都是两个长度为 n 的向量的内积。

对角形方程组

我们首先考虑一个最简单的线性方程组

$$\begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix} \mathbf{x} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

其中 $a_{ii} \neq 0, i = 1, 2, \dots, n$ 。那么就有

$$\mathbf{x} = \begin{pmatrix} b_1/a_{11} \\ b_2/a_{22} \\ \vdots \\ b_n/a_{nn} \end{pmatrix}$$

我们只需要经过 n 次浮点运算就可以求得。

三角形线性方程组

在前面，我们考虑了理论上如何对一个一般的线性方程组求解。

接下来，我们将考虑如何使用计算机对一个线性方程组求解，尤其是一个规模巨大的线性方程组。

首先来考虑一个稍微简单些的情况，三角形线性方程组。

定义 5.1.5. 如果一个矩阵 A 主对角线以上所有元素为 0，则称其为下三角矩阵。

如果一个矩阵 A 主对角线以下的所有元素为 0，则称其为上三角矩阵。

定义 5.1.6. 如果一个线性方程组 $A\mathbf{x} = \mathbf{b}$ 的系数矩阵 A 是上三角形矩阵或者下三角矩阵，我们就称其为上三角形线性方程组或者下三角形线性方程组。

针对上三角形线性方程组和下三角形线性方程组，我们可以分别用两种特别的方法解出方程组的解。

接下来，我们分别介绍这两种方法。

我们利用前代法计算下三角形线性方程组。

注意，我们要求系数矩阵主对角线上元素均非 0。从而保证方程组有且仅有一个解。

$$\begin{pmatrix} a_{11} & & & & \\ a_{21} & a_{22} & & & \\ a_{31} & a_{32} & a_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}$$

其中 $a_{11}, a_{22}, \dots, a_{nn}$ 非 0.

在前代法的第 (k) 个循环中，我们将会遇到下面这样一个形式

$$\left(\begin{array}{cccccc|c} a_{11} & & & & & & b_1 \\ 0 & a_{22} & & & & & b_2^{(1)} \\ 0 & 0 & a_{33} & & & & b_3^{(2)} \\ \vdots & \vdots & \vdots & \ddots & & & \vdots \\ 0 & 0 & 0 & \dots & a_{kk} & & b_k^{(k-1)} \\ 0 & 0 & 0 & \dots & a_{k+1k} & a_{k+1k+1} & b_{k+1}^{(k-1)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{nk} & a_{nk+1} & \dots & a_{nn} \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ \vdots \\ b_k^{(k-1)} \\ b_{k+1}^{(k-1)} \\ \vdots \\ b_n^{(k-1)} \end{pmatrix}$$

此时我们将第 k 列从第 $k+1$ 行到第 n 行化为 0，同时更新 b 。

$$\left(\begin{array}{cccccc|c} a_{11} & & & & & & b_1 \\ 0 & a_{22} & & & & & b_2^{(1)} \\ 0 & 0 & a_{33} & & & & b_3^{(2)} \\ \vdots & \vdots & \vdots & \ddots & & & \vdots \\ 0 & 0 & 0 & \dots & a_{kk} & & b_k^{(k-1)} \\ 0 & 0 & 0 & \dots & 0 & a_{k+1k+1} & b_{k+1}^{(k)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & a_{nk+1} & \dots & a_{nn} \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ \vdots \\ b_k^{(k-1)} \\ b_{k+1}^{(k)} \\ \vdots \\ b_n^{(k)} \end{pmatrix}$$

所以前代法，就从前 (x_1) 往后 (x_n) 来依次求解。

Algorithm 8 前代法

```

1:  $x_1 = b_1/a_{11}$ 
2: for  $i = 2$  to  $n$  do
3:    $s = b_i$ 
4:   for  $j = 1, \dots, i - 1$  do
5:      $s = s - a_{ij}x_j$ 
6:   end for
7:    $x_i = s/a_{ii}$ 
8: end for
```

回代法则恰好相反，他是从后往前一次求解。

回代法是用于上三角形的线性方程组求解。

同样我们要求其系数矩阵对角线上元素非 0.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{22} & \dots & a_{2n} \\ \ddots & & \vdots \\ & & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

与前代法类似，在回代法第 $(n - k + 1)$ 个循环内。

$$\left(\begin{array}{ccccccc} a_{11} & a_{12} & \dots & a_{1k-1} & a_{1k} & 0 & \dots & 0 \\ a_{22} & \dots & a_{2k-1} & a_{2k} & 0 & \dots & 0 \\ \ddots & & \vdots & \vdots & \vdots & & \vdots \\ a_{k-1k-1} & a_{k-1k} & 0 & \dots & 0 \\ a_{kk} & 0 & \dots & 0 \\ a_{k+1k+1} & \dots & 0 \\ \ddots & & \vdots \\ & & a_{nn} \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^{(n-k)} \\ b_2^{(n-k)} \\ \vdots \\ b_{k-1}^{(n-k)} \\ b_k^{(n-k)} \\ b_{k+1}^{(n-k-1)} \\ \vdots \\ b_n \end{pmatrix}$$

此时我们将第 k 列从第 1 行到第 $k - 1$ 行化为 0，同时更新 b 。

$$\left(\begin{array}{ccccccc} a_{11} & a_{12} & \dots & a_{1k-1} & 0 & 0 & \dots & 0 \\ a_{22} & \dots & a_{2k-1} & 0 & 0 & \dots & 0 \\ \ddots & & \vdots & \vdots & & & \vdots \\ a_{k-1k-1} & 0 & 0 & \dots & 0 \\ a_{kk} & 0 & \dots & 0 \\ a_{k+1k+1} & \dots & 0 \\ \ddots & & \vdots \\ & & a_{nn} \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^{(n-k+1)} \\ b_2^{(n-k+1)} \\ \vdots \\ b_{k-1}^{(n-k+1)} \\ b_k^{(n-k)} \\ b_{k+1}^{(n-k-1)} \\ \vdots \\ b_n \end{pmatrix}$$

草稿请勿外传

Algorithm 9 回代法

```

1:  $x_n = b_n/a_{nn}$ 
2: for  $i = n - 1$  to 1 do
3:    $s = b_i$ 
4:   for  $j = i + 1, \dots, n$  do
5:      $s = s - a_{ij}x_j$ 
6:   end for
7:    $x_i = s/a_{ii}$ 
8: end for

```

正交矩阵

- 矩阵 $A \in \mathbb{R}^{n \times n}$ 被称为正交矩阵的条件是 $A^T A = I$ 即 $A^{-1} = A^T$ 。这种情况下可以通过简单的矩阵-向量乘积 $\mathbf{x} = A^T \mathbf{b}$ 计算 $\mathbf{x} = A^{-1} \mathbf{b}$ ，一般情况其计算成本为 $2n^2$ 次浮点运算。

- 如果矩阵 A 有其他结构，计算 $\mathbf{x} = A^{-1} \mathbf{b}$ 的效率可以超过 $2n^2$ 。例如，如果 A 具有 $A = I - 2\mathbf{u}\mathbf{u}^T$ 的形式，其中 $\|\mathbf{u}\|_2 = 1$ ，此时

$$\mathbf{x} = A^{-1} \mathbf{b} = (I - 2\mathbf{u}\mathbf{u}^T)^T \mathbf{b} = \mathbf{b} - 2(\mathbf{u}^T \mathbf{b})\mathbf{u}$$

我们可以先计算 $\mathbf{u}^T \mathbf{b}$ ，然后计算 $\mathbf{b} - 2(\mathbf{u}^T \mathbf{b})\mathbf{u}$ ，其计算成本为 $4n$ 次浮点运算。

排列矩阵

- 令 $\pi = (\pi_1, \dots, \pi_n)$ 为 $(1, 2, \dots, n)$ 的一个排列或置换。相应的排列矩阵或置换矩阵 $A \in \mathbb{R}^{n \times n}$ 定义为

$$A_{ij} = \begin{cases} 1 & j = \pi_i \\ 0 & \text{otherwise} \end{cases}$$

排列矩阵的每行（或每列）仅有一个元素等于 1，所有其他元素都等于 0。用排列矩阵乘一个向量就是对其分量进行如下排列：

$$Ax = (x_{\pi_1}, \dots, x_{\pi_n})$$

排列矩阵的逆矩阵就是逆排列 π^{-1} 对应的排列矩阵，实际上就是 A^T 。由此可知排列矩阵是正交矩阵。

排列矩阵或置换矩阵

- 如果 A 是排列矩阵，求解 $Ax = \mathbf{b}$ 将非常容易，用 π^{-1} 对 \mathbf{b} 元素进行排列就可以得到 \mathbf{x} 。这样做并不需要我们定义浮点运算（但是，取决于具体实现，可能要复制浮点数）。从方程

$\mathbf{x} = \mathbf{A}^T \mathbf{b}$ 可以达到同样的结论。矩阵 \mathbf{A}^T （像 \mathbf{A} 一样）的每行仅有一个等于 1 的非零元素。因此不需要加法运算，而唯一需要的乘法是和 1 相乘。

5.1.3 线性方程组的直接解法

本小节从矩阵分解的角度探讨线性方程组的直接解法。

基于矩阵分解的方阵系统的一般求解方法

我们首先讨论的一类特殊的方阵线性方程组

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{A} \in \mathbb{R}^{n \times n}$$

的求解，其中 \mathbf{A} 可逆。其基本思路是：

- 我们可以将矩阵分解成一系列特殊结构矩阵的乘积，包括：对角矩阵、上下三角矩阵、正交矩阵和排列矩阵等。
- 然后我们通过对具有特殊结构更简单的方程组的求解来获得原方程组的解。

这种方法的一个优势是，一旦我们对系数矩阵进行了分解，那么对于不同的右侧项就无需重新计算。而且从计算复杂性的角度看，计算成本主要集中在矩阵的因式分解上。

求解 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 的基本途径是将 \mathbf{A} 表示为一系列非奇异矩阵的乘积

$$\mathbf{A} = \mathbf{A}_1 \mathbf{A}_2 \dots \mathbf{A}_k$$

因此

$$\mathbf{x} = \mathbf{A}^{-1} \mathbf{b} = \mathbf{A}_k^{-1} \mathbf{A}_{k-1}^{-1} \dots \mathbf{A}_1^{-1} \mathbf{b}$$

我们可以从右到左利用这个公式计算 \mathbf{x} :

$$\begin{aligned} \mathbf{z}_1 &:= \mathbf{A}_1^{-1} \mathbf{b} \\ \mathbf{z}_2 &:= \mathbf{A}_2^{-1} \mathbf{z}_1 = \mathbf{A}_2^{-1} \mathbf{A}_1^{-1} \mathbf{b} \\ &\vdots \\ \mathbf{z}_{k-1} &:= \mathbf{A}_{k-1}^{-1} \mathbf{z}_{k-2} = \mathbf{A}_{k-1}^{-1} \dots \mathbf{A}_1^{-1} \mathbf{b} \\ \mathbf{x} &:= \mathbf{A}_k^{-1} \mathbf{z}_{k-1} = \mathbf{A}_k^{-1} \dots \mathbf{A}_1^{-1} \mathbf{b} \end{aligned}$$

- 这个过程的第 i 步需要计算 $\mathbf{z}_i = \mathbf{A}_i^{-1} \mathbf{z}_{i-1}$ 即求解线性方程组 $\mathbf{A}_i \mathbf{z}_i = \mathbf{z}_{i-1}$ 如果这些方程组都容易求解（即如果 \mathbf{A}_i 是对角矩阵，下三角矩阵或上三角矩阵，排列矩阵等等），这就形成了计算 $\mathbf{x} = \mathbf{A}^{-1} \mathbf{b}$ 的一种方法。

- 将 \mathbf{A} 表示为因式分解形式（即计算 $\mathbf{A} = \mathbf{A}_1 \mathbf{A}_2 \dots \mathbf{A}_k$ ）的步骤被称为矩阵分解步骤，而通过递推求解一系列 $\mathbf{A}_i \mathbf{z}_i = \mathbf{z}_{i-1}$ 来计算 $\mathbf{x} = \mathbf{A}^{-1} \mathbf{b}$ 的过程经常被称为求解步骤。

- 采用这种矩阵因式分解求解方法求解 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 的总的浮点运算次数是 $f + s$, 其中 f 是进行因式分解的浮点运算次数, s 是求解步骤的总的浮点运算次数。很多情况下, 因式分解的成本 f , 相对总的求解成本 s 占主导地位。因此求解 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 的成本, 即计算 $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ 就是 f 。

LU 分解解方程

设矩阵 \mathbf{A} 有 LU 分解 $\mathbf{A} = \mathbf{PLU}$, 其中 \mathbf{P} 是排列矩阵, \mathbf{L} 是下三角矩阵, \mathbf{U} 是上三角矩阵。这种形式被称为 \mathbf{A} 的 LU 因式分解。我们也可以把因式分解写成 $\mathbf{P}^T\mathbf{A} = \mathbf{LU}$, 其中矩阵 $\mathbf{P}^T\mathbf{A}$ 通过重排列 \mathbf{A} 的行向量得到。那么我们在求解方程组

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

时, 等价求解一系列如下方程组

$$\mathbf{P}\mathbf{z}_1 = \mathbf{b}, \mathbf{L}\mathbf{z}_2 = \mathbf{z}_1, \mathbf{U}\mathbf{x} = \mathbf{z}_2$$

对于第一个方程, 我们只需要根据其排列规则来将 \mathbf{b} 重新排列。对于第二个下三角方程, 我们使用前代法来求解。对于第三个上三角方程, 我们使用回代法来求解。

Algorithm 10 利用 LU 因式分解求解线性方程组

- 1: LU 因式分解。将 \mathbf{A} 因式分解为 $\mathbf{A} = \mathbf{PLU}$ ($(2/3)n^3$ 次浮点运算)。
- 2: 排列。求解 $\mathbf{P}\mathbf{z}_1 = \mathbf{b}$ (0 次浮点运算)。
- 3: 前向代入。求解 $\mathbf{L}\mathbf{z}_2 = \mathbf{z}_1$ (n^2 次浮点运算)。
- 4: 后向代入。求解 $\mathbf{U}\mathbf{x} = \mathbf{z}_2$ (n^2 次浮点运算)。

因为在计算机上求解方程, 我们还需要考虑资源问题, 为了节约资源, 下面给出一种紧凑的求解方式。

给定矩阵 \mathbf{A} 和向量 \mathbf{b} , 我们先对 \mathbf{A} 进行 LU 分解。并且使用 \mathbf{A} 的上三角部分存储上三角矩阵, 用下三角部分存储下三角矩阵。

比如矩阵

$$\begin{pmatrix} 3 & 2 & -1 \\ 6 & 6 & -2 \\ -3 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 3 & 2 & -1 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

就可以使用

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

来存储。

然后我们来重新给出 LU 分解的算法流程。

Algorithm 11 计算机上的 LU 分解

- 1: $u_{1i} = a_{1i}, \quad i = 1, \dots, n;$
 - 2: $l_{i1} = a_{i1}/u_{11}, i = 2, \dots, n;$
 - 3: **for** $k = 2, \dots, n$ **do**
 - 4: $u_{ki} = a_{ki} - \sum_{r=1}^{k-1} l_{kr} u_{ri}, \quad i = k, \dots, n;$
 - 5: $l_{ik} = (a_{ik} - \sum_{r=1}^{k-1} l_{kr} u_{ri})/u_{kk}, \quad k = 2, \dots, n.$
 - 6: **end for**
-

最后再使用前代法和回代法求出最终的解。

例 5.1.5. 求解 $\begin{pmatrix} 3 & 2 & -1 \\ 6 & 6 & -2 \\ -3 & 2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -2 \\ -5 \end{pmatrix}$

我们先对 $\begin{pmatrix} 3 & 2 & -1 \\ 6 & 6 & -2 \\ -3 & 2 & 0 \end{pmatrix}$ LU 分解。

$$\rightarrow \begin{pmatrix} 3 & 2 & -1 \\ 2 & 6 & -2 \\ -1 & 2 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & 2 & -1 \\ 2 & 2 & 0 \\ -1 & 4 & -1 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & 2 & -1 \\ 2 & 2 & 0 \\ -1 & 2 & -1 \end{pmatrix} \rightarrow \begin{pmatrix} 3 & 2 & -1 \\ 2 & 2 & 0 \\ -1 & 2 & -1 \end{pmatrix}$$

然后再进行前代法，注意此处，我们只关心 \mathbf{y} ，不关注 \mathbf{L} 如何变换，故 \mathbf{L} 并不需要实际上去化为 0。可得

$$\hat{\mathbf{y}} = \begin{pmatrix} 0 \\ -2 \\ -1 \end{pmatrix}$$

最后进行回代法，便得解 $(1, -1, 1)^T$.

其他类型的分解解法

基于 Cholesky 分解求解对称正定线性方程组 设矩阵 \mathbf{A} 有 Cholesky 分解 $\mathbf{A} = \mathbf{LL}^T$ ，其中 \mathbf{L} 是下三角矩阵。那么我们在求解方程组

$$\mathbf{Ax} = \mathbf{b}$$

时，等价求解一系列如下方程组

$$\mathbf{L}\mathbf{z}_1 = \mathbf{b}, \mathbf{L}^T \mathbf{x} = \mathbf{z}_1$$

对于第一个下三角方程，我们使用前代法来求解。对于第二个上三角方程，我们使用回代法来求解。

Algorithm 12 基于 Cholesky 分解求解对称正定线性方程组

- 1: Cholesky 因式分解。将 \mathbf{A} 因式分解为 $\mathbf{A} = \mathbf{L}\mathbf{L}^T$ ($(1/3)n^3$ 次浮点运算)。
 - 2: 前向代入。求解 $\mathbf{L}\mathbf{z}_1 = \mathbf{b}$ (n^2 次浮点运算)。
 - 3: 后向代入。求解 $\mathbf{L}^T \mathbf{x} = \mathbf{z}_1$ (n^2 次浮点运算)。
-

基于 QR 分解求解线性方程组 设可逆矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 有 QR 分解 $\mathbf{A} = \mathbf{Q}\mathbf{R}$ 其中 \mathbf{Q} 是正交矩阵 \mathbf{R} 是主对角线均为正的上三角矩阵。那么我们在求解方程组

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

时，等价求解方程组

$$\mathbf{R}\mathbf{x} = \mathbf{Q}^T \mathbf{b}$$

而对于上三角矩阵的方程组，我们可以使用回代法来求解。

Algorithm 13 基于 QR 分解求解线性方程组

- 1: QR 因式分解。将 \mathbf{A} 因式分解为 $\mathbf{A} = \mathbf{Q}\mathbf{R}$ ($4n^3$ 次浮点运算)。
 - 2: 矩阵-向量乘法。求解 $\mathbf{z} = \mathbf{Q}^T \mathbf{b}$ ($2n^2$ 次浮点运算)。
 - 3: 后向代入。求解 $\mathbf{R}\mathbf{x} = \mathbf{z}$ (n^2 次浮点运算)。
-

基于 SVD 求解线性方程组 设矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 的奇异值分解为 $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ ，其中 \mathbf{U}, \mathbf{V} 是正交矩阵， Σ 是对角矩阵且可逆。那么我们在求解方程组

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

时，等价求解一系列如下方程组

$$\mathbf{U}\mathbf{y} = \mathbf{b}, \Sigma\mathbf{z} = \mathbf{y}, \mathbf{V}^T \mathbf{x} = \mathbf{z}$$

而这些方程对应的解为

$$\mathbf{y} = \mathbf{U}^T \mathbf{b}, \mathbf{z} = \Sigma^{-1} \mathbf{y}, \mathbf{x} = \mathbf{V} \mathbf{z}$$

Algorithm 14 基于 SVD 求解线性方程组

- 1: SVD 因式分解。将 \mathbf{A} 因式分解为 $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$ (n^3 次浮点运算)。
 - 2: 矩阵-向量乘法。求解 $\mathbf{U}\mathbf{y} = \mathbf{b}$ ($2n^2$ 次浮点运算)。
 - 3: 求解对角方程组。求解 $\Sigma\mathbf{z} = \mathbf{y}$ (n 次浮点运算)。
 - 4: 矩阵-向量乘法。求解 $\mathbf{V}^T \mathbf{x} = \mathbf{z}$ ($2n^2$ 次浮点运算)。
-

非方阵系统的一般求解方法

上面考虑了方阵系统，我们接下来考虑非方阵系统

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m$$

非方阵系统求解方法：欠定系统的求解 设 $\mathbf{A} \in \mathbb{R}^{m \times n}, m < n$, 此时方程组为欠定系统, 如果 $\text{rank}(\mathbf{A}) = m$, 则对任意的 \mathbf{b} 至少存在一个解。很多实际应用中找到一个具体的解 $\hat{\mathbf{x}}$ 就足以解决问题。其他一些情况下我们可能需要给出所有解的参数化描述

$$\{\mathbf{x} | \mathbf{A}\mathbf{x} = \mathbf{b}\} = \{\mathbf{F}\mathbf{z} + \hat{\mathbf{x}} | \mathbf{z} \in \mathbb{R}^{n-m}\}$$

其中 \mathbf{F} 的列向量构成 \mathbf{A} 的零空间的基。

如果已知 \mathbf{A} 的一个 $m \times m$ 的非奇异子矩阵, 可以直接求解非方阵系统。假设 \mathbf{A} 的前 m 个列向量线性无关。于是可以将方程 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 写成

$$\mathbf{A}\mathbf{x} = (\mathbf{A}_1, \mathbf{A}_2) \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 = \mathbf{b}$$

其中 $\mathbf{A}_1 \in \mathbb{R}^{m \times m}$ 是非奇异矩阵。我们可以将 \mathbf{x}_1 表示成

$$\mathbf{x}_1 = \mathbf{A}_1^{-1}(\mathbf{b} - \mathbf{A}_2\mathbf{x}_2) = \mathbf{A}_1^{-1}\mathbf{b} - \mathbf{A}_1^{-1}\mathbf{A}_2\mathbf{x}_2$$

该表达式让我们能很容易地计算一个解: 简单取 $\hat{\mathbf{x}}_2 = 0, \hat{\mathbf{x}}_1 = \mathbf{A}_1^{-1}\mathbf{b}$ 。其计算成本等于求解 m 个线性方程组 $\mathbf{A}_1\hat{\mathbf{x}}_1 = \mathbf{b}$ 的成本。我们也可以用 $\mathbf{x}_2 \in \mathbb{R}^{n-m}$ 做自由参数表示 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 的所有解。方程 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 的一般性解可以表示成

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} -\mathbf{A}_1^{-1}\mathbf{A}_2 \\ \mathbf{I} \end{pmatrix} \mathbf{x}_2 + \begin{pmatrix} \mathbf{A}_1^{-1}\mathbf{b} \\ 0 \end{pmatrix}$$

综上所述, 假设 \mathbf{A}_1 的因式分解成本是 f , 而求解形如 $\mathbf{A}_1\mathbf{x} = \mathbf{d}$ 的系统成本为 s 那么找出 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 的一个解的成本为 $f + s$ 。参数化描述所有解的成本是 $f + s(n - p + 1)$ 。

现在我们考虑一般情况, 此时 \mathbf{A} 的前 m 个列向量不一定线性独立。因为 $\text{rank}(\mathbf{A}) = m$, 我们可以选出 \mathbf{A} 的 m 个线性独立的列向量, 将他们排列到前面, 然后应用上面描述的方法。换句话说, 我们要找到一个排列矩阵 \mathbf{P} 使 $\tilde{\mathbf{A}} = \mathbf{AP}$ 的前 m 个列向量线性无关, 即

$$\tilde{\mathbf{A}} = \mathbf{AP} = (\mathbf{A}_1, \mathbf{A}_2)$$

其中 \mathbf{A}_1 可逆。方程 $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \mathbf{b}$ 其中 $\tilde{\mathbf{x}} = \mathbf{P}^T\mathbf{x}$, 其一般解

$$\tilde{\mathbf{x}} = \begin{pmatrix} -\mathbf{A}_1^{-1}\mathbf{A}_2 \\ \mathbf{I} \end{pmatrix} \tilde{\mathbf{x}}_2 + \begin{pmatrix} \mathbf{A}_1^{-1}\mathbf{b} \\ 0 \end{pmatrix}$$

于是 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 的一般解为

$$\mathbf{x} = \mathbf{P}\tilde{\mathbf{x}} = \mathbf{P} \begin{pmatrix} -\mathbf{A}_1^{-1}\mathbf{A}_2 \\ \mathbf{I} \end{pmatrix} \mathbf{z} + \mathbf{P} \begin{pmatrix} \mathbf{A}_1^{-1}\mathbf{b} \\ 0 \end{pmatrix}$$

其中 $\mathbf{z} \in \mathbb{R}^{n-m}$ 是自由参数。该想法可用于容易发现 \mathbf{A} 的一个非奇异或便于求逆的子矩阵的情况。例如, 具有非零对角元素的对角矩阵的情况。

QR 因式分解 如果 $A \in \mathbb{R}^{n \times m}$ 满足 $m \leq n$ 和 $\text{rank}(A) = m$, 那么它可以因式分解为

$$A = (\mathbf{Q}_1, \mathbf{Q}_2) \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$$

其中 $(\mathbf{Q}_1, \mathbf{Q}_2)$ 是正交矩阵, $\mathbf{R} \in \mathbb{R}^{m \times m}$ 是具有非零对角元素的上三角矩阵。这称为 A 的 QR 因式分解。QR 因式分解的浮点运算次数是 $2m^2(n - m/3)$ (以因式分解的方式存储 \mathbf{Q} 能够有效计算乘积 $\mathbf{Q}\mathbf{x}$ 和 $\mathbf{Q}^T\mathbf{x}$)

QR 因式分解可以用来解方程组 $A\mathbf{x} = \mathbf{b}, A \in \mathbb{R}^{m \times n}, m < n$ 。假设

$$A^T = (\mathbf{Q}_1, \mathbf{Q}_2) \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$$

是 A^T 的 QR 因式分解。将其代入上述方程组可以看出 $\tilde{\mathbf{x}} = \mathbf{Q}_1 \mathbf{R}^{-T} \mathbf{b}$ 是明显满足该方程组的:

$$A\tilde{\mathbf{x}} = \mathbf{R}^T \mathbf{Q}_1^T \mathbf{Q}_1 \mathbf{R}^{-T} \mathbf{b} = \mathbf{b}$$

此外, \mathbf{Q}_2 的列向量构成 A 的零空间的基, 于是所有的解可以参数化为

$$\{x = \tilde{\mathbf{x}} + \mathbf{Q}_2 z | z \in \mathbb{R}^{n-m}\}$$

QR 因式分解方法是求解非方阵方程组最常用方法。一个缺点是难以利用稀疏性。因式 \mathbf{Q} 通常是稠密的, 甚至 A 很稀疏时也这样。

矩形矩阵的 LU 因式分解 如果 $A \in \mathbb{R}^{n \times m}$ 满足 $m \leq n$ 和 $\text{rank}(A) = m$ 那么它可以因式分解为

$$A = PLU$$

其中 $P \in \mathbb{R}^{n \times n}$ 是排列矩阵, $L \in \mathbb{R}^{n \times m}$ 是单位下三角矩阵, $U \in \mathbb{R}^{m \times m}$ 是非奇异上三角矩阵。如果没有 A 的结构可以利用, 计算成本是 $(2/3)m^3 + m^2(n - m)$ 次浮点运算。

如果 A 是稀疏矩阵, LU 因式分解通常包括行列排列, 即我们将 A 因式分解为

$$A = P_1 L U P_2$$

其中 $P_1, P_2 \in \mathbb{R}^{m \times m}$ 是排列矩阵。一个稀疏的矩形矩阵的 LU 因式分解可以非常有效地完成, 其计算成本比稠密矩阵低得多。LU 因式分解可以用于求解非方阵方程组。

假设 $A^T = PLU$ 是方程组 $A\mathbf{x} = \mathbf{b}, A \in \mathbb{R}^{m \times n}, m < n$ 中矩阵 A^T 的 LU 因式分解, 我们将 L 划分为

$$L = \begin{pmatrix} L_1 \\ L_2 \end{pmatrix}$$

其中 $L_1 \in \mathbb{R}^{m \times m}, L_2 \in \mathbb{R}^{(n-m) \times m}$, 容易验证参数化解为

$$\mathbf{x} = P \begin{pmatrix} -L_1^{-T} L_2^T \\ I \end{pmatrix} z + P(L_1^{-T} U^{-T} \mathbf{b})$$

其中 $z \in \mathbb{R}^{n-m}$

基于奇异值分解的非方阵系统求解方法 考虑非方阵系统

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m$$

我们可以计算 \mathbf{A} 的奇异值分解。设 $\mathbf{A} = \mathbf{U}\tilde{\Sigma}\mathbf{V}^T$, 记 $\tilde{\mathbf{x}} = \mathbf{V}^T\mathbf{x}$, $\tilde{\mathbf{b}} = \mathbf{U}^T\mathbf{b}$, 那么我们就得到了一个关于对角矩阵的方程组

$$\tilde{\Sigma}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$$

其中 $\tilde{\mathbf{b}}$ 是将右侧项进行旋转后的结果, 因为 $\tilde{\Sigma}$ 只有对角线有元素, 所以得到方程组

$$\begin{cases} \sigma_i \tilde{x}_i = \tilde{b}_i & i = 1, 2, \dots, r \\ 0 = \tilde{b}_i & i = r + 1, \dots, m \end{cases}$$

上述这个方程组是很容易计算的。但是同样可能会出现两种情况:

- 如果 $\tilde{\mathbf{b}}$ 最后 $m - r$ 个分量不为零。因为这 $m - r$ 个方程左边为 0, 所以方程组将无解。这种情况表明 \mathbf{b} 不在 \mathbf{A} 的列空间中。
- 而当 \mathbf{b} 在 \mathbf{A} 的列空间中, 那么最后 $m - r$ 个方程成立, 我们可以用前面 r 个方程进行求解, 即

$$\tilde{x}_i = \frac{\tilde{b}_i}{\sigma_i}, i = 1, \dots, r$$

$\tilde{\mathbf{x}}$ 中后 $n - r$ 个分量可以取任意值。如果 \mathbf{A} 是一个列满秩矩阵 (即他的零空间为 $\{\mathbf{0}\}$), 那么我们就会有唯一解。

我们一旦计算得到了 $\tilde{\mathbf{x}}$ 那么就可以通过 $\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$ 来得到方程的解。

5.1.4 敏度分析与其他方法

敏度分析问题引入 考虑如下两组线性方程组:

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix}, \quad \begin{bmatrix} 2 & 0 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix} \quad (5.19)$$

的解都是 $\mathbf{x} = (1, 1)^T$ 。

但是如果我们将方程的常数项做一点微小的变动, 求解方程组

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad \begin{bmatrix} 2 & 0 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad (5.20)$$

前者的解为 $\mathbf{x} = (2, 0)^T$, 而后者的解为 $\mathbf{x} = (1, \frac{10000}{10001})^T \approx (1, 0.9999)^T$ 。

可以看到左边方程的解变化的非常大, 而右边方程的解几乎没有变化。

在本节中, 我们将分析数据的小扰动对非奇异方阵线性方程解的影响。

输入的扰动敏感性

令 \mathbf{x} 为线性方程 $A\mathbf{x} = \mathbf{y}$ 的解, 其中 A 为非奇异方阵, 且 $\mathbf{y} \neq 0$ 。假设我们通过向它添加一个小的扰动项 $\Delta\mathbf{y}$ 来略微改变 \mathbf{y} , 并将 $\mathbf{x} + \Delta\mathbf{x}$ 称为扰动方程组的解:

$$A(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{y} + \Delta\mathbf{y}$$

我们的关键问题是: 如果 $\Delta\mathbf{y}$ 变小, $\Delta\mathbf{x}$ 将会不会变小? 我们从上面的公式看出, 并且从 $A\mathbf{x} = \mathbf{y}$ 的事实看, 扰动 $\Delta\mathbf{x}$ 本身就是线性方程组的解。

$$A\Delta\mathbf{x} = \Delta\mathbf{y}$$

并且, 由于认为 A 是可逆的, 我们可以写成

$$\Delta\mathbf{x} = A^{-1}\Delta\mathbf{y}$$

采用该方程两边的 2- 范数得出

$$\|\Delta\mathbf{x}\|_2 = \|A^{-1}\Delta\mathbf{y}\|_2 \leq \|A^{-1}\|_2 \|\Delta\mathbf{y}\|_2$$

其中 $\|A^{-1}\|_2$ 是 A^{-1} 的谱(最大奇异值)范数。类似地, 从 $A\mathbf{x} = \mathbf{y}$ 得出 $\|\mathbf{y}\|_2 = \|A\mathbf{x}\|_2 \leq \|A\|_2 \|\mathbf{x}\|_2$, 因此

$$\|\mathbf{x}\|_2^{-1} \leq \frac{\|A\|_2}{\|\mathbf{y}\|_2}$$

将上面两个公式相乘, 我们得到

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \|A^{-1}\|_2 \|A\|_2 \frac{\|\Delta\mathbf{y}\|_2}{\|\mathbf{y}\|_2}$$

这个结果是我们正在寻找的, 因为它将“输入项” \mathbf{y} 的相对变化与“输出” \mathbf{x} 的相对变化联系起来。

定义 5.1.7. 设 $A \in \mathbb{R}^{n \times n}$ 是可逆矩阵, 称数

$$\kappa(A) = \|A^{-1}\|_2 \|A\|_2,$$

是矩阵 A 的条件数。

设 σ_1, σ_n 分别是矩阵 A 的最大奇异值和最小奇异值, 那么

$$\|A\|_2 = \sigma_1, \quad \|A^{-1}\|_2 = 1/\sigma_n$$

因此矩阵 A 的条件数也可以定义为:

$$\kappa(A) = \frac{\sigma_1}{\sigma_n}, 1 \leq \kappa(A) \leq \infty.$$

大的 $\kappa(A)$ 意味着 \mathbf{b} 上的扰动可能导致 \mathbf{x} 上有很大的扰动, 即方程对输入数据的变化非常敏感。如果 A 是奇异的, 那么 $\kappa = \infty$ 。非常大的 $\kappa(A)$ 表明 A 接近奇异; 我们说在这种情况下 A 是病态的。

我们在以下引理中总结了我们的发现。

引理 5.1.1. (对于输入的敏感性) 令 A 为非奇异方阵, $x, \Delta x$ 满足

$$Ax = y$$

$$A(x + \Delta x) = y + \Delta y$$

然后它认为

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \kappa(A) \frac{\|\Delta y\|_2}{\|y\|_2}$$

其中 $\kappa(A) = \|A^{-1}\|_2 \|A\|_2$ 是矩阵 A 的条件数

系数矩阵中的扰动敏感性

接下来我们考虑 A 矩阵的扰动对 X 的影响。令 $Ax = y$ 并且令 ΔA 为一个扰动, 满足下面等式

$$(A + \Delta A)(x + \Delta x) = y, \quad \text{对于一些 } \Delta x$$

那么有

$$A\Delta x = -\Delta A(x + \Delta x)$$

因此 $\Delta x = -A^{-1}\Delta A(x + \Delta x)$ 。则

$$\|\Delta x\|_2 = \|A^{-1}\Delta A(x + \Delta x)\|_2 \leq \|A^{-1}\|_2 \|\Delta A\|_2 \|x + \Delta x\|_2$$

并且

$$\frac{\|\Delta x\|_2}{\|x + \Delta x\|_2} \leq \|A^{-1}\|_2 \|A\|_2 \frac{\|\Delta A\|_2}{\|A\|_2}$$

我们再次看到只有在条件数不是太大时, 小扰动 $\frac{\|\Delta A\|_2}{\|A\|_2} \ll 1$ 对 x 的相对影响才很小。就是说, 它离 1 不太远, $\kappa(A) \simeq 1$ 。这个会在下一个引理中总结。

引理 5.1.2. (系数矩阵中的扰动敏感性) 令 A 为非奇异方阵, $x, \Delta A, \Delta x$ 满足

$$Ax = y$$

$$(A + \Delta A)(x + \Delta x) = y$$

然后它认为

$$\frac{\|\Delta x\|_2}{\|x + \Delta x\|_2} \leq \kappa(A) \frac{\|\Delta A\|_2}{\|A\|_2}$$

对 A, y 联合扰动的敏感性

我们最后考虑了 A 和 y 的同时扰动对 x 的影响。令 $Ax = y$, 并且令 $\Delta A, \Delta y$ 为扰动, 满足下面等式

$$(A + \Delta A)(x + \Delta x) = y + \Delta y, \quad \text{对于一些 } \Delta x$$

然后, $A\Delta x = \Delta y - \Delta A(x + \Delta x)$, 因此 $\Delta x = A^{-1}\Delta y - A^{-1}\Delta A(x + \Delta x)$ 。则

$$\begin{aligned}\|\Delta x\|_2 &= \|A^{-1}\Delta y - A^{-1}\Delta A(x + \Delta x)\|_2 \\ &\leq \|A^{-1}\Delta y\|_2 + \|A^{-1}\Delta A(x + \Delta x)\|_2 \\ &\leq \|A^{-1}\|_2 \|\Delta y\|_2 + \|A^{-1}\| \|\Delta A\|_2 \|x + \Delta x\|_2\end{aligned}$$

接着, 上式除以 $\|x + \Delta x\|_2$,

$$\frac{\|\Delta x\|_2}{\|x + \Delta x\|_2} \leq \|A^{-1}\|_2 \frac{\|\Delta y\|_2}{\|y\|_2} \frac{\|y\|_2}{\|x + \Delta x\|_2} + \kappa(A) \frac{\|\Delta A\|_2}{\|A\|_2}$$

但是 $\|y\|_2 = \|Ax\|_2 \leq \|A\|_2 \|x\|_2$, 因此

$$\frac{\|\Delta x\|_2}{\|x + \Delta x\|_2} \leq \kappa(A) \frac{\|\Delta y\|_2}{\|y\|_2} \frac{\|x\|_2}{\|x + \Delta x\|_2} + \kappa(A) \frac{\|\Delta A\|_2}{\|A\|_2}$$

下一步, 我们根据 $\|x\|_2 = \|x + \Delta x - \Delta x\|_2 \leq \|x + \Delta x\|_2 + \|\Delta x\|_2$ 去写

$$\frac{\|\Delta x\|_2}{\|x + \Delta x\|_2} \leq \kappa(A) \frac{\|\Delta y\|_2}{\|y\|_2} \frac{\|x\|_2}{\|x + \Delta x\|_2} + \kappa(A) \frac{\|\Delta A\|_2}{\|A\|_2}$$

从中我们得到

$$\frac{\|\Delta x\|_2}{\|x + \Delta x\|_2} \leq \kappa(A) \frac{\|\Delta y\|_2}{\|y\|_2} \left(1 + \frac{\|\Delta x\|_2}{\|x + \Delta x\|_2}\right) + \kappa(A) \frac{\|\Delta A\|_2}{\|A\|_2}$$

因此

$$\frac{\|\Delta x\|_2}{\|x + \Delta x\|_2} \leq \frac{\kappa(A)}{1 - \kappa(A) \frac{\|\Delta y\|_2}{\|y\|_2}} \left(\frac{\|\Delta y\|_2}{\|y\|_2} + \frac{\|\Delta A\|_2}{\|A\|_2}\right)$$

扰动的“放大因子”是受 $\frac{\kappa(A)}{1 - \kappa(A) \frac{\|\Delta y\|_2}{\|y\|_2}}$ 的约束。因此, 该界限小于某些给定的 γ , 如果

$$\kappa(A) \leq \frac{\gamma}{1 + \gamma \frac{\|\Delta y\|_2}{\|y\|_2}}$$

因此, 我们看到关节扰动的影响仍然由 A 的条件数控制, 如下所述

引理 5.1.3. (对 A, y 扰动的敏感性) 令 A 为非奇异方阵, 令 $\gamma > 1$ 已知, 并且令 $x, \Delta y, \Delta A, \Delta x$ 满足下面等式

$$Ax = y$$

$$(A + \Delta A)(x + \Delta x) = y + \Delta y$$

然后

$$\kappa(A) \leq \frac{\gamma}{1 + \gamma \frac{\|\Delta y\|_2}{\|y\|_2}}$$

这意味着

$$\frac{\|\Delta x\|_2}{\|x + \Delta x\|_2} \leq \gamma \left(\frac{\|\Delta y\|_2}{\|y\|_2} + \frac{\|\Delta A\|_2}{\|A\|_2}\right)$$

5.2 最小二乘问题

最小二乘 (Least Squares, LS) 法起源于 18 世纪天文学和测地学的应用需要：有一组容易观测的量和一组不易观测的量，它们之间满足线性关系，如何根据易观测数据去估计不易观测的量 (它们称为模型的参数)。在计算上主要涉及超定线性方程组的求解。

最小二乘问题的历史可以追溯到欧拉 (L. Euler) 在 1749 年研究木星对土星轨道的影响时，得到 $n = 75$ 和 $k = 8$ 的一组方程，欧拉用方程分组的思想求解此方程。梅耶 (J. T. Mayer) 在 1750 年由确定地球上一点的经度问题，得到 $n = 27$ 和 $k = 3$ 的一组方程，也用方程分组的思想求解。勒让德 (A. M. Legendre) 于 1805 年在其著作《计算慧星轨道的新方法》中首次提出了最小二乘法。高斯则于 1809 年他的著作《天体运动论》中发表了最小二乘法的方法。

本节的重点是如何通过解决最小二乘问题来解决超定方程组和欠定方程组。

5.2.1 最小二乘问题与线性回归

最小二乘问题多产生于线性回归或者数据拟合问题。比如给定平面上 m 个点 $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ 其中 $x_i \in \mathbb{R}$ 是输入 X 的观测值， $y_i \in \mathbb{R}$ 是输出 Y 的观测值。我们要求给出一条直线 $y = kx + b$ ， k, b 是直线的参数， $k, b \in \mathbb{R}$ ，使得在所有输入观测值 x_i 上， $y = kx_i + b$ 能最佳地逼近这些输出观测值 y_i ，也即使得输出观测值 y_i 与直线所预测的值的残差 $r(x_i; k, b) = y_i - y = y_i - (kx_i + b)$ 的平方和最小，即

$$\min_{k, b} \sum_{i=1}^m (y_i - (kx_i + b))^2 = \min_{k, b} \sum_{i=1}^m (r(x_i; k, b))^2.$$

这样实际上就是一个求解线性回归的参数 k, b 的问题。

在高维情况下，我们用一个超平面来拟合数据点 (在三维情况下就是用一个平面来拟合)。

我们用 $y = \mathbf{w}^T \mathbf{x} + b$ 来表示超平面，其中 $\mathbf{x} \in \mathbb{R}^n$ ， n 是输入 X 的特征数， $\mathbf{w} \in \mathbb{R}^n$ 是超平面预测函数中特征的权重向量参数， $b \in \mathbb{R}$ 是偏差参数。希望所有输出观测值 y_i 与预测函数的预测值 $y(\mathbf{x}_i; \mathbf{w}, b)$ 的残差 $r(\mathbf{x}_i; \mathbf{w}, b) = y_i - y = y_i - (\mathbf{w}^T \mathbf{x}_i + b)$ 尽可能的小。记 \mathbf{x}_i 的第 j 个特征分量为 x_{ij} ，残差向量 $\mathbf{r} \in \mathbb{R}^m$ 的第 i 个分量为 $r(\mathbf{x}_i; \mathbf{w}, b)$ ：

$$\mathbf{r} = \begin{pmatrix} y_1 - (w_1 x_{11} + w_2 x_{12} + \cdots + w_n x_{1n} + b) \\ y_2 - (w_1 x_{21} + w_2 x_{22} + \cdots + w_n x_{2n} + b) \\ y_3 - (w_1 x_{31} + w_2 x_{32} + \cdots + w_n x_{3n} + b) \\ \vdots \\ y_m - (w_1 x_{m1} + w_2 x_{m2} + \cdots + w_n x_{mn} + b) \end{pmatrix}$$

化成矩阵的形式表示为

$$\mathbf{r} = \mathbf{y} - \mathbf{A}\hat{\mathbf{w}},$$

$$\text{其中: } A = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} & 1 \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} & 1 \\ x_{31} & x_{32} & x_{33} & \dots & x_{3n} & 1 \\ \vdots & & & & & \\ x_{m1} & x_{m2} & x_{m3} & \dots & x_{mn} & 1 \end{pmatrix}, \hat{\mathbf{w}} = \begin{pmatrix} \mathbf{w} \\ b \end{pmatrix}, \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{pmatrix}$$

问题就变为求参数 $\hat{\mathbf{w}}$, 使得残差 \mathbf{r} 尽可能的小。若使残差 \mathbf{r} 在 2 范数意义下最小, 也即,

$$\arg \min_{\hat{\mathbf{w}}} \|A\hat{\mathbf{w}} - \mathbf{y}\|_2$$

把上式中的符号调整成我们常用的符号:

$$\arg \min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2 \quad (5.21)$$

这就是最小二乘问题。

下面给出最小二乘问题的定义。

定义 5.2.1. 给定矩阵 $A \in \mathbb{R}^{m \times n}$ 和向量 $\mathbf{b} \in \mathbb{R}^m$, 确定 $\mathbf{x} \in \mathbb{R}^n$ 使得

$$\|\mathbf{b} - A\mathbf{x}\|_2 = \|\mathbf{r}(\mathbf{x})\|_2 = \min_{\mathbf{y} \in \mathbb{R}^n} \|\mathbf{r}(\mathbf{y})\|_2 = \min_{\mathbf{y} \in \mathbb{R}^n} \|A\mathbf{y} - \mathbf{b}\|_2 \quad (5.22)$$

其中 $\mathbf{r}(\mathbf{x}) = \mathbf{b} - A\mathbf{x}$, 称为残差向量, 该问题称为最小二乘问题, 简记为 *LS(Least Squares)* 问题, 而 \mathbf{x}_0 则称为最小二乘解或极小解。

如果残差向量 \mathbf{r} 线性依赖于 \mathbf{x} , 则称其为线性最小二乘问题; 如果 \mathbf{r} 非线性的依赖于 \mathbf{x} , 则称其为非线性最小二乘问题。我们主要讨论线性最小二乘问题, 简称最小二乘问题。所有最小二乘解的集合记为 \mathcal{X}_{LS} , 即

$$\mathcal{X}_{LS} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \text{ 满足(5.22)}\}.$$

解集 \mathcal{X}_{LS} 中 2 范数最小的解称为最小范数解, 记为 \mathbf{x}_{LS} , 即

$$\|\mathbf{x}_{LS}\|_2 = \min\{\|\mathbf{x}\|_2 : \mathbf{x} \in \mathcal{X}_{LS}\}.$$

对于残差向量选择不同的范数, 便得到不同的问题, 我们主要讨论残差向量选择 2 范数的情况。

下面给出关于列满秩或行满秩矩阵的 2 条有用的性质:

定理 5.2.1. 1. $A \in \mathbb{R}^{m \times n}$ 是一个列满秩矩阵 (*i.e.*, $\text{rank}(A) = n$) 当且仅当 $A^T A$ 是可逆的。

2. $A \in \mathbb{R}^{m \times n}$ 是一个行满秩矩阵 (*i.e.*, $\text{rank}(A) = m$) 当且仅当 $A A^T$ 是可逆的。

证明. 对于第 1 条性质: 如果 $A^T A$ 不是可逆的, 则存在 $\mathbf{x} \neq 0$ 使得 $A^T A \mathbf{x} = 0$. $\mathbf{x}^T A^T A \mathbf{x} = 0$, 因此 $A \mathbf{x} = 0$. 所以 A 不是一个列满秩矩阵. 反之, 如果 $A^T A$ 是可逆的, 对于每个 $\mathbf{x} \neq 0$, $A^T A \mathbf{x} \neq 0$, 也能推出对于每一个非零的 \mathbf{x} , $A \mathbf{x} \neq 0$. 第 2 条性质的证明过程与第 1 条的证明过程相似. \square

最小二乘问题的解 \mathbf{x} 又称为线性方程组(5.23)的最小二乘解。

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{A} \in \mathbb{R}^{m \times n} \quad (5.23)$$

即残差向量 $\mathbf{r}(\mathbf{x})$ 的 2 范数最小的意义下满足方程组(5.23)。

根据 m 与 n 以及矩阵 \mathbf{A} 的秩 $r(\mathbf{A})$ 的不同, 最小二乘问题可分为下面几种情况:

(1) $m = n$

$$(1a) r(\mathbf{A}) = m = n$$

$$(1b) r(\mathbf{A}) < m = n$$

(2) $m > n$

$$(2a) r(\mathbf{A}) = n < m$$

$$(2b) r(\mathbf{A}) < n < m$$

(3) $m < n$

$$(3a) r(\mathbf{A}) = m < n$$

$$(3b) r(\mathbf{A}) < m < n$$

每一种情形, 根据矩阵 \mathbf{A} 的列是线性无关或线性相关, 也即矩阵 \mathbf{A} 为列满秩或秩亏的, 又可分为两种情形: 满秩最小二乘问题或秩亏最小二乘问题。

最小范数解与最小范数最小二乘解 (1) 当方程组(??)有解时, 显然也满足最小二乘问题(??), 如何确定 $\mathbf{x}_0 \in \mathbb{R}^n$, 使得

$$\|\mathbf{x}_0\|_2 = \min_{\mathbf{A}\mathbf{x}=\mathbf{b}} \|\mathbf{x}\|_2$$

称这样的 \mathbf{x}_0 为方程组(??)的最小范数解 (特别对于欠定情形, 方程组有无穷多解, 我们总是对具有最小 2 范数的解感兴趣)。

(2) 当方程组(??)无解时, 此时相应 LS 问题的最小二乘解不是方程组 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 的解, 如何确定 $\mathbf{x}_0 \in \mathbb{R}^n$, 使得

$$\|\mathbf{x}_0\|_2 = \min_{\min \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2} \|\mathbf{x}\|_2$$

称这样的 \mathbf{x}_0 为方程组(??)的最小范数最小二乘解 (方程组无解时相应的 LS 问题的最小二乘解可以看成方程组的近似解, 我们总是对使得 2 范数最小的近似解感兴趣)。

矩阵的广义逆是研究一般线性方程组最小范数解和最小范数最小二乘解的强有力工具。

定理 5.2.2. 如果方程组 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 有解, 则它的最小范数解 \mathbf{x}_0 唯一, 并且 $\mathbf{x}_0 = \mathbf{A}^\dagger \mathbf{b}$.

定理 5.2.3. 如果线性方程组 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 无解, 则它的最小范数最小二乘解 \mathbf{x}_0 唯一, 并 $\mathbf{x}_0 = \mathbf{A}^\dagger \mathbf{b}$.

考虑一类具体的方程组, 针对欠定方程组的情形: 当矩阵 \mathbf{A} 的列数比行数多: $m < n$ 。

- 假设矩阵 \mathbf{A} 是行满秩, 我们有 $\dim \{\text{Null}(\mathbf{A})\} = n - m > 0$, 因此得出 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 有无数个解
并且解的集合是 $\mathbb{S}_{\mathbf{x}} = \{\mathbf{x} : \mathbf{x} = \tilde{\mathbf{x}} + \mathbf{z}, \mathbf{z} \in \text{Null}(\mathbf{A})\}$, 其中 $\tilde{\mathbf{x}}$ 是任意满足 $\mathbf{A}\tilde{\mathbf{x}} = \mathbf{b}$ 的向量。

- 我们想从这个解的集合 \mathbb{S}_x 中挑选出一个 2 范数最小的解 \mathbf{x}^* 。也即求解:

$$\min_{\mathbf{x}: \mathbf{A}\mathbf{x}=\mathbf{b}} \|\mathbf{x}\|_2$$

这个式子等价于 $\min_{\mathbf{x} \in \mathbb{S}_x} \|\mathbf{x}\|_2$.

- 因为 (唯一的) 解 \mathbf{x}^* 必须与 $\text{Null}(\mathbf{A})$ 相互垂直, 等价地, $\mathbf{x}^* \in \text{Col}(\mathbf{A}^T)$, 这意味着存在 ζ , 使得 $\mathbf{x}^* = \mathbf{A}^T \zeta$ 。因为 \mathbf{x}^* 是方程组的解, 必须满足 $\mathbf{A}\mathbf{x}^* = \mathbf{b}$, 所以有 $\mathbf{A}\mathbf{A}^T \zeta = \mathbf{b}$.
- 因为矩阵 \mathbf{A} 是行满秩, $\mathbf{A}\mathbf{A}^T$ 是可逆的并且有唯一的 ζ 是方程组的解, 所以有 $\zeta = (\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b}$.

这样我们得到了唯一的最小范数解:

$$\mathbf{x}^* = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{b}. \quad (5.24)$$

因为 $\mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1}$ 正是 \mathbf{A} 是行满秩矩阵时的伪逆 \mathbf{A}^\dagger , 所以

$$\mathbf{x}^* = \mathbf{A}^\dagger \mathbf{b}$$

定理 5.2.4. 设 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m \leq n$ 是行满秩的, 并且令 $\mathbf{b} \in \mathbb{R}^m$ 。在线性方程组 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 的所有解中, 存在唯一的 2 范数最小的解, 这个解由(5.24)给出。

为了说明最小二程问题解的存在性, 我们验证如下的定理。

定理 5.2.5. 线性最小二乘问题(5.22)的解总是存在的, 而且其解唯一的充分必要条件是 $\text{nullity}(\mathbf{A}) = 0$.

证明. 因为 $\mathbb{R}^m = \text{Col}(\mathbf{A}) \oplus \text{Col}(\mathbf{A})^\perp$, 所以向量 \mathbf{b} 可以唯一地表示为 $\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2$, 其中 $\mathbf{b}_1 \in \text{Col}(\mathbf{A})$, $\mathbf{b}_2 \in \text{Col}(\mathbf{A})^\perp$. 于是对于任意 $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b}_1 - \mathbf{A}\mathbf{x} \in \text{Col}(\mathbf{A})$ 且与 \mathbf{b}_2 正交, 从而

$$\|\mathbf{r}(\mathbf{x})\|_2^2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 = \|(\mathbf{b}_1 - \mathbf{A}\mathbf{x}) + \mathbf{b}_2\|_2^2 = \|\mathbf{b}_1 - \mathbf{A}\mathbf{x}\|_2^2 + \|\mathbf{b}_2\|_2^2$$

由此即知, $\|\mathbf{r}(\mathbf{x})\|_2^2$ 达到极小当且仅当 $\|\mathbf{b}_1 - \mathbf{A}\mathbf{x}\|_2^2$ 达到极小;

而 $\mathbf{b}_1 \in \text{Col}(\mathbf{A})$ 又蕴涵着 $\|\mathbf{b}_1 - \mathbf{A}\mathbf{x}\|_2^2$ 达到极小的充分与必要条件是

$$\mathbf{A}\mathbf{x} = \mathbf{b}_1$$

这样, 由 $\mathbf{b}_1 \in \text{Col}(\mathbf{A})$ 和 $\mathbf{A}\mathbf{x} = \mathbf{b}_1$ 有唯一解的充要条件是 $\text{nullity}(\mathbf{A}) = 0$ 。

立即推出定理的结论成立。 \square

下面这个定理则给出了求解最小二程问题的方法。

定理 5.2.6. $\mathbf{x} \in \mathcal{X}_{LS}$ 当且仅当

$$\mathbf{A}^T \mathbf{A}\mathbf{x} = \mathbf{A}^T \mathbf{b} \quad (5.25)$$

其中方程组(5.25)称为最小二乘问题的正规化方程组或法方程组。

证明. 设 $x \in \mathcal{X}_{LS}$. 由定理 5.2.5 证明知 $\mathbf{A}x = \mathbf{b}_1$, 其中 $\mathbf{b}_1 \in \text{Col}(\mathbf{A})$, 而且

$$\mathbf{r}(x) = \mathbf{b} - \mathbf{A}x = \mathbf{b} - \mathbf{b}_1 = \mathbf{b}_2 \in \text{Col}(\mathbf{A})^\perp$$

因而 $\mathbf{A}^T \mathbf{r}(x) = \mathbf{A}^T \mathbf{b}_2 = 0$. 将 $\mathbf{r}(x) = \mathbf{b} - \mathbf{A}x$ 代入 $\mathbf{A}^T \mathbf{r}(x) = 0$ 即得(5.25). 反之, 设 $x \in \mathbb{R}^n$ 满足 $\mathbf{A}^T \mathbf{A}x = \mathbf{A}^T \mathbf{b}$, 则对任意的 $z \in \mathbb{R}^n$ 有

$$\begin{aligned}\|\mathbf{b} - \mathbf{A}(x + z)\|_2^2 &= \|\mathbf{b} - \mathbf{A}x\|_2^2 - 2z^T \mathbf{A}^T (\mathbf{b} - \mathbf{A}x) + \|Az\|_2^2 \\ &= \|\mathbf{b} - \mathbf{A}x\|_2^2 + \|Az\|_2^2 \geq \|\mathbf{b} - \mathbf{A}x\|_2^2\end{aligned}$$

由此即得 $x \in \mathcal{X}_{LS}$. □

由定理 5.2.6 可知, 可以通过求解正规化方程组或法方程组 $\mathbf{A}^T \mathbf{A}x = \mathbf{A}^T \mathbf{b}$ 来求解 $\mathbf{A}x = \mathbf{b}$ 的最小二乘解。如果 $\mathbf{A}^T \mathbf{A}$ 可逆, 那么最小二乘解为 $x = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ 。

推论 5.2.1. 若矩阵 \mathbf{A} 列满秩, 则线性最小二乘问题(5.22)的解是唯一的, 并且解为

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} = \mathbf{A}^\dagger \mathbf{b},$$

其中 $\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ 。

注记 11. 关于正规化方程组的一点注解 对于最优化问题, 正规化方程只不过是在求最优化问题:

$$\min_x f(x)$$

的解, 其中 $f(x) = \|\mathbf{A}x - \mathbf{b}\|_2^2$. 稍后我们将知道, 如果目标函数是可微, 具有凸性并且问题是没有约束的, 最优点由条件 $\nabla f(x) = 0$ 决定。在我们这种情况下, 函数 f 在一点 x 处的梯度是非常容易看出是 $\nabla f(x) = \mathbf{A}^T(\mathbf{A}x - \mathbf{b})$ 。

列满秩和行满秩以外的情况 如果 \mathbf{A} 既不是列满秩也不是行满秩, 它的最小二乘解仍是方程:

$$\mathbf{A}^T \mathbf{A}x = \mathbf{A}^T \mathbf{b}$$

的解, 但是, $\mathbf{A}^T \mathbf{A}$ 虽然是方阵, 却并不一定可逆。

然而, 这个方程是一定有解的, 我们总可以通过初等行变换将其化为行满秩的方程组。

这样我们就可以利用求解欠定问题的最小范数解的方法, 求得方程的最小范数最小二乘解。

实际上, 可以通过 SVD 的方法, 求得最小范数最小二乘解, 这里不展开介绍。

最小二乘解的解释 根据应用的场景不同, 可以给出最小二乘问题几种不同的解释。

- **线性方程组的近似解:** 最小二乘问题的解, 是使得残差 $\mathbf{r} = \mathbf{A}x - \mathbf{b}$ 在 2 范数意义下最小的解。
- **在 $\text{Col}(\mathbf{A})$ 上的投影:** 最小二乘问题的解, 是 \mathbf{b} 在 $\text{Col}(\mathbf{A})$ 上的投影。

- **线性回归模型:** 最小二乘问题的解, 是线性回归模型 $f(\mathbf{a}_i) = \mathbf{x}^T \mathbf{a}_i$ 使得 $f(\mathbf{a}_i) \approx y_i$, 求解出的参数 \mathbf{x} 。数据集表示为 $m \times (n+1)$ 大小的矩阵 \mathbf{A} , 每一行对应一个实例, 前 n 项对应实例的 n 个特征, 最后一项为 1。 \mathbf{b} 是 m 维向量, 每一行对应 \mathbf{x}_i 是观测值。
- **最小程度地干扰可行性:** 最小二乘问题的解, 是使得 $\mathbf{Ax} = \mathbf{b}$ 右侧添加在 2 范数意义下的最小扰动项 $\delta\mathbf{b}$ 后 $\mathbf{Ax} = \mathbf{b} + \delta\mathbf{b}$ 有解时的方程组的解。
- **最好的线性无偏估计:** 在统计估计的背景下, 线性模型的最好无偏估计与最小二乘问题的解是一致的。

5.2.2 最小二乘问题的求解方法

最小二乘问题按照矩阵 \mathbf{A} 是否满秩, 可分为满秩最小二乘问题和秩亏最小二乘问题。本小节我们讨论在 \mathbf{A} 为列满秩的情形下超定方程组

$$\mathbf{Ax} = \mathbf{b}$$

的最小二乘解的求解方法: 此时, $\mathbf{A}^T \mathbf{A}$ 可逆, 我们的目标是求出方程组唯一的最小二乘解。

对于秩亏最小二乘问题的求解, 我们并不涉及。

正规化法

正规化方法 (Cholesky 分解法) 方程组 $\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$ 称为最小二乘问题的正规化方程组或法方程组, 这是一个含有 n 个变量 n 个方程的线性方程组。在 \mathbf{A} 的列向量线性无关的条件下, $\mathbf{A}^T \mathbf{A}$ 对称正定, 故可用平方根法求解(5.25). 这样, 我们就得到了求解最小二乘问题最古老的算法—正规化方法, 其基本步骤如下:

1. 计算 $\mathbf{C} = \mathbf{A}^T \mathbf{A}, \mathbf{d} = \mathbf{A}^T \mathbf{b}$;
2. 用平方根法计算 \mathbf{C} 的 Cholesky 分解: $\mathbf{C} = \mathbf{LL}^T$;
3. 求解三角方程组 $\mathbf{L}\mathbf{b} = \mathbf{d}$ 和 $\mathbf{L}^T \mathbf{x} = \mathbf{b}$.

注意, 正规化方程组 $\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$ 的解 \mathbf{x} 可以表为

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} = \mathbf{A}^\dagger \mathbf{b}$$

例 5.2.1. 利用正规化方法求 $\mathbf{Ax} = \mathbf{b}$ 得最小二乘解, 其中

$$\mathbf{A} = \begin{pmatrix} 1 & 4 & 5 \\ 1 & -2 & 3 \\ 1 & 4 & 1 \\ 1 & -2 & -1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 6 \\ 0 \\ -4 \\ 2 \end{pmatrix}$$

解.

$$\mathbf{A}^T \mathbf{A} = \begin{pmatrix} 4 & 4 & 8 \\ 4 & 40 & 20 \\ 8 & 20 & 36 \end{pmatrix}, \mathbf{A}^T \mathbf{b} = \begin{pmatrix} 4 \\ 4 \\ 24 \end{pmatrix}$$

对 $\mathbf{A}^T \mathbf{A}$ 做 Cholesky 分解:

$$\mathbf{A}^T \mathbf{A} = \begin{pmatrix} 2 & 0 & 0 \\ 2 & 6 & 0 \\ 4 & 2 & 4 \end{pmatrix} \begin{pmatrix} 2 & 2 & 4 \\ 0 & 6 & 2 \\ 0 & 0 & 4 \end{pmatrix}$$

解方程

$$\begin{pmatrix} 2 & 0 & 0 \\ 2 & 6 & 0 \\ 4 & 2 & 4 \end{pmatrix} \mathbf{y} = \begin{pmatrix} 4 \\ 4 \\ 24 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 2 \\ 0 \\ 4 \end{pmatrix}$$

再解方程

$$\begin{pmatrix} 2 & 2 & 4 \\ 0 & 6 & 2 \\ 0 & 0 & 4 \end{pmatrix} \mathbf{x}^* = \begin{pmatrix} 2 \\ 0 \\ 4 \end{pmatrix} \quad \text{得 } \mathbf{x}^* = \begin{pmatrix} -2/3 \\ -1/3 \\ 1 \end{pmatrix}$$

QR 分解法

由 2-范数的正交不变性, 即若 \mathbf{Q} 是正交矩阵, $\|\mathbf{Q}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$ 。

可以使用 QR 分解求解最小二乘问题。对于 $\mathbf{A}^{m \times n}$ 的列满秩矩阵, 其 QR 分解后

$$\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R}^{n \times n} \\ \mathbf{0}^{(m-n) \times n} \end{pmatrix},$$

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 = \left\| \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix} \mathbf{x} - \mathbf{Q}^T \mathbf{b} \right\|_2 = \left\| \begin{pmatrix} \mathbf{R}\mathbf{x} \\ \mathbf{0} \end{pmatrix} - \mathbf{Q}^T \mathbf{b} \right\|_2$$

我们把 $\mathbf{Q}^T \mathbf{b}$ 拆成 $\begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}$, 其中 \mathbf{b}_1 是 $\mathbf{Q}^T \mathbf{b}$ 的前 n 项, \mathbf{b}_2 是 $\mathbf{Q}^T \mathbf{b}$ 的后 $m - n$ 项。那么

$$\arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \arg \min_{\mathbf{x}} (\|\mathbf{R}\mathbf{x} - \mathbf{b}_1\|_2^2 + \|\mathbf{b}_2\|_2^2) = \arg \min_{\mathbf{x}} \|\mathbf{R}\mathbf{x} - \mathbf{b}_1\|_2^2$$

通过之前最小二乘问题和方程组的关系, 我们只需要求 $\mathbf{R}\mathbf{x} = \mathbf{b}_1$ 的解即可。

QR 分解法求解最小二乘问题的基本步骤如下:

(1) 计算 \mathbf{A} 的 QR 分解: $\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$;

(2) 计算 $\mathbf{b}_1 = (\mathbf{Q}^T \mathbf{b})[1 : n]$;

(3) 求解上三角方程组 $Rx = b_1$ 。

在矩阵分解部分，我们介绍过 Gram-Schmidt 正交化、Householder 变换、Givens 变换三种方法进行 QR 分解。

在计算机中一般使用基于 Householder 变换的 QR 分解，该算法有良好的数值性质，结果通常要比正规化方法精确。但是运算量也比较大，大约为 $2mn^2 - \frac{2}{3}n^3$ 。

我们也可以使用 Givens 变换来实现 QR 分解，所需的运算量大约是 Householder 方法的两倍，但是如果 A 有较多的零元素，则灵活地使用 Givens 变换会使运算量大为减少。

例 5.2.2. 利用 QR 分解求 $Ax = b$ 得最小二乘解，其中

$$A = \begin{pmatrix} 1 & 4 & 5 \\ 1 & -2 & 3 \\ 1 & 4 & 1 \\ 1 & -2 & -1 \end{pmatrix}, b = \begin{pmatrix} 6 \\ 0 \\ -4 \\ 2 \end{pmatrix}$$

解. 求矩阵 A 的 QR 分解

$$A = Q \begin{pmatrix} R \\ O \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 & -1/2 \\ 1/2 & 1/2 & -1/2 & -1/2 \\ 1/2 & -1/2 & -1/2 & 1/2 \end{pmatrix} \begin{pmatrix} 2 & 2 & 4 \\ 0 & 6 & 2 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{pmatrix}$$

$$Q = \begin{pmatrix} 1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 & -1/2 \\ 1/2 & 1/2 & -1/2 & -1/2 \\ 1/2 & -1/2 & -1/2 & 1/2 \end{pmatrix}, R = \begin{pmatrix} 2 & 2 & 4 \\ 0 & 6 & 2 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{pmatrix}, Q^T b = \begin{pmatrix} 2 \\ 0 \\ 4 \\ 6 \end{pmatrix}, b^* = \begin{pmatrix} 2 \\ 0 \\ 4 \\ 6 \end{pmatrix}$$

解方程

$$Rx^* = b^*$$

$$x^* = \begin{pmatrix} -2/3 \\ -1/3 \\ 1 \end{pmatrix}$$

奇异值分解法

也可以使用奇异值分解来解决最小二乘问题。设 $A \in \mathbb{R}^{m \times n}$ ($m \geq n$) 列满秩, $A = U \begin{bmatrix} \Sigma \\ O \end{bmatrix} V^T$

是 A 的奇异值分解，令 U_n 为 U 的前 n 列组成的矩阵，即 $U = [U_n, \tilde{U}]$ ，其中 U 是正交矩阵，根据 2 范数的正交不变性得

$$\begin{aligned}
 \|Ax - b\|_2^2 &= \left\| U \begin{bmatrix} \Sigma \\ O \end{bmatrix} V^T x - b \right\|_2^2 = \left\| \begin{bmatrix} \Sigma \\ O \end{bmatrix} V^T x - \begin{bmatrix} U_n^T \\ \tilde{U}^T \end{bmatrix} b \right\|_2^2 \\
 &= \left\| \begin{bmatrix} \Sigma V^T x - U_n^T b \\ -\tilde{U}^T b \end{bmatrix} \right\|_2^2 = \|\Sigma V^T x - U_n^T b\|_2^2 + \|\tilde{U}^T b\|_2^2 \\
 &\geq \|\tilde{U}^T b\|_2^2
 \end{aligned}$$

等号当且仅当 $\Sigma V^T x - U_n^T b = 0$ 时成立, 即

$$x = (\Sigma V^T)^{-1} U_n^T b = V \Sigma^{-1} U_n^T b$$

例 5.2.3. 利用 SVD 分解求 $Ax = b$ 得最小二乘解, 其中

$$A = \begin{pmatrix} 1 & 4 & 5 \\ 1 & -2 & 3 \\ 1 & 4 & 1 \\ 1 & -2 & -1 \end{pmatrix}, b = \begin{pmatrix} 6 \\ 0 \\ -4 \\ 2 \end{pmatrix}$$

解. 这里, 我们直接使用 Matlab 计算 A 的 SVD, 并利用

$$x = V \Sigma^{-1} U_n^T b$$

求解。

```

>> [U, Sigma, V]=svd(A)                                V =
U =
-0.8321    0.2178    0.1010    0.5000    -0.1495    0.1952   -0.9693
-0.0929    0.8517   -0.1263   -0.5000   -0.7252   -0.6880   -0.0267
-0.4831   -0.4314   -0.5749   -0.5000   -0.6721    0.6990    0.2444
  0.2561    0.2025   -0.8021    0.5000

>> V*diag(1./diag(Sigma))*U(:,1:3)'*b

Sigma =
  7.7046      0      0      -0.6667
      0    4.3066      0      -0.3333
      0      0    1.4466      1.0000
      0      0      0
ans =

```

也可以在 Python 中使用 NumPy 库中的 `linalg.svd()` 函数, 计算 SVD, 注意这个函数返回的是 U, Σ, V^T 。

```
In [1]: import numpy as np
In [2]: A = np.matrix("1,4,5;1,-2,3;1,4,1;1,-2,-1")
In [3]: b = np.matrix("6;0;-4;2")
In [4]: U,Sigma,Vt = np.linalg.svd(A)
In [5]: V=Vt.T
In [6]: print(U)
[[ -0.83208589  0.21778105  0.10101724  0.5
[-0.0928533   0.8517268  -0.12625113 -0.5
[-0.48314949 -0.43140465 -0.57485354 -0.5
[ 0.2560811   0.2025411  -0.80212192  0.5
In [7]: print(np.diag(Sigma))
[[ 7.70455139  0.          0.          ]
[ 0.          4.3066429  0.          ]
[ 0.          0.          1.44662184]
In [8]: print(V)
[[ -0.14952325  0.19519712 -0.96929917
[-0.72520681 -0.68801391 -0.02668222
[-0.67209961  0.69895275  0.24443236]
In [9]: V*np.diag(1./Sigma)*U[:,0:3].T*b
Out[9]: matrix([[-0.66666667,
[-0.33333333],
[ 1.        ]])
```

5.2.3 最小二乘问题的变体

对基本的最小二乘问题做一些修改，会得到其他形式的最小二乘问题。

加权最小二乘

在普通的最小二乘法中，我们想要最小化误差向量各项的平方和：

$$\|\mathbf{Ax} - \mathbf{y}\|_2^2 = \sum_{i=1}^m r_i^2, \quad r_i = \mathbf{a}_i^\top \mathbf{x} - y_i$$

其中 $\mathbf{a}_i^\top, i = 1, \dots, m$ 是 \mathbf{A} 的各列。但是，在某些情形下，方程的残差项并不是同样重要的，相比其他方程，有可能满足某一个方程更重要，这样，我们需要在残差项赋予权重：

$$f_0(\mathbf{x}) = \sum_{i=1}^m w_i^2 r_i^2,$$

其中 $w_i \geq 0$ 是给定的权重。

这样最小化目标函数重写为：

$$f_0(\mathbf{x}) = \|\mathbf{W}(\mathbf{Ax} - \mathbf{y})\|_2^2 = \|\mathbf{A}_w \mathbf{x} - \mathbf{y}_w\|_2^2$$

其中

$$\mathbf{W} = \text{diag}(w_1, \dots, w_m), \mathbf{A}_w \doteq \mathbf{WA}, \mathbf{y}_w \doteq \mathbf{Wy}$$

加权最小二乘仍然是普通最小二乘的形式，其权重最小二乘解为：

$$\begin{aligned} \hat{\mathbf{x}}_{\text{WLS}} &= (\mathbf{A}_w^\top \mathbf{A}_w)^{-1} \mathbf{A}_w^\top \mathbf{y}_w \\ &= (\mathbf{A}^\top \mathbf{W}^\top \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y} \end{aligned}$$

约束最小二乘

考虑带有约束的最小二乘问题

$$\begin{aligned} \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 \\ \text{s.t. } \mathbf{Bx} = \mathbf{f} \end{aligned}$$

其中 $\mathbf{Bx} = \mathbf{f}$ 是约束条件。求解需要凸优化知识，在这里只列出解。如果 $\mathbf{A}^\top \mathbf{A}$ 非奇异，且 \mathbf{B} 行满秩，则

$$\mathbf{x} = (\mathbf{A}^\top \mathbf{A})^{-1}(\mathbf{A}^\top \mathbf{b} - \mathbf{B}^\top \boldsymbol{\lambda})$$

其中 $\boldsymbol{\lambda} = [\mathbf{B}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{B}^\top]^{-1} [\mathbf{B}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b} - \mathbf{f}]$.

总体最小二乘

考虑得到的数据矩阵和数据向量 \mathbf{A}, \mathbf{b} 都有误差，设实际观测的数据矩阵和数据向量

$$\mathbf{A} = \mathbf{A}_0 + \mathbf{E}, \quad \mathbf{b} = \mathbf{b}_0 + \mathbf{e}$$

其中 \mathbf{E} 和 \mathbf{e} 分别表示误差数据矩阵和误差数据向量。总体最小二乘的基本思想是：不仅用校正向量 $\Delta \mathbf{b}$ 去干扰数据向量 \mathbf{b} ，同时用校正矩阵 $\Delta \mathbf{A}$ 去干扰数据矩阵 \mathbf{A} ，以便对 \mathbf{A} 和 \mathbf{b} 二者内存在的误差或噪声进行联合补偿

$$\mathbf{b} + \Delta \mathbf{b} = \mathbf{b}_0 + \mathbf{e} + \Delta \mathbf{b} \rightarrow \mathbf{b}_0$$

$$\mathbf{A} + \Delta \mathbf{A} = \mathbf{A}_0 + \mathbf{E} + \Delta \mathbf{A} \rightarrow \mathbf{A}_0$$

以抑制观测误差或噪声对矩阵方程求解的影响，从而实现从有误差的矩阵方程到精确矩阵方程的求解的转换

$$(\mathbf{A} + \Delta \mathbf{A})\mathbf{x} = \mathbf{b} + \Delta \mathbf{b} \implies \mathbf{A}_0 \mathbf{x} = \mathbf{b}_0 \tag{5.26}$$

自然地，我们希望矫正数据矩阵和校正数据向量都尽量小。因此，总体最小二乘问题可以用约束优化问题叙述为：

$$\begin{aligned} \text{TLS: } \min_{\Delta \mathbf{A}, \Delta \mathbf{b}, \mathbf{x}} & \|[\Delta \mathbf{A}, \Delta \mathbf{b}]\|_{\text{F}}^2 = \|\Delta \mathbf{A}\|_{\text{F}}^2 + \|\Delta \mathbf{b}\|_2^2 \\ \text{s.t. } & (\mathbf{A} + \Delta \mathbf{A})\mathbf{x} = \mathbf{b} + \Delta \mathbf{b} \end{aligned}$$

约束条件有时也表示为 $(\mathbf{b} + \Delta \mathbf{b}) \in \text{Col}(\mathbf{A} + \Delta \mathbf{A})$

由(5.26)，校正过方程的解满足：

$$([\mathbf{A}, \mathbf{b}] + [\Delta \mathbf{A}, \Delta \mathbf{b}]) \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = \mathbf{0} \tag{5.27}$$

如果 $([\mathbf{A}, \mathbf{b}] + [\Delta \mathbf{A}, \Delta \mathbf{b}])$ 是列满秩的矩阵，记 $\hat{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix}$ ，则以 $\hat{\mathbf{x}}$ 为未知量的方程：

$$([\mathbf{A}, \mathbf{b}] + [\Delta\mathbf{A}, \Delta\mathbf{b}])\hat{\mathbf{x}} = \mathbf{0} \quad (5.28)$$

只有零解，与 $\hat{\mathbf{x}}$ 的最后一个分量为 -1 矛盾。因此， $([\mathbf{A}, \mathbf{b}] + [\Delta\mathbf{A}, \Delta\mathbf{b}])$ 是一个列亏损矩阵。问题转化为求一个最接近 $[\mathbf{A}, \mathbf{b}]$ 的列亏损矩阵。设 $[\mathbf{A}, \mathbf{b}]$ 的奇异值分解为

$$[\mathbf{A}, \mathbf{b}] = \sum_{i=1}^{n+1} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

其中 σ_i 为 $[\mathbf{A}, \mathbf{b}]$ 的第 i 个奇异值， $\mathbf{u}_i, \mathbf{v}_i$ 分别为对应的左右奇异向量。

也就是 $[\Delta\mathbf{A}, \Delta\mathbf{b}] = \sigma_{n+1} \mathbf{u}_{n+1} \mathbf{v}_{n+1}^T$ 。

设 $\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{x} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m$ ，其解为：

$$\hat{\mathbf{x}}_{\text{TLS}} = (\mathbf{A}^T \mathbf{A} - \sigma_{n+1}^2 \mathbf{I})^{-1} \mathbf{A}^T \mathbf{b}$$

其中 σ_{n+1} 为 $[-\mathbf{b}, \mathbf{A}]$ 的第 $n+1$ 个奇异值， $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{n+1}$ 。

5.2.4 最小二乘问题的解的敏感性

现在考虑向量 \mathbf{b} 的扰动对最小二乘解的影响。假定 \mathbf{b} 有扰动 $\Delta\mathbf{b}$ 且 \mathbf{x} 和 $\mathbf{x} + \Delta\mathbf{x}$ 分别是最小二乘问题

$$\min \|\mathbf{b} - \mathbf{Ax}\|_2 \quad \text{和} \quad \min \|(\mathbf{b} + \Delta\mathbf{b}) - \mathbf{Ax}\|_2$$

的解，即

$$\mathbf{x} = \mathbf{A}^\dagger \mathbf{b},$$

$$\mathbf{x} + \Delta\mathbf{x} = \mathbf{A}^\dagger (\mathbf{b} + \Delta\mathbf{b}) = \mathbf{A}^\dagger \tilde{\mathbf{b}}$$

其中 $\tilde{\mathbf{b}} = \mathbf{b} + \Delta\mathbf{b}$ 。下面的定理给出了由于 \mathbf{b} 的扰动而引起的 \mathbf{x} 的相对误差的界。

定理 5.2.7. 设 \mathbf{b}_1 和 $\tilde{\mathbf{b}}_1$ 分别是 \mathbf{b} 和 $\tilde{\mathbf{b}}$ 在 $\text{Col}(\mathbf{A})$ 上的正交投影。若 $\mathbf{b}_1 \neq 0$ ，则

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \kappa_2(\mathbf{A}) \frac{\|\mathbf{b}_1 - \tilde{\mathbf{b}}_1\|_2}{\|\mathbf{b}_1\|_2}$$

其中 $\kappa_2(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^\dagger\|_2$ 。

证明。证明：设 \mathbf{b} 在 $\text{Col}(\mathbf{A})^\perp$ 上的正交投影为 \mathbf{b}_2 ，则 $\mathbf{A}^T \mathbf{b}_2 = 0$ 。由 $\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2$ 可得

$$\mathbf{A}^\dagger \mathbf{b} = \mathbf{A}^\dagger \mathbf{b}_1 + \mathbf{A}^\dagger \mathbf{b}_2 = \mathbf{A}^\dagger \mathbf{b}_1 + (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}_2 = \mathbf{A}^\dagger \mathbf{b}_1$$

同理可证 $\mathbf{A}^\dagger \tilde{\mathbf{b}} = \mathbf{A}^\dagger \tilde{\mathbf{b}}_1$ 。因此

$$\|\Delta\mathbf{x}\|_2 = \|\mathbf{A}^\dagger \mathbf{b} - \mathbf{A}^\dagger \tilde{\mathbf{b}}\|_2 = \|\mathbf{A}^\dagger (\mathbf{b}_1 - \tilde{\mathbf{b}}_1)\|_2 \quad (5.29)$$

$$\leq \|\mathbf{A}^\dagger\|_2 \|\mathbf{b}_1 - \tilde{\mathbf{b}}_1\|_2 \quad (5.30)$$

由 $\mathbf{Ax} = \mathbf{b}_1$ 得

$$\|\mathbf{b}_1\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|_2 \quad (5.31)$$

由 (5.30) 和 (5.31) 立即得到定理的结论。 \square

这个定理告诉我们，在考虑 \mathbf{x} 的相差误差时，若 \mathbf{b} 有变化，只有它在 $\text{Col}(\mathbf{A})$ 上的投影会对解产生影响。此外，这个定理还告诉我们，最小二乘问题之解的敏感性依赖于数 $\kappa_2(\mathbf{A})$ 的大小。因此，我们称它为最小二乘问题的条件数。若 $\kappa_2(\mathbf{A})$ 很大，则称最小二乘问题是病态的；否则称为良态的。

作为本节的结束，我们给出 $\kappa_2(\mathbf{A})$ 与方阵 $\mathbf{A}^T \mathbf{A}$ 的条件数之间的关系。

定理 5.2.8. 设 \mathbf{A} 的列向量线性无关，则 $\kappa_2(\mathbf{A})^2 = \kappa(\mathbf{A}^T \mathbf{A})$ 。

证明.

$$\begin{aligned}\|\mathbf{A}\|_2^2 &= \lambda_{\max}(\mathbf{A}^T \mathbf{A}) = \|\mathbf{A}^T \mathbf{A}\|_2, \\ \|\mathbf{A}^\dagger\|_2^2 &= \|\mathbf{A}^\dagger (\mathbf{A}^\dagger)^T\|_2 = \|(\mathbf{A}^T \mathbf{A})^{-1}\|_2\end{aligned}$$

于是，有

$$\kappa_2(\mathbf{A})^2 = \|\mathbf{A}\|_2^2 \|\mathbf{A}^\dagger\|_2^2 = \|\mathbf{A}^T \mathbf{A}\|_2 \|(\mathbf{A}^T \mathbf{A})^{-1}\|_2 = \kappa(\mathbf{A}^T \mathbf{A}).$$

□

刚才我们仅仅考虑了 \mathbf{b} 的扰动对最小二乘解的影响问题，而要全面讨论最小二乘问题的敏感性问题，就必须考虑 \mathbf{A} 和 \mathbf{b} 同时都有微小扰动时，最小二乘解将有何变化，而这是一个非常复杂的问题，由于篇幅所限这里将不再进行讨论。

5.3 特征值计算

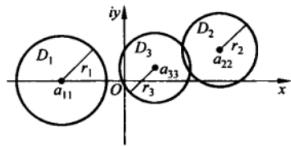
工程中许多实际问题都归结为求某些矩阵的特征值和特征向量：例如物理中的振动问题、稳定性问题，在数据科学以及机器学习中的网页链接分析问题（Google PageRank）、流形学习、谱聚类、线性判别分析、主成分分析等问题。

在数据科学中，我们一般只讨论实矩阵的特征值问题。应注意，实矩阵的特征值和特征向量不一定是实数和实向量，但实特征值一定对应于实特征向量，而一般的复特征值对应的特征向量一定不是实向量。此外，由于特征方程为实系数方程，若一个特征值不是实数，则其复共轭也一定是它的特征值。

对于一个实对称矩阵来说，它的 n 个特征值均为实数，并且存在 n 个正交的实特征向量。

5.3.1 矩阵特征值分布范围的估计

本节我们首先讨论矩阵特征值的分布范围或它们的界，其在理论上或者实际中都有重要作用，比如在敏感性分析和迭代法计算中都需要对矩阵的特征值分布范围的了解：

图 5.3: 复坐标平面, 以及 3×3 复矩阵 A 的盖氏圆

- 计算矩阵的 2-条件数

$$\text{cond}(A)_2 = \sqrt{\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)}}$$

- 考察一阶定常迭代法 $x^{(k+1)} = Bx^{(k)} + f$ 的收敛性、收敛速度, 收敛的判据是谱半径 $\rho(B) = \max_{1 \leq j \leq n} |\lambda_j(B)| < 1$, 收敛速度为 $R = -\log_{10} \rho(B)$

前面说明过谱半径的大小不超过任何一种算子范数, 即

$$\rho(A) \leq \|A\|$$

这是关于特征值上界的一个重要结论。

盖氏 (Gerschgorin) 圆盘和圆盘定理

为了细致描述 n 阶矩阵的特征值在复平面的分布范围, 首先引进 Gerschgorin 圆盘 (简称盖尔圆或盖氏圆)。本讲我们假设矩阵都是复矩阵。

定义 5.3.1. 设 $A = (a_{kj}) \in \mathbb{C}^{n \times n}$, 令 $R_k = \sum_{j=1, j \neq k}^n |a_{kj}|$, 则称集合 $D_k = \{z \mid z \in \mathbb{C} : |z - a_{kk}| \leq R_k\}$, $k = 1, 2, \dots, n$ 为在复平面内以 a_{kk} 为圆心、 R_k 为半径的圆盘, 称为 A 的第 k 个盖氏圆。

在很多情况下, 我们并不需要确切地知道矩阵的每一个特征值的大小, 而是要估计出这个矩阵各个特征值大概的范围。

定理 5.3.1. 圆盘定理 设 $A = (a_{kj}) \in \mathbb{C}^{n \times n}$, 则:

(I) A 的每一个特征值必属于 A 的格什戈林圆盘之中, 即对任一特征值 λ 必定存在 $k (1 \leq k \leq n)$, 使得

$$|\lambda - a_{kk}| \leq \sum_{j=1, j \neq k}^n |a_{kj}| \quad (5.32)$$

用集合的关系来说明, 这意味着 $\lambda(A) \subseteq \bigcup_{k=1}^n D_k$, 其中 $D_k = \{z \mid |z - a_{kk}| \leq \sum_{j=1, j \neq k}^n |a_{kj}|\}$

(2) 若 A 的格什戈林圆盘中有 m 个圆盘组成一连通并集 S , 且 S 与余下的 $n - m$ 个圆盘分离, 则 S 内恰好包含 A 的 m 个特征值 (重特征值按重数计)。

下面对定理 5.3.1 的结论 (1) 进行证明, 结论 (2) 的证明超出了本书的范围。

证明. 设 λ 为 A 的任一特征值, 则有 $Ax = \lambda x$. x 为非零常量. 设 x 中第 k 个分量最大, 即

$$|x_k| = \max_{1 \leq j \leq n} |x_j| > 0,$$

考虑线性方程中第 k 个方程

$$\sum_{j=1}^n a_{kj} x_j = \lambda x_k,$$

将其中与 x_k 有关的项移到等号左边, 其余移到右边, 再两边取模得

$$|\lambda - a_{kk}| |x_k| = \left| \sum_{\substack{j=1 \\ j \neq k}}^n a_{kj} x_j \right| \leq \sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}| |x_j| \leq |x_k| \sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}| \quad (5.33)$$

最后一个不等式的推导利用了“ x 中第 k 个分量最大”的假设. 将不等式 (5.33) 除以 $|x_k|$, 即得到式(5.32), 因此证明了定理 5.3.1 的结论 (1). 上述证明过程还说明, 若某个特征向量的第 k 个分量的模最大, 则相应的特征值必定属于第 k 个圆盘中. \square

还可以按照矩阵的每一列元素定义 n 个圆盘, 对于它们定理 5.32 仍然成立. 下面的定理是圆盘定理的重要推论, 其证明留给感兴趣的读者.

定理 5.3.2. 设 $A \in \mathbb{R}^{n \times n}$, 且 A 的对角元均大于 0, 则

- (1) 若 A 严格对角占优, 则 A 的特征值的实部都大于 0.
- (2) 若 A 为对角占优的对称矩阵, 则 A 一定是对称半正定矩阵, 若同时 A 非奇异, 则 A 为对称正定矩阵.

例 5.3.1. 试估计矩阵

$$\begin{pmatrix} 4 & 1 & 0 \\ 1 & 0 & -1 \\ 1 & 1 & -4 \end{pmatrix}$$

的特征值范围。

解. 直接应用圆盘定理, 该矩阵的三个圆盘如下:

$$D_1 : |\lambda - 4| \leq 1, \quad D_2 : |\lambda| \leq 2, \quad D_3 : |\lambda + 4| \leq 2.$$

D_1 与其他圆盘分离, 则它仅含一个特征值, 且必定为实数 (若为虚数则其共轭也是特征值, 这与 D_1 仅含一个特征值矛盾). 所以对矩阵特征值的范围的估计是

$$3 \leq \lambda_1 \leq 5, \quad \lambda_2, \lambda_3 \in D_2 \cup D_3.$$

再对矩阵 A^T 应用圆盘定理, 则可以进一步优化上述结果. 矩阵 A^T 对应的三个圆盘为

$$D_1' : |\lambda - 4| \leq 2, \quad D_2' : |\lambda| \leq 2, \quad D_3' : |\lambda + 4| \leq 1.$$

这说明 D_3' 中存在一个特征值，且为实数，它属于区间 $[-5, -3]$ ，经过综合分析可知三个特征值均为实数，它们的范围是

$$\lambda_1 \in [3, 5], \quad \lambda_2 \in [-2, 2], \quad \lambda_3 \in [-5, -3].$$

事实上，使用 MATLAB 的 *eig* 命令可求出矩阵 A 的特征值为 $4.2030, -0.4429, -3.7601$.

在估计特征值范围的时候，我们希望各个圆盘的半径越小越好。所以我们可以通过对矩阵 A 做相似变换，例如取 X 为对角阵，然后再应用圆盘定理估计特征值的范围.

例 5.3.2. (特征值范围的估计)：选取适当的矩阵 X ，应用定理 5.3.1 估计例 5.3.1 中矩阵的特征值范围.

解. 取

$$X^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.9 \end{pmatrix}$$

则

$$A_1 = X^{-1}AX = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 0 & -\frac{10}{9} \\ 0.9 & 0.9 & -4 \end{pmatrix}$$

的特征值与 A 的相同。对 A_1 应用圆盘定理，得到三个分离的圆盘，它们分别包含一个实特征值，由此得到特征值的范围估计

$$\lambda_1 \in [3, 5], \lambda_2 \in \left[-\frac{19}{9}, \frac{19}{9} \right], \lambda_3 \in [-5.8, -2.2].$$

此外，还可以进一步估计 $\rho(A)$ 的范围，即 $3 \leq \rho(A) \leq 5.8$ 。

上述例子表明，综合运用圆盘定理和矩阵特征值的性质，可对特征值的范围进行一定的估计。对具体例子，可适当设置相似变换矩阵，尽可能让圆盘相互分离，从而提高估计的有效性。

5.3.2 幂法

幂法是通过求矩阵的特征向量来求出特征值的一种迭代法。它主要用来求按模最大的特征值和相应的特征向量。其优点是算法简单，容易实现，缺点是收敛速度慢，其有效性依赖于矩阵特征值的分布情况。本节接下来将介绍幂法、反幂法以及加快幂法迭代收敛的技术。

定义 5.3.2. 在矩阵 A 的特征值中，模最大的特征值称为主特征值，也叫“第一特征值”，它对应的特征向量称为主特征向量。

应注意的是，主特征值有可能不唯一，因为模相同的复数可以有很多，例如模为 5 的特征值可能是 $5, -5, 3 + 4i, 3 - 4i$ 等等。另外注意谱半径和主特征值的区别。

如果矩阵 A 有唯一的主特征值，则一般通过幂法能够方便地计算出主特征值及其对应的特征向量。对于实矩阵，这个主特征值显然是实数，但不排除它是重特征值的情况。幂法的计算过程是，首先任取一非零向量 $\mathbf{x}_0 \in \mathbb{R}^n$ ，再进行迭代计算

$$\mathbf{x}_k = A\mathbf{x}_{k-1}, k = 1, 2, \dots$$

得到向量序列 $\{\mathbf{x}_k\}$ ，根据它即可求出主特征值与特征向量。下面我们来看一下具体的计算过程。

假设 A 的特征值可按模的大小排列为 $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$ ，且其对应特征向量 $\xi_1, \xi_2, \dots, \xi_n$ 线性无关。此时，任意非零向量 $\mathbf{x}^{(0)}$ 均可用 $\xi_1, \xi_2, \dots, \xi_n$ 线性表示，即

$$\mathbf{x}^{(0)} = \alpha_1 \xi_1 + \alpha_2 \xi_2 + \dots + \alpha_n \xi_n$$

且 $\alpha_1, \alpha_2, \dots, \alpha_n$ 不全为零。做向量序列 $\mathbf{x}^{(k)} = A^k \mathbf{x}^{(0)}$ ，则

$$\begin{aligned}\mathbf{x}^{(k)} &= A^k \mathbf{x}^{(0)} = \alpha_1 A^k \xi_1 + \alpha_2 A^k \xi_2 + \dots + \alpha_n A^k \xi_n \\ &= \alpha_1 \lambda_1^k \xi_1 + \alpha_2 \lambda_2^k \xi_2 + \dots + \alpha_n \lambda_n^k \xi_n \\ &= \lambda_1^k [\alpha_1 \xi_1 + \alpha_2 (\frac{\lambda_2}{\lambda_1})^k \xi_2 + \dots + \alpha_n (\frac{\lambda_n}{\lambda_1})^k \xi_n]\end{aligned}$$

由此可见，若 $\alpha_1 \neq 0$ ，则有

$$\lim_{k \rightarrow \infty} (\frac{\lambda_i}{\lambda_1})^k = 0, i = 2, \dots, n$$

故当 k 充分大的时候，必有

$$\mathbf{x}^{(k)} \approx \lambda_1^k \alpha_1 \xi_1$$

即 $\mathbf{x}^{(k)}$ 可以近似看成 λ_1 对应的特征向量，而 $\mathbf{x}^{(k)}$ 与 $\mathbf{x}^{(k-1)}$ 分量之比为

$$\frac{\mathbf{x}^{(k)}}{\mathbf{x}^{(k-1)}} \approx \frac{\lambda_1^k \alpha_1 \xi_1}{\lambda_1^{k-1} \alpha_1 \xi_1} = \lambda_1$$

于是利用向量序列 $\{\mathbf{x}^{(k)}\}$ 即可求出按模最大的特征值 λ_1 ，又可以求出对应的特征向量 ξ_1 。

在实际计算中，考虑到当 $|\lambda_1| > 1$ 时， $\lambda_1^k \rightarrow \inf$ ； $|\lambda_1| < 1$ 时， $\lambda_1^k \rightarrow 0$ ，因而计算 $\mathbf{x}^{(k)}$ 时可能会发生上溢或者下溢，故每一步将 $\mathbf{x}^{(k)}$ 归一化处理，即将 $\mathbf{x}^{(k)}$ 的各分量都除以模最大的分量，使 $\|\mathbf{x}^{(k)}\| = 1$ ，于是求 A 按模最大的特征值 λ_1 和对应的特征向量 ξ_1 的算法，可归纳为如下步骤。

- (1) 输入矩阵 A ，初始向量 $\mathbf{v}^{(0)}$ ，误差限 ϵ ，最大迭代次数 N 。记 m_0 是 $\mathbf{v}^{(0)}$ 按模最大的分量，
 $\mathbf{x}^{(0)} = \mathbf{v}^{(0)} / m_0$ 。置 $k = 0$
- (2) 计算 $\mathbf{v}^{(k+1)} = A\mathbf{x}^{(k)}$ 。记 m_{k+1} 是 $\mathbf{v}^{(k+1)}$ 按模最大的分量，
 $\mathbf{x}^{(k+1)} = \mathbf{v}^{(k+1)} / m_{k+1}$
- (3) 若 $|m_{k+1} - m_k| < \epsilon$ ，停止计算，输出近似特征值 m_{k+1} 和近似特征向量 $\mathbf{x}^{(k+1)}$ 否则转
(4)
- (4) 若 $k < N$ 置 $k = k + 1$ 转 (2) 否则输出计算失败信息，停止计算。

上述算法我们称为幂法。

Algorithm 15 幂法

-
- 1: $k = 0; \mathbf{x}^{(k)} = \mathbf{x}$
 - 2: **repeat**
 - 3: $\mathbf{y}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)}$
 - 4: $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)} / \|\mathbf{y}^{(k+1)}\|_\infty$
 - 5: $\lambda^{(k+1)} = \mathbf{x}^{(k+1)T} \mathbf{A} \mathbf{x}^{(k+1)}$
 - 6: $k = k + 1$
 - 7: **until** 收敛
-

我们将经过归一化处理的幂法总结为如下的定理:

定理 5.3.3. 设矩阵 \mathbf{A} 的特征值可按模的大小排列为 $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$, 且对应特征向量 $\xi_1, \xi_2, \dots, \xi_n$ 线性无关. 序列 $\{\mathbf{x}^{(k)}\}$ 有算法产生, 则有

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \frac{\xi_1}{\max\{\xi_1\}} = \xi_1^0, \quad \lim_{k \rightarrow \infty} m_k = \lambda_1, \quad (5.34)$$

式中: ξ_1^0 为将 ξ_1 归一化后得到的向量: $\max\{\xi_1\}$ 为向量 ξ_1 模最大的分量.

证明. 由算法15的步 2 和步 3 知

$$\mathbf{x}^{(k)} = \frac{\mathbf{v}^{(k)}}{m_k} = \frac{\mathbf{A}\mathbf{x}^{(k-1)}}{m_k} = \frac{\mathbf{A}^2\mathbf{x}^{k-2}}{m_k m_{k-1}} = \dots = \frac{\mathbf{A}^k \mathbf{x}^{(0)}}{m_k m_{k-1} \dots m_1}.$$

由于 $\mathbf{x}^{(k)}$ 的最大分量为 1, 即 $\max\{\mathbf{x}^{(k)}\} = 1$, 故

$$m_k m_{k-1} \dots m_1 = \max\{\mathbf{A}^k \mathbf{x}^{(0)}\}$$

从而

$$\begin{aligned} \mathbf{x}^{(k)} &= \frac{\mathbf{A}^k \mathbf{x}^{(0)}}{\max\{\mathbf{A}^k \mathbf{x}^{(0)}\}} = \frac{\lambda_1^k [\alpha_1 \xi_1 + \sum_{i=2}^n \alpha_i (\frac{\lambda_i}{\lambda_1})^k \xi_i]}{\max\{\lambda_1^k [\alpha_1 \xi_1 + \sum_{i=2}^n \alpha_i (\frac{\lambda_i}{\lambda_1})^k \xi_i]\}} \\ &= \frac{\alpha_1 \xi_1 + \sum_{i=2}^n \alpha_i (\frac{\lambda_i}{\lambda_1})^k \xi_i}{\max\{\alpha_1 \xi_1 + \sum_{i=2}^n \alpha_i (\frac{\lambda_i}{\lambda_1})^k \xi_i\}} \end{aligned}$$

可见

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \frac{\alpha_1 \xi_1}{\max\{\alpha_1 \xi_1\}} = \frac{\xi_1}{\max\{\xi_1\}} = \xi_1^0.$$

又

$$\begin{aligned}\mathbf{v}^{(k)} &= \mathbf{A}\mathbf{x}^{(k-1)} = \frac{\mathbf{A}^k \mathbf{x}^{(0)}}{m_{k-1} \cdots m_1} = \frac{\mathbf{A}^k \mathbf{x}^{(0)}}{\max\{\mathbf{A}^{(k-1)} \mathbf{x}^{(0)}\}} \\ &= \frac{\lambda_1^k [\alpha_1 \xi_1 + \sum_{i=2}^n \alpha_i (\frac{\lambda_i}{\lambda_1})^k \xi_i]}{\lambda_1^{k-1} \max\{[\alpha_1 \xi_1 + \sum_{i=2}^n \alpha_i (\frac{\lambda_i}{\lambda_1})^{k-1} \xi_i]\}}\end{aligned}$$

注意到 m_k 是 $\mathbf{v}^{(k)}$ 模的最大的分量，既有

$$m_k = \max\{\mathbf{v}^{(k)}\} = \lambda_1 \frac{\max\{\alpha_1 \xi_1 + \sum_{i=2}^n \alpha_i (\frac{\lambda_i}{\lambda_1})^k \xi_i\}}{\max\{\alpha_1 \xi_1 + \sum_{i=2}^n \alpha_i (\frac{\lambda_i}{\lambda_1})^{k-1} \xi_i\}}$$

从而 $\lim_{k \rightarrow \infty} m_k = \lambda_1$ 成立。证毕。 \square

5.3.3 加速幂法的方法

幂法的收敛速度与比值 $|\lambda_2/\lambda_1|$ 的大小有关， $|\lambda_2/\lambda_1|$ 越小，收敛速度越快，当此比值接近于 1 时，收敛速度是非常缓慢的。我们可以用一些改进方法加速幂法迭代收敛过程。加速幂法收敛的方法主要有两种：

- 原点位移法
- 瑞利商加速

原点位移法

对原矩阵作一原点位移，令

$$\mathbf{B} = \mathbf{A} - \alpha \mathbf{I}$$

α 为要选取的参数。设矩阵 \mathbf{A} 的特征值为 $\lambda_1, \lambda_2, \dots, \lambda_n$ ，对应的特征向量为 $\xi_1, \xi_2, \dots, \xi_n$ ，则矩阵 \mathbf{B} 的特征值为 $\lambda_1 - \alpha, \lambda_2 - \alpha, \dots, \lambda_n - \alpha$ ， \mathbf{B} 的特征向量与 \mathbf{A} 的特征向量相同。假设原点位移后， \mathbf{B} 的特征值 $\lambda_1 - \alpha$ 仍为模最大的特征值，选择 α 的目的是使

$$\max_{2 \leq i \leq n} \frac{\lambda_i - \alpha}{\lambda_1 - \alpha} < \frac{\lambda_2}{\lambda_1}$$

适当地选择 α 可使幂法的收敛速度得到加速。此时 $m_k \rightarrow \lambda_1 - \alpha, m_k + \alpha \rightarrow \lambda_1$ ，而 $\mathbf{x}^{(k)}$ 仍然收敛于 \mathbf{A} 的特征向量 ξ_1^0 。这种加速方法叫做原点位移法。

Algorithm 16 原点位移法

```

1:  $k = 0; \mathbf{x}^{(k)} = \mathbf{x}$ 
2: repeat
3:    $\mathbf{y}^{(k+1)} = (\mathbf{A} - \alpha\mathbf{I})^{-1}\mathbf{x}^{(k)}$ 
4:    $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)} / \|\mathbf{y}^{(k+1)}\|_\infty$ 
5:    $\lambda(k+1) = \mathbf{x}^{(k+1)T}\mathbf{A}\mathbf{x}^{(k+1)}$ 
6:    $k = k + 1$ 
7: until 收敛

```

瑞利商加速

假设在原点位移法的某个步骤中，我们有一个近似特征向量 $\mathbf{x}^{(k)} \neq 0$ 。然后，我们寻找近似特征值 λ_k ，也就是满足下列方程的特征值和特征向量

$$\mathbf{x}^{(k)} \lambda_k = \mathbf{A}\mathbf{x}^{(k)}$$

我们寻找特征值 λ_k ，就是要使得方程残差的平方范数最小，即 $\min \|\mathbf{x}^{(k)} \lambda_k - \mathbf{A}\mathbf{x}^{(k)}\|_2^2$ 。通过令导数为 0 得到

$$\lambda_k = \frac{\mathbf{x}^{(k)T}\mathbf{A}\mathbf{x}^{(k)}}{\mathbf{x}^{(k)T}\mathbf{x}^{(k)}}$$

这个量称为瑞利商。

我们如果在原点位移法中根据瑞利商来选择位移，则可以得到瑞利商迭代算法。可以证明瑞利商迭代算法具有局部二次收敛性，即经过一定次数的迭代后，迭代 $k+1$ 次时运行解的收敛间隙与迭代 k 次时该解的间隙平方成正比。

Algorithm 17 瑞利商加速

```

1:  $k = 0; \mathbf{x}^{(k)} = \mathbf{y}$ 
2: repeat
3:    $\lambda_k = \frac{\mathbf{x}^{(k)T}\mathbf{A}\mathbf{x}^{(k)}}{\mathbf{x}^{(k)T}\mathbf{x}^{(k)}}$ 
4:    $\mathbf{y}^{(k+1)} = (\mathbf{A} - \lambda_k\mathbf{I})\mathbf{x}^{(k)}$ 
5:    $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)} / \|\mathbf{y}^{(k+1)}\|_\infty$ 
6:    $k = k + 1$ 
7: until 收敛

```

5.3.4 反幂法

反幂法 (inverse iteration) 基于幂法，可看成是幂法的一种应用，它能够求矩阵 \mathbf{A} 按模最小的特征值及其特征向量。对于一个非奇异矩阵 \mathbf{A} , \mathbf{A}^{-1} 的特征值为矩阵 \mathbf{A} 的特征值的倒数， \mathbf{A}^{-1}

的主特征值便是 A 按模最小的特征值的倒数。因此，可对 A^{-1} 应用幂法求出矩阵 A 的最小特征值。这就是反幂法的基本思想。

与幂法相对应，反幂法的适用条件是：矩阵 A 按模最小的特征值唯一，且几何重数等于代数重数。对于实矩阵，满足此条件时这个最小特征值一定是实数，相应的特征向量也为实向量。算法过程描述如下：

设 A 可逆，由于 $A\xi_i = \lambda_i \xi_i$ 时，成立 $A^{-1}\xi_i = \lambda_i^{-1}\xi_i$ 。因此，若 $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_{n-1}| > |\lambda_n|$ ，则 λ_n^{-1} 是 A^{-1} 按模最大的特征值，此时按反幂法，必有

$$m_k \rightarrow \lambda_n^{-1}, \mathbf{x}^{(k)} \rightarrow \xi_n^0$$

其收敛率为 $|\lambda_n/\lambda_{n-1}|$ 。任取初始向量 $\mathbf{x}^{(0)}$ ，构造向量序列

$$\mathbf{x}^{(k+1)} = A^{-1}\mathbf{x}^{(k)}, k = 0, 1, 2, \dots$$

按幂法计算即可。

但用上述式子计算，首先要求 A^{-1} ，这比较麻烦而且是不经济的，实际计算中通常用解方程组的办法，即用

$$A\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}, k = 0, 1, 2, \dots$$

求 $\mathbf{x}^{(k+1)}$ 。为防止计算机溢出，实际计算时所用公式为

$$\begin{aligned} \mathbf{v}^{(k)} &= \mathbf{x}^{(k)} / \max(\mathbf{x}^{(k)}), \\ A\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)}, \end{aligned} \quad k = 0, 1, 2, \dots$$

式中： $\max(\mathbf{x}^{(k)})$ 为 $\mathbf{x}^{(k)}$ 模最大的分量。

在实际计算中，若知道某个矩阵特征值的估计值，常利用反幂法结合原点位移技术来求其精确值和对应的特征向量。

若 A 的特征值是 λ ，则 $\lambda - \alpha$ 是 $A - \alpha I$ 的特征值。因此反幂法可以用于已知矩阵的近似特征值为 α 时，求矩阵的特征向量并且提高特征值精度。

此时，可以用原点位移法来加速迭代过程，于是上式相应为

$$(A - \alpha I)\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}, k = 0, 1, 2, \dots$$

求 $\mathbf{x}^{(k+1)}$ 。为防止计算机溢出，实际计算时所用公式为

$$\begin{aligned} \mathbf{v}^{(k)} &= \mathbf{x}^{(k)} / \max(\mathbf{x}^{(k)}), \\ (A - \alpha I)\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)}, \end{aligned} \quad k = 0, 1, 2, \dots$$

总结如下：

算法（反幂法）

(1) 选取初值 $\mathbf{x}^{(0)}$ ，近似值 α ，误差限 ϵ ，最大迭代次数 N 。记 m_0 为 $\mathbf{x}^{(0)}$ 中按模最大的分量，
 $\mathbf{v}^{(0)} = \mathbf{x}^{(0)} / m_0$ 。置 $k := 0$

(2) 解方程组 $(A - \alpha I)\mathbf{x}^{(k+1)} = \mathbf{v}^{(k)}$ 得 $\mathbf{x}^{(k+1)}$ 。

(3) 记 m_{k+1} 为 $\mathbf{x}^{(k+1)}$ 中按模最大的分量，
 $\mathbf{v}^{(k+1)} = \mathbf{x}^{(k+1)} / m_{k+1}$ 。

- (4) 若 $|m_{k+1}^{-1} - m_k^{-1}| < \epsilon$, 则置 $\lambda := m_{k+1}^{-1} + \alpha$, 输出 λ 和 $\mathbf{x}^{(k+1)}$, 停算; 否则, 转 (5)
 (5) 若 $k < N$, 置 $k := k + 1$, 转 (2), 否则输出计算失败信息, 停算。

Algorithm 18 反幂法

```

1:  $k = 0; \mathbf{x}^{(k)} = \mathbf{x}$ 
2: repeat
3:    $\mathbf{y}^{(k+1)} = (\mathbf{A} - \alpha\mathbf{I})^{-1}\mathbf{x}^{(k)}$ 
4:    $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)} / \|\mathbf{y}^{(k+1)}\|_\infty$ 
5:    $\lambda^{(k+1)} = \mathbf{x}^{(k+1)T} \mathbf{A} \mathbf{x}^{(k+1)} + \alpha$ 
6:    $k = k + 1$ 
7: until 收敛

```

5.3.5 特征值计算的应用：PageRank 网页排名

接下来我们将介绍一个用于网页排名的算法——PageRank。

(1) 问题背景 互联网 (Internet) 的使用已经深入到人们的日常生活中, 其巨大的信息量和强大的功能给生产、生活带来了很大的便利。随着网络的信息量越来越庞大, 如何有效地搜索出用户真正需要的信息变得十分重要。自 1998 年搜索引擎网站 Google 创立以来, 网络搜索引擎成为解决上述问题的重要手段。

1998 年, 美国斯坦福大学的博士生 Larry Page 和 Sergey Brin 创立了 Google 公司, 他们的核心技术就是通过 PageRank 技术对海量的网页进行重要性分析。该技术利用网页相互链接的关系对网页进行组织, 确定出每个网页的重要级别 (PageRank)。当用户进行搜索时, Google 找出符合搜索要求的网页, 并按它们的 PageRank 大小依次列出。这样, 用户一般显示结果的第一页或者前几页就能找到真正有用的结果。

形象地解释, PageRank 技术的基本原理是: 如果网页 A 链接到网页 B, 则认为“网页 A 投了网页 B”一票, 而且如果网页 A 是级别高的网页, 则网页 B 的级别也相应地高。

(2) 数学问题建模 假设 n 是 Internet 中所有可访问网页的数目, 此数值非常大, 再 2010 年已接近 100 亿。定义 $n \times n$ 的网页连接矩阵 $\mathbf{G} = (g_{ij}) \in \mathbb{R}^{n \times n}$, 若从网页 j 有一个链接到网页 i , 则 $g_{ij} = 1$, 否则 $g_{ij} = 0$ 。矩阵 \mathbf{G} 有如下特点:

- (1) \mathbf{G} 矩阵是大规模稀疏矩阵;
- (2) 第 j 列非零元素的位置表示了从网页 j 链接出去的所有网页;
- (3) 第 i 列非零元素的位置表示了所有链接到网页 i 的网页;
- (4) \mathbf{G} 中非零元的数目为整个 Internet 中存在的超链接的数量;
- (5) 记 \mathbf{G} 矩阵行元素之和 $r_i = \sum_j g_{ij}$, 它表示第 i 个网页的“入度”;
- (6) 记 \mathbf{G} 矩阵列元素之和 $c_j = \sum_i g_{ij}$, 它表示第 j 个网页的“出度”。

要计算 PageRank，可假设一个随机上网“冲浪”的过程，即每次看完当前网页后，有两种选择：

- (1) 在当前网页中随机选一个超链接进入下一个网页；
- (2) 随机地新开一个网页；

设 p 为选择当前网页上链接的概率（比如 $p = 0.85$ ），则 $1 - p$ 为不选当前网页的链接而随机打开一个网页的概率。若当前网页是网页 j ，则如何计算下一步浏览到达网页 i 的概率（网页 j 到 i 的转移概率）？它有两种可能性：

- (1) 若网页 i 在网页 j 的链接上，其概率为 $p \cdot 1/c_j + (1 - p) \cdot 1/n$ ；
- (2) 若网页 i 不在网页 j 的链接上，其概率为 $(1 - p) \cdot 1/n$ 。

由于网页 i 是否在网页 j 的链接上由 g_{ij} 决定，网页 j 到 i 的转移概率为

$$a_{ij} = g_{ij} \left[p \cdot \frac{1}{c_j} + (1 - p) \cdot \frac{1}{n} \right] + (1 - g_{ij}) \left[(1 - p) \cdot \frac{1}{n} \right] = \frac{pg_{ij}}{c_j} + \frac{1-p}{n}$$

应注意的是，若 $c_j = 0$ 意味着 $g_{ij} = 0$ ，上式改为 $a_{ij} = 1/n$ 。任意两个网页之间的转移概率形成了一个转移矩阵 $\mathbf{A} = (a_{ij})_{n \times n}$ 。设矩阵 \mathbf{D} 为各个网页出度的倒数（若没有出度，设为 1）构成的 n 阶对角阵， \mathbf{e} 为全是 1 的 n 维向量，则

$$\mathbf{A} = p\mathbf{G}\mathbf{D} + \frac{1-p}{n}\mathbf{e}\mathbf{e}^T.$$

这在数学上称为马尔可夫过程。若这样的随机“冲浪”一直进行下去，某个网页被访问的极限概率就是它的 PageRank。

设 $x_i^{(k)}, i = 1, 2, \dots, n$ 表示某时刻 k 浏览网页 i 的概率 $(\sum x_i^{(k)}=1)$ ，向量 $\mathbf{x}^{(k)}$ 表示当前时刻浏览个网页的概率分布。那么下一时刻浏览到网页 i 的概率为 $\sum_{j=1}^n a_{ij}x_i^{(k)}$ ，此时浏览个网页的概率分布为 $\mathbf{x}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)}$ 。

当这个过程无限进行下去，达到极限情况，即网页访问概率 $\mathbf{x}^{(k)}$ 收敛到一个极限值，这个极限向量 \mathbf{x} 为个网页的 PageRank，他满足 $\mathbf{A}\mathbf{x} = \mathbf{x}$ ，且 $\sum_{i=1}^n x_i = 1$ 。

总结一下，我们要求解的问题是在给定 $n \times n$ 的网页连接矩阵 \mathbf{G} ，以及选择当前网页链接的概率 p 时，计算特征值 1 对应的特征向量 \mathbf{x}

$$\begin{cases} \mathbf{A}\mathbf{x} = \mathbf{x} \\ \sum_{i=1}^n x_i = 1 \end{cases}$$

易知 $\|\mathbf{A}\|_1 = 1$ ，所以 $\rho(\mathbf{A}) \leq 1$ 。又考虑矩阵 $\mathbf{L} = \mathbf{I} - \mathbf{A}$ ，容易验证它各列元素和均为 0，则 \mathbf{L} 为奇异矩阵，所以 $\det(\mathbf{I} - \mathbf{A}) = 0$ ，1 是 \mathbf{A} 的特征值且为主特征值。更进一步，用圆盘定理考察矩阵 \mathbf{A}^T 的特征值分布，图(a) 显示了第 j 个圆盘 $D_j (j = 1, 2, \dots, n)$ ，显然其圆心 $a_{jj} > 0$ ，半径 r_j 满足 $a_{jj} + r_j = 1$ ，因此除了 1 这一点，圆盘上任何一点到圆心的距离（即复数的模）都小于 1。这就说明，1 是矩阵 \mathbf{A}^T 和 \mathbf{A} 的唯一主特征值。对于实际的大规模稀疏矩阵 \mathbf{A} ，幂法是求其主特征向量的可靠的、唯一的选择。

网页的 PageRank 完全由所有网页的超链接结构所决定，隔一段时间重新算一次 PageRank 以反映互联网的发展变化，此时将上一次计算的结果作为幂法的迭代初值可提高收敛速度。由于迭代向量以及矩阵 A 的物理意义。在使用幂法时并不需要对向量进行规格化，而且不需要形成矩阵 A 。通过遍历整个网页的数据库，根据网页间超链接关系即可得到 $Ax^{(k)}$ 的结果。

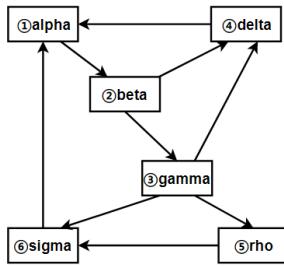


图 5.4: 网页链接关系

用一个只有 6 个网页的微型网络作为例子，其网页链接关系如图5.4所示。

通过下述 MATLAB 命令可生成矩阵 G

```
i = [2 3 4 4 5 6 1 6 1]; //网页 i
j = [1 2 2 3 3 3 4 5 6]; //网页 j
n = 6;
G = sparse(i, j, 1, n, n); //生成 j → i 的网络连接稀疏矩阵
```

再使用下述命令得到矩阵 A

```
c = full(sum(G));
D = spdiags(1./c', 0, n, n);
e = ones(n, 1);
p = .85; delta = (1 - p)/n;
A = p * G * D + delta * e * e';
```

也可以使用 Python 中的 NumPy 库生成 G

```
i = np.array([2, 3, 4, 4, 5, 6, 1, 6, 1]) - 1
j = np.array([1, 2, 2, 3, 3, 3, 4, 5, 6]) - 1
data = np.ones(len(i))
n = 6;
G = csr_matrix((data, (i, j)), shape = (n, n)).toarray()
```

再使用下述命令得到矩阵 A

```
c = np.sum(G, axis = 0)
```

```

 $D = np.diag(1/c)$ 
 $e = np.ones((n, n))$ 
 $p = .85; delta = (1 - p)/n$ 
 $A = p * np.matmul(G, D) + delta * e$ 

```

使用幂方法可求出其主特征向量，其步骤如下：

(1) 给出初始向量 $\mathbf{x}_0 = [1 \ 1 \ 1 \ 1 \ 1]^T$

(2) $\mathbf{x}^{k+1} = A\mathbf{x}^k$

(3) 归一化：

$$\mathbf{x}^{k+1} = \frac{\mathbf{x}^{k+1}}{\sum_{i=1}^n x_i^{k+1}}$$

(4) 当 $\mathbf{x}^{k+1} - \mathbf{x}^k > \varepsilon$, 重复计算 (2)(3)

设置迭代次数为 1000，在每 k 次迭代里，计算 (3)(4)，最后可得到 PageRank 为

$$\mathbf{x} = [0.2675 \ 0.2524 \ 0.1323 \ 0.1697 \ 0.0625 \ 0.1156]^T$$

禁书请勿外传

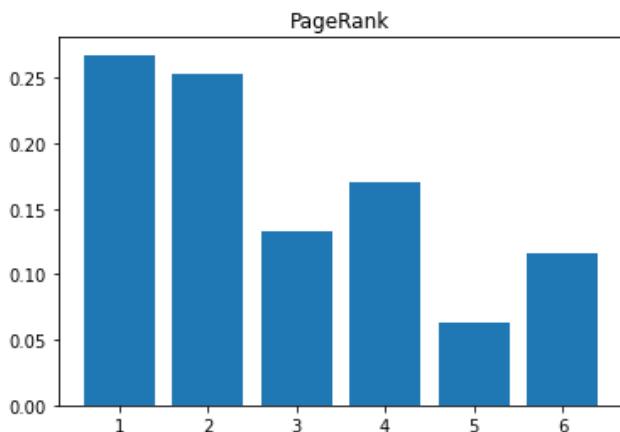


图 5.5: 网页的级别高低

使用 MATLAB 的 bar 命令或 Python 中 Matplotlib 库的 `pyplot.bar()` 函数，将 \mathbf{x} 的各分量显示如图 5.5 所示，从中看出各个网页的级别高低，虽然链接数目一样，但是网页 alpha 1 的链接比 delta 4 和 sigma 6 都高，而 beta 2 的级别第二高，因为高级别的 alpha 1 链接到它上面，它沾了 alpha 1 的光。

5.4 阅读材料

本章介绍了数值线性代数三大核心主题内容，包括线性方程的求解、最小二乘问题和特征值的求解。数据科学中的很多问题最终都归结为线性方程的求解，因此这一章主要介绍线性方程组的类型和解的结构，引入线性方程组和最小二乘问题的求解方法，并讨论解的敏感性，这些内容将与后续优化问题求解、数据科学中的线性回归问题相联系。此外，还介绍了大规模矩阵求解特征值的一些计算方法，包括幂迭代法，这已被广泛应用于数据科学中的搜索技术 pagerank 的矩阵特征值计算。此外，用于对高维数据进行非线性降维和聚类的更现代的谱方法，如 Isomap (Tenenbaum 等, 2000), Laplacian 特征映射 (Belkin 和 Niyogi, 2003), Hessian 特征映射 (Donoho 和 Grimes, 2003), 谱聚类 (Shi 和 Malik, 2000) 等，每一个都需要计算正定核的特征向量和特征值，这些核心计算通常由低秩矩阵近似技术 (Belabbas 和 Wolfe, 2009) 支持，正如我们在 SVD 中遇到的那样。另外，关于稠密数值线性代数可参考 (Golub 和 Van Loan, 1989), (Trefethen 和 Bau, 1997) 等。(Gill, Murray, 1981) 和 (Wright, 1997), (Wright 以及 Nocedal, 1999) 等书籍注重于数值优化问题的数值线性代数介绍。关于数值线性代数软件包，可以参考 LAPACK，其包括常规的稠密的线性代数算法的高质量实现。LAPACK 在基本线性代数子程序 (BLAS) 的基础上建成，后者是基本的向量和矩阵运算的程序库，可以很容易地根据具体的计算机结构的优点进行定制，也可得到求解稀疏的线性方程组的一些源代码，包括 SPOOLES, SuperLU, UMFPACK 以及 WSMP 等等，这里提到的只是其中少数几个。

习题

习题 5.1. 设 $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ 用正则化方法求对应的 LS 问题的解。

习题 5.2. 设 $A = \begin{bmatrix} 1 & 3 & 1 & 1 \\ 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ 求对应的 LS 问题的全部解。

习题 5.3. 设 $A \in \mathbb{R}^{m \times n}$ 且存在 $X \in \mathbb{R}^{n \times m}$ 使得对每一个 $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{x} = X\mathbf{b}$ 均极小化 $\|A\mathbf{x} - \mathbf{b}\|_2$. 证明 $A\mathbf{X}\mathbf{A} = A$ 和 $(A\mathbf{X})^T = A\mathbf{X}$.

习题 5.4. 利用等式

$$\|A(\mathbf{x} + \alpha\mathbf{w}) - \mathbf{b}\|_2^2 = \|A\mathbf{x} - \mathbf{b}\|_2^2 + 2\alpha\mathbf{w}^T A^T(A\mathbf{x} - \mathbf{b}) + \alpha^2\|\mathbf{A}\mathbf{w}\|_2^2$$

证明：如果 $\mathbf{x} \in X_{LS}$, 那么 $A^T A\mathbf{x} = A^T \mathbf{b}$

习题 5.5. 给定点集 $\mathbf{p}_1, \dots, \mathbf{p}_m \in \mathbb{R}^n$ 构成的 $m \times n$ 矩阵 $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_m]$. 考虑问题

$$\min_{\mathbf{X}} F(\mathbf{X}) = \sum_{i=1}^m \|\mathbf{x}_i - \mathbf{p}_i\|_2^2 + \frac{\lambda}{2} \sum_{1 \leq i, j \leq m} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$$

其中 $\lambda \geq 0$ 为参数, 变量是一个 $m \times n$ 矩阵 $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]$, 其中 $\mathbf{x}_i \in \mathbb{R}^n$ 是 \mathbf{X} 的第 i 列, $i = 1, \dots, m$. 上述问题尝试聚类点集 \mathbf{p}_i , 第一项鼓励聚类中心 \mathbf{x}_i 靠近对应的点 \mathbf{p}_i , 第二项鼓励 \mathbf{x}_i 们之间彼此靠近, 当 λ 增大的时候, 对应更高的组群影响。

1. 请说明这个问题属于最小二乘类问题。不需要明确阐述这个问题的形式。

2. 证明 $\frac{1}{2} \sum_{1 \leq i, j \leq m} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = \text{trace}(\mathbf{X}\mathbf{H}\mathbf{X}^T)$, 其中 $\mathbf{H} = m\mathbf{I}_m - \mathbf{1}\mathbf{1}^T$ 是一个 $m \times m$ 矩阵, \mathbf{I}_m 是 $m \times m$ 单位矩阵, $\mathbf{1}$ 是 \mathbb{R}^n 中的单位向量。

3. 证明 \mathbf{H} 是半正定的。

4. 证明函数 F 在矩阵 \mathbf{X} 处的梯度是一个 $n \times m$ 矩阵, 为:

$$\nabla F(\mathbf{X}) = 2(\mathbf{X} - \mathbf{P} + \lambda \mathbf{X}\mathbf{H})$$

提示: 对于第二项, 找到函数的一阶展式, $\Delta \rightarrow \text{trace}((\mathbf{X} + \Delta)\mathbf{H}(\mathbf{X} + \Delta)^T)$, 其中 $\Delta \in \mathbb{R}^{n,m}$ 。

5. 依据最小二乘问题的最优条件为目标函数的梯度为零。证明最优点集的形式为:

$$\mathbf{x}_i = \frac{1}{m\lambda + 1} \mathbf{p}_i + \frac{m\lambda}{m\lambda + 1} \hat{\mathbf{p}}, i = 1, \dots, m,$$

其中 $\hat{\mathbf{p}} = (1/m)(\mathbf{p}_1 + \dots + \mathbf{p}_m)$ 是给定点集的中心。

6. 阐述你的结果, 你认为这是聚类点集的一个好的模型么?

习题 5.6. 判断 $[1, 3, 4]$ 的转置是否在 \mathbf{A} 的零空间中

$$\mathbf{A} = \begin{bmatrix} 3 & 5 & -3 \\ 6 & -2 & 0 \\ -8 & 4 & 1 \end{bmatrix}$$

习题 5.7. 求矩阵

$$\begin{bmatrix} 5 & 21 & 19 \\ 13 & 23 & 2 \\ 8 & 14 & 1 \end{bmatrix}$$

的行空间和列空间

习题 5.8. 简答: 阐述非负矩阵分解和主成分分析的相同点和不同点

参考文献

- [1] E.Anderson, Z.Bai, C.Bischof, S.Blackford, J.Demrnel, J.Dongarra, J.DuCroz, A.Greenbaum, S.Hammarling, A.McKenney, and D.Sorensen. LAPACK Users' Guide. Society for Industrial and Applied Mathematics, third edition, 1999. Available from www.netlib.org/lapack.

- [2] C.Ashcraft, D.Pierce, D.K.Wah, and J.Wu.The Reference Manual for SPOOLES Version 2.2: An Object Oriented Software Library for Solving Sparse Linear Systems of Equations, 1999. Available from www.netlib.org/linalg/spooles/spooles.2.2.html.
- [3] T.A.Davis.UMFPACK User Guide, 2003. Available from www.cise.ufl.edu/research/sparse/umfpack.
- [4] J.W.Demmel.Applied Numerical Linear Algebra.Society for Industrial and Applied Mathematics, 1997.
- [5] I.S.Duff, A.M.Erisman, and J.K.Reid.Direct Methods for Sparse Matrices.Clarendon Press, 1986.
- [6] J.W.Demmel, J.R.Gilbert, and X.S.Li.SuperLU Users' Guide, 2003. Available from crd.lbl.gov/xiaoye/SuperLU.
- [7] I.S.Duff.The solution of augmented systems.In D.F.Griffiths and G.A.Watson, editors, Numerical Analysis 1993.Proceedings of the 15th Dundee Conference, pages 40- 55.Longman Scientific & Technical, 1993.
- [8] G.Golub and C.F.Van Loan.Matrix Computations.Johns Hopkins University Press, second edition, 1989.
- [9] A.George and J.W.-H.Liu.Computer sohstion of large sparse positive definite systems.Prentice-Hall, 1981.
- [10] Belkin, Mikhail, and Niyogi, Partha. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6), 1373–1396.
- [11] Tenenbaum, Joshua B, De Silva, Vin, and Langford, John C. 2000. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500), 2319–2323.
- [12] Donoho, David L, and Grimes, Carrie. 2003. Hessian eigenmaps: Locally linear em- bedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10), 5591–5596.
- [13] Belabbas, Mohamed-Ali, and Wolfe, Patrick J. 2009. Spectral methods in machine learning and new strategies for very large datasets. *Proceedings of the National Academy of Sciences*, pnas – 0810600105.
- [14] Shi, Jianbo, and Malik, Jitendra. 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8), 888–905.
- [15] A.Gupta.WSMP: Watson Sparse Matrix Package.Part I—Direct Solution of Symmetric Sparse Systems.Part II—Direct Solution of General Sparse Systems, 2000. Available from www.cs.umn.edu/~agupta/wsmp.
- [16] N.J.Higham.Accuracy and Stability of Numerical Algorithms.Society for Industrial and Applied Mathematics, 1996,
- [17] P.E.Gill, W.Murray, and M.H.Wright.Practical Optimization, Academic Press, 1981.

- [18] J.Nocedal and S.J.Wright.Numerical OptimizationL Springer, 1999.
- [19] L.N.Trefethen and D.Bau, III.Numerical Linear Algebra.Society for Industrial and Applied Mathematics, 1997.
- [20] S.J.Wright.Primal-Dual Interior-Point Methods.Society for Industrial and Applied Mathematics, 1997.

草稿请勿外传

草稿请勿外传

第六章 向量与矩阵微分

机器学习中的很多任务可以看作是学习某个函数，比如，判断一张图片是猫还是狗或是其它，就是学习一个从图片集到标签集的函数。这样的函数往往是由一些简单的函数通过组合或复合构成的。线性函数是机器学习中最为常用也是最为简单的函数之一，对于非线性函数，在局部小的范围内也可以看作线性函数。线性函数应用的例子包括线性回归，在这里我们研究曲线拟合问题，通过优化线性权重参数来最大化可能性；神经网络自编码器，用于降维和数据压缩，其中参数是每一层的权值和偏差，通过重复应用链规则来最小化重构误差；高斯混合模型用于数据分布的建模，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。一般需要优化的方法通过学习参数来学习函数。这就需要对各参数求导数或微分。机器学习中的参数，常常是向量或者矩阵，因此需要学习函数对向量或矩阵的求导或微分方法。向量微分是机器学习中最基本的数学工具之一。

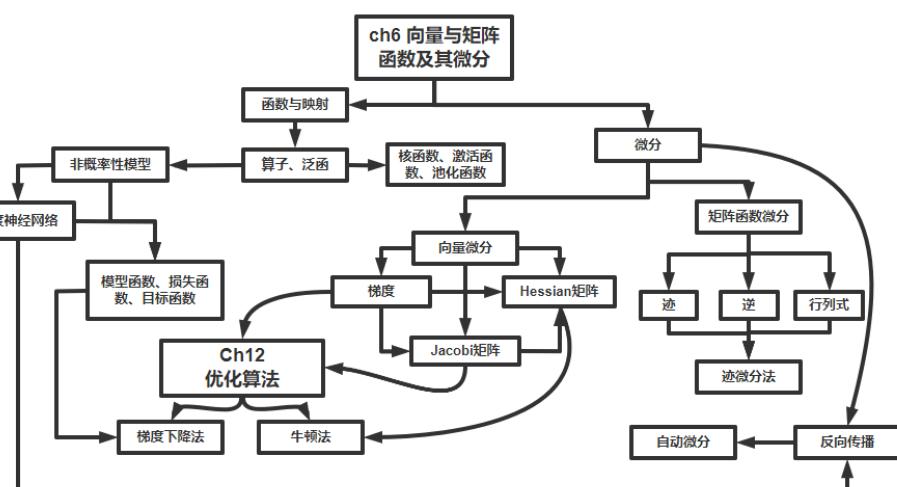


图 6.1: 本章导图

6.1 向量函数和矩阵函数

学习问题是依据经验数据选取所期望的依赖关系的问题，有两种处理学习问题的方法：一是基于经验风险泛函最小化，二是基于估计所期望的随机依赖关系（密度、条件密度和条件概率）。学习过程是一个从给定的函数集中，非概率相关的函数或概率相关的函数，选择一个适当函数的过程。

在机器学习领域，函数集有时也称为假设空间，从数学上，在假设空间中引入恰当的数学结构，可以形成如下空间：

- 距离空间（度量空间）
- 赋范线性空间
- Banach 空间（完备的赋范线性空间）
- 内积空间
- Hilbert 空间（完备的内积空间）
- 欧氏空间（特殊的 Hilbert 空间）

6.1.1 函数

设有两个集合 M 和 N ，如果 M 中每一个元素对应 N 中唯一的一个元素，则我们称这两个集合是通过函数依赖关系相互关联的。

定义 6.1.1. 设 M 和 N 是两非空集合，若有对应法则 T ，使得 M 内每一个元素 x ，都有唯一的一个元素 $y \in N$ 与它相对应，则称 T 是定义在 M 上的函数，记作

$$T : M \rightarrow N, \quad x \mapsto y$$

M 称为 T 的定义域； $T(M) = \{y | y = T(x), x \in M\}$ 称为 T 的值域。

A. 标量值 (scalar-valued function) 函数

定义 6.1.2. 设 M 是一非空集合，当 $N = \mathbb{R}$ 时，函数 $T : M \mapsto \mathbb{R}$ 称为实值函数或标量函数。特别当 $M = N = \mathbb{R}$ 时，函数 $y = T(x)$ 称为一元实值函数或一元函数。当 $M = \mathbb{R}^n, N = \mathbb{R}$ 时，函数 $y = T(x) = T(x_1, x_2, \dots, x_n)$ 称为多元函数。

注：当 $M = \mathbb{R}^{m \times n}, N = \mathbb{R}$ 时，函数 $y = T(A) = T(a_{11}, a_{12}, \dots, a_{nn})$ 也可称为多元函数，此时，我们相当于把矩阵进行了向量化。

例 6.1.1. 假设 a 是一个 n 维向量，我们可以定义关于 n 维向量 x 的标量值函数：

$$f(x) = a^T x,$$

称为内积函数。

定义 6.1.3. 叠加性：对于所有的 n 维向量 \mathbf{x}, \mathbf{y} 和标量 α, β ，例 6.1.1 中定义的函数满足性质：

$$\begin{aligned} f(\alpha\mathbf{x} + \beta\mathbf{y}) &= \mathbf{a}^T(\alpha\mathbf{x} + \beta\mathbf{y}) = \mathbf{a}^T(\alpha\mathbf{x}) + \mathbf{a}^T(\beta\mathbf{y}) \\ &= \alpha(\mathbf{a}^T\mathbf{x}) + \beta(\mathbf{a}^T\mathbf{y}) = \alpha f(\mathbf{x}) + \beta f(\mathbf{y}) \end{aligned}$$

这个性质叫做叠加性。

一个函数如果满足叠加性，则这个函数称为线性函数。因此内积函数是线性函数。

叠加性有时会被拆成两个性质：齐次性：对于任意 n 维向量 \mathbf{x} 和标量 α 有 $f(\alpha\mathbf{x}) = \alpha f(\mathbf{x})$

可加性：对于任意 n 维向量 \mathbf{x}, \mathbf{y} 有 $f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$

如果一个函数 f 是线性的，叠加性可以拓展到多个向量上：

$$f(\alpha_1\mathbf{x}_1 + \cdots + \alpha_k\mathbf{x}_k) = \alpha_1 f(\mathbf{x}_1) + \cdots + \alpha_k f(\mathbf{x}_k)$$

对任意的 n 维向量 $\mathbf{x}_1, \dots, \mathbf{x}_k$ 和标量 $\alpha_1, \dots, \alpha_k$ 成立。

我们看到与一个固定向量做内积的函数是线性的。反过来也是正确的，如果一个函数式线性的，那么它就可以表示为与某个固定的向量做内积的函数。

定理 6.1.1. 假设函数 f 是一个 n 维向量的标量值函数，并且是线性的。那么存在一个 n 维向量 \mathbf{a} 使得 $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$ 对于任意 \mathbf{x} 成立。我们称 $\mathbf{a}^T\mathbf{x}$ 为 f 的内积表示，并且是唯一表示。

证明。(1) 首先证明存在性。我们可以把 \mathbf{x} 表示为 $\mathbf{x} = \mathbf{x}_1\mathbf{e}_1 + \cdots + \mathbf{x}_n\mathbf{e}_n$ 。如果 f 是线性的那么根据叠加性有

$$f(\mathbf{x}) = f(\mathbf{x}_1\mathbf{e}_1 + \cdots + \mathbf{x}_n\mathbf{e}_n) = \mathbf{x}_1 f(\mathbf{e}_1) + \cdots + \mathbf{x}_n f(\mathbf{e}_n) = \mathbf{a}^T\mathbf{x}$$

其中 $\mathbf{a} = (f(\mathbf{e}_1), f(\mathbf{e}_2), \dots, f(\mathbf{e}_n))$ 。

(2) 下证唯一性。我们不妨设 $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$ 并且 $f(\mathbf{x}) = \mathbf{b}^T\mathbf{x}$ 。令 $\mathbf{x} = \mathbf{e}_i$ ，当使用 $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$ 时有 $f(\mathbf{e}_i) = \mathbf{a}^T\mathbf{e}_i = a_i$ 。当使用 $f(\mathbf{x}) = \mathbf{b}^T\mathbf{x}$ 时有 $f(\mathbf{e}_i) = \mathbf{b}^T\mathbf{e}_i = b_i$ 。所以 $a_i = b_i$ 对 $i = 1, \dots, n$ 成立。所以 $\mathbf{a} = \mathbf{b}$ 。□

定义 6.1.4. 一个线性函数加上一个常数叫做仿射函数。函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 是仿射的当且仅当它能够表示成 $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b$ ，其中 \mathbf{a} 是 n 维向量， b 是标量，有时候被叫做偏置项。

例 6.1.2. 比如一个 3 维向量的函数

$$f(\mathbf{x}) = 2.3 - 2\mathbf{x}_1 + 1.3\mathbf{x}_2 - \mathbf{x}_3$$

它的 $b = 2.3, \mathbf{a} = (-2, 1.3, -1)$ 。

定理 6.1.2. 任意仿射函数满足如下约束叠加性：

$$f(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha f(\mathbf{x}) + \beta f(\mathbf{y})$$

其中 \mathbf{x}, \mathbf{y} 是 n 维向量， α, β 是标量，并且 $\alpha + \beta = 1$ 。

证明. 为了证明带约束的叠加性, 我们有:

$$\begin{aligned} f(\alpha \mathbf{x} + \beta \mathbf{y}) &= \mathbf{a}^T(\alpha \mathbf{x} + \beta \mathbf{y}) + b \\ &= \alpha \mathbf{a}^T \mathbf{x} + \beta \mathbf{a}^T \mathbf{y} + (\alpha + \beta)b \\ &= \alpha(\mathbf{a}^T \mathbf{x} + b) + \beta(\mathbf{a}^T \mathbf{y} + b) \\ &= \alpha f(\mathbf{x}) + \beta f(\mathbf{y}) \end{aligned}$$

□

对于线性函数, 叠加性对于任意的 α, β 都成立, 但是对于仿射函数只有它们是仿射组合 (即它们的和为 1) 时才成立。仿射函数的约束叠加性在证明一个函数不是仿射的时候非常有用, 我们只需要寻找向量 \mathbf{x}, \mathbf{y} 和数 α, β 满足 $\alpha + \beta = 1$ 并且验证 $f(\alpha \mathbf{x} + \beta \mathbf{y}) \neq \alpha f(\mathbf{x}) + \beta f(\mathbf{y})$ 即可。例如, 我们可以证明最大值函数不满足约束叠加性。定理6.1.2的结论反过来也是正确的, 任意标量值函数只要满足约束叠加性就是仿射函数。

如果 \mathbf{x} 是标量, 此时函数 $f(\mathbf{x}) = \alpha \mathbf{x} + \beta$ 是一条直线, 仿射函数也被称作是线性函数。但是在标准的数学场景下, 当 $\beta \neq 0$ 时, $f(\mathbf{x}) = \alpha \mathbf{x} + \beta$ 不是 \mathbf{x} 的线性函数, 它是 \mathbf{x} 的仿射函数。在本课程中, 我们将区分线性函数和仿射函数。但是由线性函数和仿射函数定义的机器学习模型我们统称为线性模型。

例 6.1.3. 二次型也是一个非常典型的标量值函数:

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$$

将二次型与仿射函数进行叠加得到:

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$$

这是我们在优化中常常会碰到的。

例 6.1.4. 常见的向量和矩阵范数也是标量值函数:

- 向量范数: $f(\mathbf{x}) = \|\mathbf{x}\|$
- 矩阵范数: $f(\mathbf{A}) = \|\mathbf{A}\|$

例 6.1.5. 常见的以矩阵为自变量的标量值函数:

- 行列式: $f(\mathbf{A}) = |\mathbf{A}|$
- 秩函数: $f(\mathbf{A}) = \text{rank}(\mathbf{A})$
- 迹函数: $f(\mathbf{A}) = \text{Tr}(\mathbf{A})$
- 向量-矩阵-向量积函数: $f(\mathbf{A}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$

注: 前面两个是非线性函数, 后面两个是线性函数。

B. 向量值函数

定义 6.1.5. 设 \mathbb{M} 是一非空集合, 当 $\mathbb{N} = \mathbb{R}^n$ 时, 函数 $T : \mathbb{M} \rightarrow \mathbb{R}^n$ 称为向量值函数, 简称向量函数。

例 6.1.6. 假设 A 是一个 $m \times n$ 矩阵。我们可以定义一个关于 n 维向量 x 的向量值函数:

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad f(x) = Ax,$$

称为矩阵-向量积函数。当 $m = 1$ 时, 其退化为内积函数。

函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 若定义为矩阵-向量积函数 $f(x) = Ax$, 则它是线性函数, 也即满足叠加性:

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

对于 n 维向量 x, y 和标量 α, β 成立。我们可以通过矩阵-向量乘法, 向量-标量乘法来验证叠加性。因此关于 A 的函数 $f(x) = Ax$ 是线性函数。

反过来也是正确的。假设 f 是一个将 n 维向量映射为 m 维向量的函数, 并且是线性的, 则 $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$ 对于所有的 n 维向量 x, y 和所有的标量 α, β 成立, 并且存在一个 $m \times n$ 矩阵 A 使得 $f(x) = Ax$ 对所有 x 成立。

定义 6.1.6. 向量值函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 如果能够写成 $f(x) = Ax + b$ 的形式, 那么 f 是一个仿射函数, 其中 A 是 $m \times n$ 矩阵, b 是 m 维向量。

定理 6.1.3. 函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 是仿射函数当且仅当 $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$ 对于所有的 n 维向量 x, y 和所有的标量 α, β 成立且 $\alpha + \beta = 1$ 。换句话说, 向量的仿射组合具有叠加性。

将仿射函数表示为 $f(x) = Ax + b$ 的形式, 矩阵 A 和向量 b 是唯一的, 并且可以使用 $f(0), f(e_1), \dots, f(e_n)$ 表示, 其中 e_k 是 \mathbb{R}^n 中的单位向量:

$$A = [f(e_1) - f(0), f(e_2) - f(0), \dots, f(e_n) - f(0)], b = f(0).$$

与标量值函数的情形下相同, 只有 $b = 0$ 时仿射函数为线性函数。

非线性向量值函数是不满足叠加性的。

例 6.1.7. 绝对值函数: $f(x) = (|x_1|, |x_2|, \dots, |x_n|)$ 是非线性向量值函数。取 $n = 1, x = 1, y = 0, \alpha = -1, \beta = 0$ 有

$$f(\alpha x + \beta y) = 1 \neq \alpha f(x) + \beta f(y) = -1$$

所以不满足叠加性。

例 6.1.8. 排序函数: f 将 x 的元素降序排列, 是非线性向量值函数 ($n > 1$)。取 $n = 2, x = (1, 0), y = (0, 1), \alpha = \beta = 1$ 则

$$f(\alpha x + \beta y) = (1, 1) \neq \alpha f(x) + \beta f(y) = (2, 0)$$

所以不满足叠加性。

C. 矩阵值函数

定义 6.1.7. 设 \mathbb{M} 和 \mathbb{N} 是两个非空的矩阵集合, 函数 $T : \mathbb{M} \mapsto \mathbb{N}$ 称为矩阵值函数, 简称矩阵函数。

例 6.1.9. 常见的矩阵函数有

- 考虑一个矩阵 $L \in \mathbb{R}^{m \times n}$ 和 $T : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{n \times n}$, $T(L) = L^T L$ 是一个矩阵函数。
- 逆函数: $f(A) = A^{-1}$

6.1.2 算子

定义 6.1.8. 设 X 和 Y 是同一数域 \mathbb{K} 上的线性赋范空间, 若 T 是 X 的某个子集 D 到 Y 中的一个映射, 则称 T 为子集 D 到 Y 中的算子。称 D 为算子 T 的定义域, 或记为 $D(T)$; 并称 Y 的子集 $TD = \{y = T(x), x \in D\}$ 为算子 T 的值域。对于 $x \in D$, 通常记 x 的像 $T(x)$ 为 Tx 。

上面算子的定义, 从狭义的角度是指从一个函数空间到另一个函数空间(或它自身)的映射; 从广义的角度看, 可以把线性赋范空间推广到一般空间, 包括向量空间和内积空间, 或更进一步 Banach 空间和 Hilbert 空间等。当 $X = Y = \mathbb{R}$ 时, 算子 T 就是微积分中的函数, 因此算子是函数概念的推广。

定义 6.1.9. 设 X 和 Y 是同一数域 \mathbb{K} 上的线性赋范空间, $x_0 \in D \subset X$, T 为 D 到 Y 中的算子, 如果 $\forall \epsilon > 0, \exists \delta > 0$, 当 $\|x - x_0\| < \delta$, 有 $\|Tx - Tx_0\| < \epsilon$, 则称算子 T 在点 x_0 处连续。若算子 T 在 D 中每一点都连续, 则称 T 为 D 上的连续算子。

定义 6.1.10. 设 X 和 Y 是同一数域 \mathbb{K} 上的线性赋范空间, $D \subset X$, T 为 D 到 Y 中的算子, 如果 $\forall x, y \in D, \forall \alpha, \beta \in \mathbb{K}$, 有 $T(\alpha x + \beta y) = \alpha T(x) + \beta T(y)$, 则称 T 为 D 上的线性算子。

定义 6.1.11. 设 X 和 Y 是同一数域 \mathbb{K} 上的线性赋范空间, $D \subset X$, $T : D \rightarrow Y$ 为线性算子, 如果存在 $M > 0, \forall x \in D$, 有 $Tx \leq Mx$, 则称 T 为 D 上的线性有界算子, 或称 T 有界。

例 6.1.10. • (1) 恒等算子 $I : X \rightarrow X$ 定义为, $\forall x \in X, Ix = x$.

• (2) 零算子 $0 : X \rightarrow Y$ 定义为, $\forall x \in X, 0x = \theta$.

• (3) 设 $C^{(1)}[a, b]$ 是 $[a, b]$ 上所有一阶导函数连续的函数组成的空间, 微分算子 $D : C^{(1)}[a, b] \rightarrow C[a, b]$ 定义为 $\forall x(t) \in C^{(1)}[a, b]$,

$$Dx = \frac{d}{dt}x(t)$$

• (4) 积分算子 $T : C[a, b] \rightarrow C[a, b]$ 定义为 $\forall x(t) \in C[a, b]$,

$$Tx = \int_a^t x(\tau) d\tau, t \in [a, b]$$

- (5) 设矩阵 $A = (a_{ij})_{m \times n}$, $a \in \mathbb{R}$, 矩阵算子 $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 定义为

$$\forall \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n, T\mathbf{x} = A\mathbf{x} = \mathbf{y}$$

其中 $\mathbf{y} = (y_1, y_2, \dots, y_m)$

例 6.1.11. 验证积分算子 T 为线性有界算子。

积分算子 $T : C[a, b] \rightarrow C[a, b]$ 定义为, $\forall x(t) \in C[a, b], Tx = \int_a^t x(\tau) d\tau \quad t \in [a, b]$ 。

证明. 证明设 $x(t), y(t) \in C[a, b] \quad \alpha, \beta \in \mathbb{R}$, 则

$$T(\alpha x + \beta y) = \int_a^t (\alpha x(\tau) + \beta y(\tau)) d\tau = \alpha \int_a^t x(\tau) d\tau + \beta \int_a^t y(\tau) d\tau = \alpha Tx + \beta Ty$$

$$Tx = \max_{t \in [a, b]} \left| \int_a^t x(\tau) d\tau \right| \leq \max_{t \in [a, b]} \int_a^t |x(\tau)| d\tau \leq \max_{t \in [a, b]} |x(t)| \int_a^t 1 d\tau = \|x(t)\|(b-a)$$

于是积分算子 T 为线性有界算子。 \square

其它常见的算子有: 梯度算子, 散度算子, 拉普拉斯算子, 哈密顿算子等。

每个算子 A 唯一地将集合 \mathbb{M} 中的元素映射到集合 \mathbb{N} 中的元素。这一过程可以用方程表示:

$$A\mathbb{M} = \mathbb{N}$$

我们从算子集中挑选出实现 \mathbb{M} 到 \mathbb{N} 的一对一映射的算子。对于这些算子, 解算子方程

$$Af(t) = F(t)$$

的问题可以看成在 \mathbb{M} 中寻找元素 $f(t)$, 它刚好对应 \mathbb{N} 中的元素 $F(x)$ 。

6.1.3 泛函

定义 6.1.12. 设 X 为实 (或复) 线性赋范空间, 则由 X 到实 (或复) 数域的算子称为泛函。

例 6.1.12. 例如, 若 $x(t)$ 是任意一个可积函数: $x(t) \in L^1[a, b]$, 则其积分

$$f(x) = \int_a^b x(t) dt$$

就是一个定义在 $L^1[a, b]$ 上的泛函, 而且是线性的:

$$f(\alpha x + \beta y) = \alpha \int_a^b x(t) dt + \beta \int_a^b y(t) dt = \alpha f(x) + \beta f(y)$$

还是有界的:

$$|f(x)| \leq \int_a^b |x(t)| dt = \|x\|$$

今后我们一般地仍限于实数范围内讨论泛函。

例 6.1.13. 设 $x(t) \in C[a, b]$, η 是 $[a, b]$ 上任一固定点, 则 $\delta_\eta(x) = x(\eta)$ 是定义在 $C[a, b]$ 上的有界线性泛函。它就是熟知的单位脉冲函数 δ 函数。

例 6.1.14. 令 $J(x) = \int_a^b g(x(t), t) dt$, 其中 g 为二元连续函数。则 $J(x)$ 是定义在 $C[a, b]$ 上的泛函, 但一般地它不是线性的。如果 $g(x, t)$ 的偏导数 g'_x 存在且有界, 则泛函 $J(x)$ 是连续的, 这是因为

$$|J(x_1) - J(x_2)| \leq \int_a^b |g(x_1, t) - g(x_2, t)| dt \leq \int_a^b |g'_x(\eta, t)| dt \|x_1 - x_2\|_{C[a, b]} \leq M \|x_1 - x_2\|$$

例 6.1.15. 设 X 为线性赋范空间, 则 $f(x) = \|x\|$ 是连续泛函, 但非线性。

6.1.4 机器学习中的风险泛函

下面讨论机器学习中寻找函数依赖关系的模型, 称之为从实例学习的模型。模型包括 3 个组成部分 (如图所示):

1. 数据 (实例) 的发生器 G。
2. 目标算子 S (有时称为训练器算子, 或简单地称为训练器)。
3. 学习机器 LM。

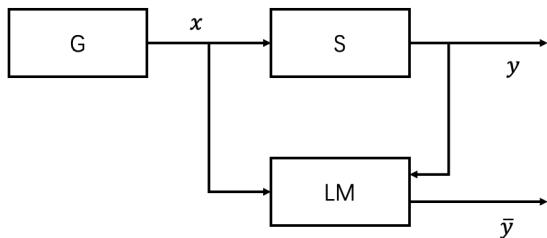


图 6.2: 从实例学习的模型。在学习过程中, 学习机器观测一系列点对 (x, y) (训练集)。训练后, 机器对任何一个给定的 x 必须返回一个 \bar{y} 值。目标是返回一个非常接近于训练器响应 y 的 \bar{y} 。

从实例学习的一般方法过程如下:

首先, 要确定训练器将采用何种类型的算子。假定训练器依据条件分布函数 $F(y|x)$ 返回向量 x 上的输出值 y (它包括了训练器采用函数 $y = f(x)$ 的情形)

学习机器观察训练集, 该训练集是依据联合分布函数 $F(x, y) = F(x)F(y|x)$ 随机独立抽取出来的。利用这一训练集, 学习机器构造对未知算子的逼近, 也即构造一个机器来实现某一固定的函数集。

因此, 学习过程是一个从给定的函数集中选择一个适当函数的过程。如何选择函数将依赖于恰当的评价准则来进行选取。

每当遇到用所期望的评价准则来选取一个函数的问题时, 都可以考虑这样一个模型: 在所有可能的函数中, 找出一个函数, 它以最佳可能方式满足给定的评价准则。

在形式上，这种处理方式的含义是，在向量空间 \mathbb{R}^n 的子集 Z 上，给定一个容许函数集 $|g(z)|, z \in Z$ ，定义一个泛函：

$$R = R(g(z))$$

该泛函就是选取函数的评价准则，然后需要从函数集 $|g(z)|$ 中找出一个最小化泛函的函数 $g^*(z)$ 。

假定泛函的最小值对应于最好的评价，且 $|g(z)|$ 中存在泛函的最小值。在显式地给出函数集 $|g(z)|$ 和泛函 $R(g(z))$ 的情况下，寻找最小化 $R(g(z))$ 的函数 $g^*(z)$ ，这个问题是变分法的研究主题。

我们考虑另外一种情况，即在 Z 上定义概率分布函数 $F(z)$ ，并将泛函定义为数学期望：

$$R(g(z)) = \int L(z, g(z)) dF(z)$$

其中，函数 $L(z, g(z))$ 对任意 $g(z) \in |g(z)|$ 都是可积的。现在的问题是，在未知概率分布 $F(z)$ ，但得到了依据 $F(z)$ 独立地随机抽取出的观测样本

$$z_1, \dots, z_t$$

的情况下，最小化泛函 $R(g(z)) = \int L(z, g(z)) dF(z)$ 。

当用公式给出上述最小化问题时，函数集 $g(z)$ 是以参数的方式给出的： $|g(z, a), a \in \Lambda|$

定义 6.1.13. 函数 $Q(z, a^*) = L(z, g(z, a^*))$ 的期望损失是由下列积分确定的

$$R(a^*) = \int Q(z, a^*) dF(z)$$

这一泛函称为风险泛函或者风险。

当概率分布函数未知，但给定了随机独立观测数据 z_1, \dots, z_t 时，我们的问题是在函数集 $Q(z, a), a \in \Lambda$ 中选取一个最小化风险的函数 $Q(z, a_0)$ 。

给定一个训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

模型 $f(X)$ 关于训练数据集的平均损失称为经验风险或经验损失，记作 R_{emp} ：

$$R_{emp}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)), \quad (6.1)$$

在假设空间、损失函数以及训练数据集确定的情况下，经验风险函数式(6.1)就可以确定。经验风险最小化 (empirical risk minimization, ERM) 的策略认为，经验风险最小的模型是最优的模型。根据这一策略，按照经验风险最小化求最优模型就是求解最优化问题：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)), \quad (6.2)$$

其中， \mathcal{F} 是假设空间。

当样本容量足够大时，经验风险最小化能保证有很好的学习效果。

经验风险最小化时常常会出现过拟合现象，我们可以通过引入所谓的结构风险最小化策略来防止过拟合。

在假设空间、损失函数以及训练数据集确定的情况下，**结构风险 (structural risk)** 定义为在经验风险上加上表示模型复杂度的正则化项或罚项：

$$R_{srm}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f), \quad (6.3)$$

其中 $J(f)$ 为模型的复杂度，是定义在假设空间 \mathcal{F} 上的泛函。模型 f 越复杂，复杂度 $J(f)$ 就越大；反之，模型 f 越简单，复杂度 $J(f)$ 就越小。也就是说，复杂度表示了对复杂模型的惩罚。 $\lambda \geq 0$ 是系数，用以权衡经验风险和模型复杂度。结构风险小需要经验风险与模型复杂度同时小。结构风险小的模型往往对训练数据以及未知的测试数据都有较好的预测。

结构风险最小化 (structural risk minimization, SRM) 策略认为结构风险最小的模型是最优的模型。所以求最优模型，就是求解最优化问题：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f), \quad (6.4)$$

上述最优化问题一般也称为正则化 (regularization)，正则化是结构风险最小化策略的实现。

在数据科学中，我们常常在三个地方遇到向量函数或者矩阵函数。

- 机器学习模型（机器学习模型部分的函数）
- 损失函数（机器学习策略部分的函数）
- 目标函数（机器学习算法部分的函数）

下面我们将分别举一些机器学习中相关的例子，并且着重讲述一些相关的特殊向量函数与矩阵函数。

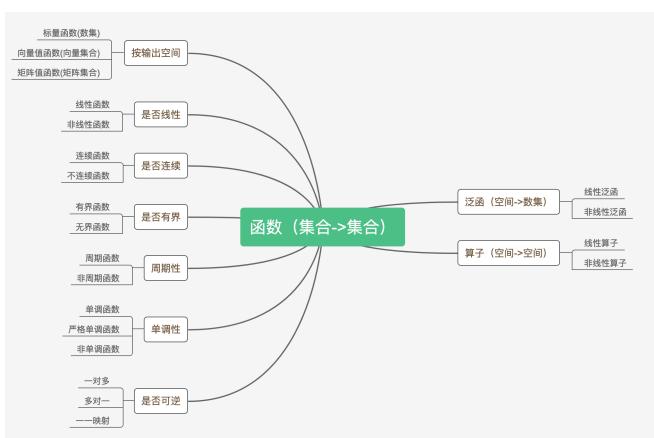


图 6.3: 函数、算子和泛函

草稿勿外传

6.2 统计机器学习中的非概率型函数模型

6.2.1 线性模型中的函数

例 6.2.1. 给定由 d 个属性描述的示例 $\mathbf{x} = (x_1; x_2; \dots; x_d)$, 其中 x_i 是 \mathbf{x} 在第 i 个属性上的取值, 线性模型 (*linear model*) 试图学得一个通过属性的线性组合来进行预测的函数, 即

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b,$$

其中 $\mathbf{w} = (w_1; w_2; \dots; w_d)$ 。 \mathbf{w} 和 b 学得之后, 模型就得以确定。

线性模型中的函数是仿射函数, 并且可以用于机器学习中的回归和分类, 分别对应于线性回归和线性判别。

线性回归模型中的函数有以下三种。给定数据集 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$, 其中 $\mathbf{x}_i = (x_{i1}; x_{i2}; \dots; x_{id})$, $y_i \in \mathbb{R}$ 。

- 模型函数。线性回归试图学得一个线性模型以尽可能准确地预测实值输出标记, 也即学得

$$f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + b \text{ 使得 } f(\mathbf{x}_i) \simeq y_i$$

- 损失函数。如何确定 \mathbf{w} 和 b 呢? 基本策略是把模型预测的结果 $f(x_i)$ 与真实标记 y_i 进行比较, 也即衡量 $f(x)$ 与 y 之间的差别。通过引入损失函数, 均方误差 (也即平方损失, 是回归任务中最常用的性能度量) 来度量:

$$L(f; T) = \frac{1}{m} \sum_{i=1}^m (f(\mathbf{x}_i) - y_i)^2$$

- 目标函数。因此我们可试图让均方误差最小化, 即

$$(\mathbf{w}^*, b^*) = \arg \min_{(\mathbf{w}, b)} \sum_{i=1}^m (f(\mathbf{x}_i) - y_i)^2 = \arg \min_{(\mathbf{w}, b)} \sum_{i=1}^m (\mathbf{w}^T \mathbf{x}_i + b - y_i)^2$$

如果我们令模型的预测值逼近 y 的变形, 比如我们认为示例所对应的输出标记是在指数尺度上变化, 那就可以将输出标记的对数作为线性模型逼近的目标, 即

$$\ln y = \mathbf{w}^T \mathbf{x} + b$$

这个模型叫做对数线性回归。

更一般地, 考虑单调可微函数 $g(\cdot)$, 令

$$y = g^{-1}(\mathbf{w}^T \mathbf{x} + b)$$

这样得到的模型称为广义线性模型, 其中 $g(\cdot)$ 称为“联系函数”。显然, 对数线性回归是广义线性模型在 $g(\cdot) = \ln(\cdot)$ 时的特例。

对二分类任务, 当任务输出标记为 $y \in \{0, 1\}$ 时, 而线性回归模型产生的预测值 $z = \mathbf{w}^T \mathbf{x} + b$ 是实值, 于是我们需要将实值 z 转换为 0/1 值, 可以通过单位阶跃函数 (unit-step function) 来

实现：

$$y = \begin{cases} 0, & z < 0 \\ 0.5, & z = 0 \\ 1, & z > 0 \end{cases}$$

即若预测值 z 大于零就判为正例，小于零则判为反例，预测值为临界值可任意判别。但是，单位阶跃函数不连续，可以使用对数几率函数 $y = \frac{1}{1+e^{-z}}$ 来替代广义线性模型中联系函数的反函数 g^{-1} 。这样我们就可以得到对数几率回归的模型：

$$y = \frac{1}{1 + e^{w^T x + b}}$$

类似对数线性回归，对数几率回归可以变化为

$$\ln \frac{y}{1-y} = w^T x + b$$

如果将 y 视为样本 x 作为正例的可能性，则 $1-y$ 是其反例可能性，两者的比值

$$\frac{y}{1-y}$$

称为几率，反应了 x 作为正例的相对可能性，对几率取对数则得到“对数几率”

$$\ln \frac{y}{1-y}$$

6.2.2 感知机模型中的函数

例 6.2.2. (感知机) 假设输入空间(特征空间)是 $\mathbb{X} \subset \mathbb{R}^n$ ，输出空间是 $\mathbb{Y} = \{+1, -1\}$ 。输入 $x \in \mathbb{X}$ 表示实例的特征向量，对应于输入空间(特征空间)的点；输出 $y \in \mathbb{Y}$ 表示实例的类别。由输入空间到输出空间的如下函数：

$$f(x) = \text{sign}(w^T x + b)$$

称为感知机。其中， w 和 b 为感知机模型参数， $w \in \mathbb{R}^n$ 叫作权值 (weight) 或权值向量 (weight vector)， $b \in \mathbb{R}$ 叫作偏置 (bias)， $w^T x$ 表示 w 和 x 的内积。**sign** 是符号函数，即

$$\text{sign}(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

感知机是一种线性分类模型，属于判别模型。感知机模型的假设空间是定义在特征空间中的所有线性分类模型 (linear classification model) 或线性分类器 (linear classifier)，即函数集合 $\{f | f(x) = w^T x + b\}$ 。

函数模型对应的线性方程 $w^T x + b = 0$ 称为对应于特征空间的分离超平面，它由法向量 w 和截距 b 决定，可用 (w, b) 来表示。分离超平面将特征空间划分为两部分，一部分是正类，一部分是负类。法向量指向的一侧为正类，另一侧为负类。

为了找出这样的超平面，即确定感知机模型参数 w, b ，需要确定一个学习策略，即定义(经验)损失函数并将损失函数极小化。

定义 6.2.1. 数据集的线性可分性 给定一个数据集

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

其中, $x_i \in \mathbb{X} = \mathbb{R}^n$, $y_i \in \mathbb{Y} = \{+1, -1\}$, $i = 1, 2, \dots, N$ 。如果存在某个超平面 S

$$\mathbf{w}^T \mathbf{x} + b = 0$$

能够将数据集的正实例点和负实例点完全正确地划分到超平面的两侧, 即对所有 $y_i = 1$ 的实例 i , 有 $\mathbf{w}^T \mathbf{x}_i + b > 0$, 对所有 $y_i = -1$ 的实例 i , 有 $\mathbf{w}^T \mathbf{x}_i + b < 0$, 则称数据集 T 为线性可分数据集 (*linearly separable dataset*); 否则, 称数据集 T 线性不可分。

假设训练数据集是线性可分的, 为了定义损失函数, 首先写出输入空间 \mathbb{R}^n 中任一点 x_0 到超平面 S 的距离:

$$\frac{1}{\|\mathbf{w}\|} |\mathbf{w}^T \mathbf{x}_0 + b|$$

这里, $\|\mathbf{w}\|$ 是 \mathbf{w} 的 L_2 范数。误分类点 x_i 到超平面 S 的距离是

$$-\frac{1}{\|\mathbf{w}\|} y_i (\mathbf{w}^T \mathbf{x}_0 + b)$$

这样, 假设超平面 S 的误分类点集合为 M , 那么所有误分类点到超平面 S 的总距离为

$$-\frac{1}{\|\mathbf{w}\|} \sum_{x_i \in M} y_i (\mathbf{w}^T \mathbf{x}_0 + b)$$

不考虑 $\frac{1}{\|\mathbf{w}\|}$, 就得到感知机学习的损失函数:

$$L(\mathbf{w}, b) = - \sum_{x_i \in M} y_i (\mathbf{w}^T \mathbf{x}_i + b)$$

其中 M 为误分类点的集合。这个损失函数就是感知机学习的经验风险函数。

感知机学习算法是对以下最优化问题的算法。给定一个训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

其中, $x_i \in \mathbb{X} = \mathbb{R}^n$, $y_i \in \mathbb{Y} = \{+1, -1\}$, $i = 1, 2, \dots, N$ 。求参数 \mathbf{w}, b , 使其为以下损失函数极小化问题的解

$$\min_{\mathbf{w}, b} L(\mathbf{w}, b) = - \sum_{x_i \in M} y_i (\mathbf{w}^T \mathbf{x}_i + b)$$

其中 M 为误分类点的集合。 $L(\mathbf{w}, b)$ 为感知机模型中的目标函数。注: 在感知机模型中, 损失函数和目标函数是一致的。

从空间的角度来理解感知机模型中的函数关系如下图所示。

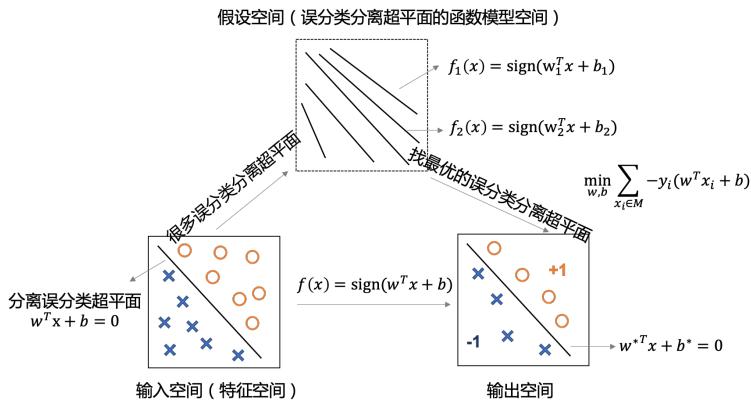


图 6.4: 感知机

6.2.3 支持向量机

支持向量机是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机。按照训练数据的特征，支持向量机分为以下 3 种类型：

- 线性可分支持向量机：当训练数据线性可分时，通过硬间隔最大化学习得到的线性分类器，又称为硬间隔支持向量机
- 线性支持向量机：当训练数据近似线性可分时，通过软间隔最大化学习得到的线性分类器，又称为软间隔支持向量机
- 非线性支持向量机：当训练数据线性不可分时，通过使用核技巧（Kernel trick）及软间隔最大化，学习得到的非线性分类器

考虑一个二类分类问题。假设输入空间与特征空间为两个不同的空间。输入空间为欧氏空间或离散集合，特征空间为欧氏空间或希尔伯特空间。

线性可分支持向量机、线性支持向量机假设这两个空间的元素一一对应，并将输入空间中的输入映射为特征空间中的特征向量。非线性支持向量机利用一个从输入空间到特征空间的非线性映射将输入映射为特征向量。所以，输入都由输入空间转换到特征空间，支持向量机的学习是在特征空间进行的。

其中线性可分支持向量机的模型函数定义如下：

定义 6.2.2. 给定线性可分训练数据集，通过间隔最大化或等价地求解相应的凸二次规划问题学习得到的分离超平面为

$$w^* x + b^* = 0$$

以及相应的分类决策函数

$$f(x) = \text{sign}(\mathbf{w}^*{}^T \mathbf{x} + b^*)$$

称为线性可分支持向量机。

一般来说,一个点距离分离超平面的远近可以表示分类预测的确信程度。在超平面 $\mathbf{w}^T \mathbf{x} + b = 0$ 确定的情况下, $|\mathbf{w}^T \mathbf{x} + b|$ 能够相对地表示点距离超平面的远近。而 $\mathbf{w}^T \mathbf{x} + b$ 的符号与类标记 y 的符号是否一致能够表示分类是否正确。所以可用量 $y(\mathbf{w}^T \mathbf{x} + b)$ 来表示分类的正确性及确信度, 这就是函数间隔 (functional margin) 的概念。

定义 6.2.3. 给定训练数据集 T 和超平面 (\mathbf{w}, b) , 定义超平面关于样本点 (x_i, y_i) 的函数间隔为

$$\hat{\gamma}_i = y_i(\mathbf{w}^T \mathbf{x}_i + b)$$

定义超平面 (\mathbf{w}, b) 关于训练数据集 T 的函数间隔为超平面 (\mathbf{w}, b) 关于 T 中所有样本点 (x_i, y_i) 的函数间隔的最小值, 即

$$\hat{\gamma} = \min_{i=1, \dots, N} \hat{\gamma}_i$$

定义 6.2.4. 给定训练数据集 T 和超平面 (\mathbf{w}, b) , 定义超平面关于样本点 (x_i, y_i) 的几何间隔为

$$\gamma_i = y_i \left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|} \right)$$

定义超平面 (\mathbf{w}, b) 关于训练数据集 T 的几何间隔为超平面 (\mathbf{w}, b) 关于 T 中所有样本点 (x_i, y_i) 的几何间隔的最小值, 即

$$\gamma = \min_{i=1, \dots, N} \gamma_i$$

函数间隔和几何间隔有如下的关系:

$$\begin{aligned} \gamma_i &= \frac{\hat{\gamma}_i}{\|\mathbf{w}\|} \\ \gamma &= \frac{\hat{\gamma}}{\|\mathbf{w}\|} \end{aligned}$$

支持向量机学习的基本想法是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。一般地, 当训练数据集线性可分时, 存在无穷个分离超平面可将两类数据正确分开。感知机利用误分类最小的策略, 求得分离超平面, 不过这时的解有无穷多个, 但是线性可分支持向量机利用几何间隔最大化求最优分离超平面, 这时, 解是唯一的。这里的间隔最大化又称为硬间隔最大化 (与将要讨论的训练数据集近似线性可分时的软间隔最大化相对应)。

间隔最大化的直观解释是: 对训练数据集找到几何间隔最大的超平面意味着以充分大的确信度对训练数据进行分类。也就是说, 不仅将正负实例点分开, 而且对最难分的实例点 (离超

平面最近的点)也有足够大的确信度将它们分开。这样的超平面应该对未知的新实例有很好的分类预测能力。

下面考虑如何求得一个几何间隔最大的分离超平面,即最大间隔分离超平面。具体地,这个问题可以表示为下面的约束最优化问题:

$$\begin{aligned} & \max_{w,b} \gamma \\ \text{s.t. } & y_i(\frac{w}{\|w\|}x_i + \frac{b}{\|w\|}) \geq \gamma, i = 1, 2, \dots, N \end{aligned}$$

考虑几何间隔和函数间隔的关系,可以将问题改写成

$$\begin{aligned} & \max_{w,b} \frac{\hat{\gamma}}{\|w\|} \\ \text{s.t. } & y_i(w^T x_i + b) \geq \hat{\gamma}, i = 1, 2, \dots, N \end{aligned}$$

更进一步,将 $\hat{\gamma} = 1$ 代入上面的最优化问题,注意到最大化 $\frac{1}{\|w\|}$ 和最小化 $\frac{1}{2}\|w\|^2$ 是等价的,于是就得到下面的线性可分支持向量机学习的最优化问题:

$$\begin{aligned} & \min_{w,b} \frac{1}{2}\|w\|^2 \\ \text{s.t. } & y_i(w^T x_i + b) - 1 \geq 0, i = 1, 2, \dots, N \end{aligned}$$

从空间的角度来理解线性可分支持向量机中的函数关系,如下图所示。

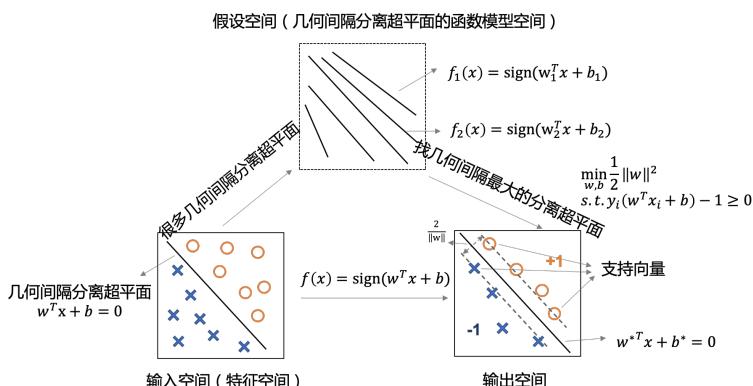


图 6.5: 线性可分支持向量机

线性可分问题的支持向量机学习方法,对线性不可分训练数据是不适用的。线性不可分意味着某些样本点 (x_i, y_i) 不能满足函数间隔大于等于 1 的约束条件。

缓解该问题的一个办法是允许支持向量机在一些样本点上出错。为此要引入“软间隔”的概念,它允许某些样本不满足约束

$$y_i(w^T x_i + b) \geq 1,$$

但是，在最大化间隔的同时，不满足约束的样本应尽可能少。

这样目标函数由原来的 $\frac{1}{2}\|\mathbf{w}\|^2$ 变成

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^N L_{0/1}(y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1)$$

且软间隔优化目标可写为

$$\min_{\mathbf{w}, b} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^N L_{0/1}(y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1),$$

这里， $L_{0/1}$ 是“0/1 损失函数”

$$L_{0/1}(z) = \begin{cases} 1, & \text{if } z < 0 \\ 0, & \text{otherwise} \end{cases};$$

$C > 0$ 称为惩罚参数，一般由应用问题决定， C 值大时对误分类的惩罚增大， C 值小时对误分类的惩罚减小。最小化目标函数包含两层含义：使 $\frac{1}{2}\|\mathbf{w}\|^2$ 尽量小即间隔尽量大，同时使误分类点的个数尽量小， C 是调和二者的系数。显然，当 C 为无穷大时，上式迫使所有样本均满足约束 $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$ ；当 C 取有限值时，上式允许一些样本不满足约束。

然而， $L_{0/1}$ 非凸、非连续，数学性质不太好，使得该优化问题不易直接求解，于是，人们通常用其它一些函数来代替 $L_{0/1}$ ，称为“替代损失”(surrogate loss)。替代损失函数一般具有较好的数学性质，如它们通常是凸的连续函数且是 $L_{0/1}$ 的上界。下面是三种常用的替代损失函数：

- hinge 损失： $L_{hinge}(z) = \max(0, 1 - z)$;
- 指数损失 (exponential loss)： $L_{exp}(z) = \exp(-z)$;
- 对率损失 (logistic loss)： $L_{log}(z) = \log(1 + \exp(-z))$

若采用 hinge 损失，则软件隔优化目标变成

$$\min_{\mathbf{w}, b} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^N \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$$

引入“松弛变量”(slack variables) $\xi_i \geq 0$ ，可重写为

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$$

$$\text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, 2, \dots, N$$

$$\xi_i \geq 0, i = 1, 2, \dots, N$$

这就是常用的线性支持向量机，也称为“软间隔支持向量机”。

线性不可分的线性支持向量机的学习问题变成如下凸二次规划 (convex quadratic programming) 问题：

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$$

$$\text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, 2, \dots, N$$

$$\xi_i \geq 0, i = 1, 2, \dots, N$$

定义 6.2.5. (线性支持向量机) 对于给定的线性不可分的训练数据集, 通过求解凸二次规划问题, 即软间隔最大化问题, 得到的分离超平面为

$$\boldsymbol{w}^*{}^T \boldsymbol{x} + b^* = 0$$

以及相应的分类决策函数

$$f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^*{}^T \boldsymbol{x} + b^*)$$

称为线性支持向量机。

从空间的角度来理解线性支持向量机中的函数关系, 如下图所示。

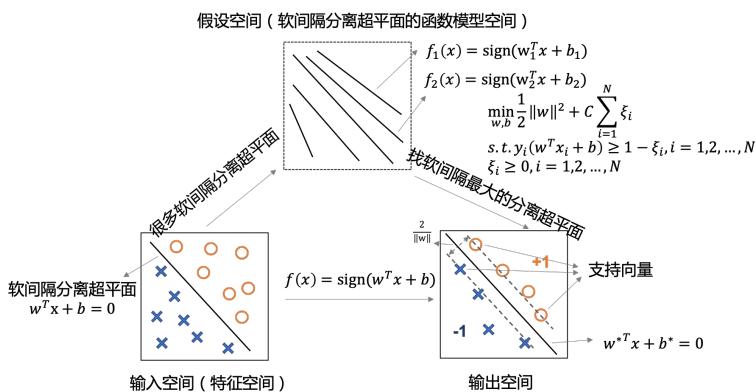


图 6.6: 线性支持向量机

非线性分类问题是通过利用非线性模型才能很好地进行分类的问题。对给定的一个训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, 其中, 实例 x_i 属于输入空间, $x_i \in \mathbb{X} = \mathbb{R}^n$, 对应的标记有两类 $y_i \in \mathbb{Y} = \{+1, -1\}, i = 1, 2, \dots, N$ 。如果能用 \mathbb{R}^n 中的一个超曲面将正负例正确分开, 则称这个问题为非线性可分问题。非线性问题往往不好求解, 所以希望能用解线性分类问题的方法解决这个问题。所采取的方法是进行一个非线性变换, 将非线性问题变换为线性问题, 通过解变换后的线性问题的方法求解原来的非线性问题。

例 6.2.3. 设原空间为 $\mathbb{X} \subset \mathbb{R}^2, \boldsymbol{x} = (x^{(1)}, x^{(2)})^T \in \mathbb{X}$, 新空间为 $\mathbb{Z} \subset \mathbb{R}^2, \boldsymbol{z} = (z^{(1)}, z^{(2)})^T \in \mathbb{Z}$, 定义从原空间到新空间的变换 (映射):

$$\boldsymbol{z} = \phi(\boldsymbol{x}) = ((x^{(1)})^2, (x^{(2)})^2)^T$$

经过变换 $\boldsymbol{z} = \phi(\boldsymbol{x})$, 原空间 $\mathbb{X} \subset \mathbb{R}^2$ 变换为新空间 $\mathbb{Z} \subset \mathbb{R}^2$, 原空间中的点相应地变换为新空间中的点, 原空间中的超曲面 (比如一个椭圆)

$$w_1(x^{(1)})^2 + w_2(x^{(2)})^2 + b = 0$$

变换成为新空间中的直线

$$w_1 z^{(1)} + w_2 z^{(2)} + b = 0$$

在变换后的新空间里，直线 $w_1 z^{(1)} + w_2 z^{(2)} + b = 0$ 可以将变换后的正负实例点正确分开。这样，原空间的非线性可分问题就变成了新空间的线性可分问题。

从上述例子可以看出，用线性分类方法求解非线性分类问题分为两步：

- 首先使用一个变换将原空间的数据映射到新空间；
- 然后在新空间里用线性分类学习方法从训练数据中学习分类模型。

关键问题：如何构造变换？可以用核函数来实现。使用核函数的分类方法称为核技巧或核方法。

定义 6.2.6. 设 \mathbb{X} 是输入空间（欧氏空间 \mathbb{R}^n 的子集或者离散集合），又设 \mathbb{H} 为特征空间（希尔伯特空间），如果存在一个从 \mathbb{X} 到 \mathbb{H} 的映射

$$\phi(x) : \mathbb{X} \rightarrow \mathbb{H}$$

使得对所有 $x, z \in \mathbb{X}$ ，函数 $K(x, z)$ 满足条件

$$K(x, z) = \langle \phi(x), \phi(z) \rangle$$

则称 $K(x, z)$ 为核函数， $\phi(x)$ 为映射函数。

核技巧的想法是：在学习与预测中只定义核函数 $K(x, z)$ ，而不显式地定义映射函数 ϕ 。通常，直接计算 $K(x, z)$ 比较容易，而通过 $\phi(x)$ 和 $\phi(z)$ 计算 $K(x, z)$ 并不容易。注意 ϕ 是输入空间 \mathbb{R}^n 到特征空间 \mathbb{H} 的映射，特征空间 \mathbb{H} 一般是高维的，甚至是无穷维的。此外，对于给定的 $K(x, z)$ ，特征空间 \mathbb{H} 和映射函数 ϕ 的取法并不唯一，可以取不同的特征空间，即便是在同一特征空间也可以取不同的映射。

核函数和映射函数的关系可由下面这个例子可知。

例 6.2.4. 假设输入空间是 \mathbb{R}^2 ，核函数是 $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z})^2$ ，试找出其相关的特征空间 \mathbb{H} 和映射 $\phi(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{H}$ 。

解：取特征空间 $\mathbb{H} = \mathbb{R}^3$ ，记 $\mathbf{x} = (x^{(1)}, x^{(2)})^T, \mathbf{z} = (z^{(1)}, z^{(2)})^T$ ，由于

$$(\mathbf{x}^T \mathbf{z})^2 = (x^{(1)} z^{(1)} + x^{(2)} z^{(2)})^2 = (x^{(1)} z^{(1)})^2 + 2x^{(1)} z^{(1)} x^{(2)} z^{(2)} + (x^{(2)} z^{(2)})^2$$

所以可以取映射 $\phi(\mathbf{x}) = ((x^{(1)})^2, \sqrt{2}x^{(1)}x^{(2)}, (x^{(2)})^2)^T$ ，

容易验证 $\phi(\mathbf{x})^T \phi(\mathbf{z}) = (\mathbf{x}^T \mathbf{z})^2 = K(\mathbf{x}, \mathbf{z})$ 。

仍取 $\mathbb{H} = \mathbb{R}^3$ 以及 $\phi(\mathbf{x}) = \frac{1}{\sqrt{2}}((x^{(1)})^2 - (x^{(2)})^2, 2x^{(1)}x^{(2)}, (x^{(1)})^2 + (x^{(2)})^2)^T$ ，

同样有 $\phi(\mathbf{x})^T \phi(\mathbf{z}) = (\mathbf{x}^T \mathbf{z})^2 = K(\mathbf{x}, \mathbf{z})$ 。

还可以取 $\mathbb{H} = \mathbb{R}^4$ 和 $\phi(\mathbf{x}) = ((x^{(1)})^2, x^{(1)}x^{(2)}, x^{(1)}x^{(2)}, (x^{(2)})^2)^T$ 。

定理 6.2.1. 设 $K : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ 是对称函数, 则 $K(\mathbf{x}, \mathbf{z})$ 为正定核函数的充要条件是对任意 $\mathbf{x}_i \in \mathbb{X}, i = 1, 2, \dots, m$, $K(\mathbf{x}, \mathbf{z})$ 对应的 Gram 矩阵

$$\mathbf{K} = [K(\mathbf{x}_i, \mathbf{x}_j)]_{m \times m}$$

是半正定矩阵。

定义 6.2.7. 设 $\mathbb{X} \subset \mathbb{R}^n$, $K(\mathbf{x}, \mathbf{z})$ 是定义在 $\mathbb{X} \times \mathbb{X}$ 上的对称函数, 如果对于任意 $\mathbf{x}_i \in \mathbb{X}, i = 1, 2, \dots, m$, $K(\mathbf{x}, \mathbf{z})$ 对应的 Gram 矩阵

$$\mathbf{K} = [K(\mathbf{x}_i, \mathbf{x}_j)]_{m \times m}$$

是半正定矩阵, 则称 $K(\mathbf{x}, \mathbf{z})$ 是正定核。

常见的核函数有以下几种:

- 线性核函数: $\kappa(x, z) = x^T z + c$
- 多项式核函数: $\kappa(x, z) = (x^T z)^d$
- 高斯核函数: $\kappa(x, z) = e^{\frac{\|x-z\|^2}{-2\sigma^2}}$
- 拉普拉斯核: $\kappa(x, z) = e^{\frac{\|x-z\|}{-\sigma}}$
- Sigmoid 核: $\kappa(x, z) = \tan(ax^T z + c)$
- 字符串核函数
- ...

核函数的性质有以下三个:

- 若 K_1, K_2 为核函数, 则对于任意正数 γ_1, γ_2 , 其线性组合

$$\gamma_1 K_1 + \gamma_2 K_2$$

是核函数。

- 若 K_1, K_2 为核函数, 则核函数的直积

$$K_1 \otimes K_2(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z})K_2(\mathbf{x}, \mathbf{z})$$

是核函数。

- 若 K_1 为核函数, 则对于任意函数 $g(x)$,

$$K(\mathbf{x}, \mathbf{z}) = g(\mathbf{x})K_1(\mathbf{x}, \mathbf{z})g(\mathbf{z})$$

是核函数。

我们将核技巧应用到支持向量机中, 其基本想法为:

- 通过一个非线性变换将输入空间 (欧氏空间 \mathbb{R}^n 或离散集合) 对应于一个特征空间 (希尔伯特空间 \mathbb{H}), 使得在输入空间 \mathbb{R}^n 中的超曲面模型对应于特征空间 \mathbb{H} 中的超平面模型 (支持向量机)。

- 这样分类问题的学习任务通过在特征空间中求解线性支持向量机就可以完成。
- 在核技巧中，我们并不需要显式地定义映射函数，而是通过核函数来隐式地定义映射函数。
- 在通常情况下，我们只需要将一个线性模型化成带有内积的形式，然后将内积部分替换成核函数即可。

定义 6.2.8. (非线性支持向量机) 从非线性分类训练集，通过核函数与软间隔最大化，学习得到的分类决策函数

$$f(x) = \text{sign}\left(\sum_{i=1}^N \alpha_i^* y_i K(x, x_i) + b^*\right)$$

称为非线性支持向量机， $K(x, z)$ 是正定核函数。

选取适当的核函数 $K(x, z)$ 和适当的参数 C ，构造并求解最优化问题：

$$\begin{aligned} & \min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i \\ & \text{s.t. } \sum_{i=1}^N \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, \dots, N \end{aligned}$$

从空间的角度来理解非线性支持向量机中的函数关系，如下图所示。

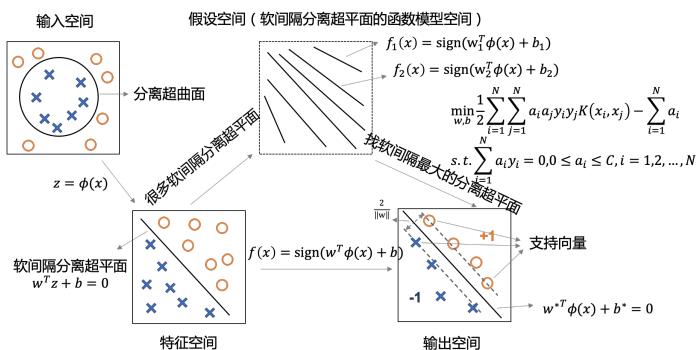


图 6.7: 支持向量机

6.2.4 降维和主成分分析中函数

在数据分析和机器学习领域，在高维情形下所有机器学习方法都面临数据样本稀疏、距离计算困难等问题，称为“维数灾难”(curse of dimensionality)。

维数约简也称降维，是缓解维数灾难的一个重要途径，即通过某种数学变换将原始高维属性空间转变为一个低维“子空间”。

一般来说，欲获得低维子空间，主要有两类方法：

- 线性降维：对原始高维空间进行线性变换，代表性的方法有主成分分析（简称 PCA）。
- 非线性降维：对原始高维空间进行非线性变换，代表性的方法有流形学习。

主成分分析属于多元统计分析的经典方法，首先由 Pearson 于 1901 年提出，但只针对非随机变量，1933 年由 Hotelling 推广到随机变量。

统计分析中，数据的变量之间可能存在相关性，以致增加了分析的难度。于是考虑由少数不相关的变量来代替相关的变量，用来表示数据，并且要求能够保留数据中的大部分信息。

主成分分析主要利用正交变换把由线性相关变量表示的观测数据转换为少数几个由线性无关变量表示的数据，线性无关的变量称为主成分。主成分的个数通常小于原始变量的个数。所以主成分分析属于降维方法。

假设给定 d 维原始空间中的 m 个样本 $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m) \in \mathbb{R}^{d \times m}$ ，主成分分析通过模型函数

$$f(\mathbf{x}) = \mathbf{W}^T \mathbf{x}$$

将 m 个样本变换到 $d' \leq d$ 维子空间中

$$\mathbf{Z} = \mathbf{W}^T \mathbf{X},$$

其中 $\mathbf{W} \in \mathbb{R}^{d \times d'}$ 是正交变换矩阵， $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m) \in \mathbb{R}^{d' \times m}$ 是新空间 (d' 维子空间) 中的 m 个样本，也即原始样本在新空间中的表达。

那么如何选择新的空间（这等价于选择正交变换矩阵 \mathbf{W} ），使得新空间中的 m 个样本是原始空间中的 m 个样本的一个恰当的表达？我们把这个新空间想象成是由超平面张成的，可通过两种策略来选择正交变换矩阵 \mathbf{W} ：

- 最近重构性：样本点到这个超平面的距离都足够近；
- 最大可分性：样本点在这个超平面上的投影能尽可能分开。

可以证明，基于最近重构性和最大可分性这两种策略，能分别得到主成分分析的两种等价推导。我们先从最近重构性来推导，关于最大可分性，我们将在后面概率论部分进行介绍。

假定数据样本进行了中心化，即 $\sum_i \mathbf{x}_i = \mathbf{0}$ ，再假定投影变换后得到的新坐标系为 $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{d'}\}$ ，其中 \mathbf{w}_i 是标准正交基向量。令 $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{d'})$ ，那么 \mathbf{W} 就是一个投影矩阵，所以新样本点在子空间中的坐标为 $\mathbf{z}_i = \mathbf{W}^T \mathbf{x}_i$ 。而这些样本点在原始空间中的坐标为 $\mathbf{WW}^T \mathbf{x}_i$ 。我们在原始空间中考虑原样本点 \mathbf{x}_i 与基于投影重构的样本点 $\mathbf{WW}^T \mathbf{x}_i$ 之间的距离为

$$\|\mathbf{x}_i - \mathbf{WW}^T \mathbf{x}_i\|_2$$

那么考虑整个训练集，对于所有样本总的距离为

$$\|\mathbf{X} - \mathbf{WW}^T \mathbf{X}\|_F,$$

这可以看成投影变换后新样本和原样本之间的损失函数。

在上述过程中，我们得到了损失函数，那我们只需要最小化损失函数即可，这等价于优化

$$\begin{aligned} \min_{\mathbf{W}} \|\mathbf{X} - \mathbf{W}\mathbf{W}^T\mathbf{X}\|_F^2 &= \min_{\mathbf{W}} \text{Tr}((\mathbf{X} - \mathbf{W}\mathbf{W}^T\mathbf{X})^T(\mathbf{X} - \mathbf{W}\mathbf{W}^T\mathbf{X})) \\ &= \min_{\mathbf{W}} \text{Tr}(\mathbf{X}^T\mathbf{X} - 2\mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{X} + \mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{W}\mathbf{W}^T\mathbf{X}) \\ &= \min_{\mathbf{W}} \text{Tr}(-\mathbf{X}^T\mathbf{W}\mathbf{W}^T\mathbf{X}) \end{aligned}$$

最后我们还需要注意 \mathbf{W} 是一个正交矩阵，并且利用迹函数的轮换性

$$\min_{\mathbf{W}} \text{Tr}(-\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W})$$

$$s.t. \mathbf{W}^T\mathbf{W} = \mathbf{I}$$

是主成分分析的优化目标。

假设空间（投影算子空间）

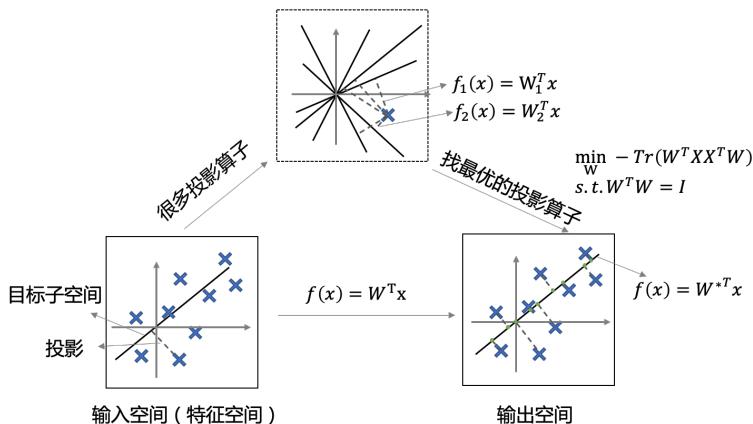


图 6.8: 降维

6.2.5 聚类中的函数

聚类是针对给定的样本，依据它们特征的相似度或距离，将其归并到若干个“类”或“簇”的数据分析问题。一个类是给定样本集合的一个子集。直观上，相似的样本聚集在相同的类，不相似的样本分散在不同的类。这里，样本之间的相似度或距离起着重要的作用。

聚类的目的是通过得到类或簇来发现数据的特点或对数据进行处理，在数据挖掘，模式识别等领域有着广泛的运用。聚类属于无监督学习，因为只是根据样本的相似度或距离来将其进行归类，而类或簇事先并不知道。

聚类算法有很多，主要有两类方法：层次聚类和 k 均值聚类。

下面讲解一下 k 均值聚类中的模型函数。

给定 n 个样本的集合 $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, 每个样本由一个特征向量表示, 特征向量的维数是 m 。 k 均值聚类的目标是将 n 个样本分到 k 个不同的类或簇中, 这里假设 $k < n$ 。 k 个类 G_1, G_2, \dots, G_k 形成对样本集合 \mathbb{X} 的划分, 其中 $G_i \cap G_j = \emptyset, \cup_{i=1}^k G_i = \mathbb{X}$ 。用 C 表示划分, 一个划分对应着一个聚类结果。划分 C 是一个多对一的函数。事实上, 如果把每个样本用一个整数 $i \in \{1, 2, \dots, n\}$ 表示, 每个类也用一个整数 $l \in \{1, 2, \dots, k\}$ 表示, 那么划分或者聚类可以用函数

$$l = C(i)$$

表示, 其中 $i \in \{1, 2, \dots, n\}, l \in \{1, 2, \dots, k\}$ 。所以 k 均值聚类的模型是一个从样本到类的函数。

k 均值聚类归结为样本集合 \mathbb{X} 的划分, 或者从样本到类的函数的选择问题。 k 均值聚类的策略是通过损失函数的最小化选取最优的划分或函数 C^* 。首先, 采用欧氏距离平方 (squared Euclidean distance) 作为样本之间的距离 $d(\mathbf{x}_i, \mathbf{x}_j)$

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^m (x_{ki} - x_{kj})^2 = \|\mathbf{x}_i - \mathbf{x}_j\|^2$$

然后, 定义样本与其所属类的中心之间的距离的总和为损失函数, 即

$$W(C) = \sum_{l=1}^k \sum_{C(i)=l} \|\mathbf{x}_i - \bar{\mathbf{x}}_l\|^2$$

式中 $\bar{\mathbf{x}}_l = (\bar{x}_{1l}, \bar{x}_{2l}, \dots, \bar{x}_{ml})$ 是第 l 个类的均值或中心, $n_l = \sum_{i=1}^n I(C(i) = l), I(C(i) = l)$ 是指示函数, 取值为 1 或 0。函数 $W(C)$ 也称为能量, 表示相同类中的样本相似的程度。

k 均值聚类就是求解最优化问题:

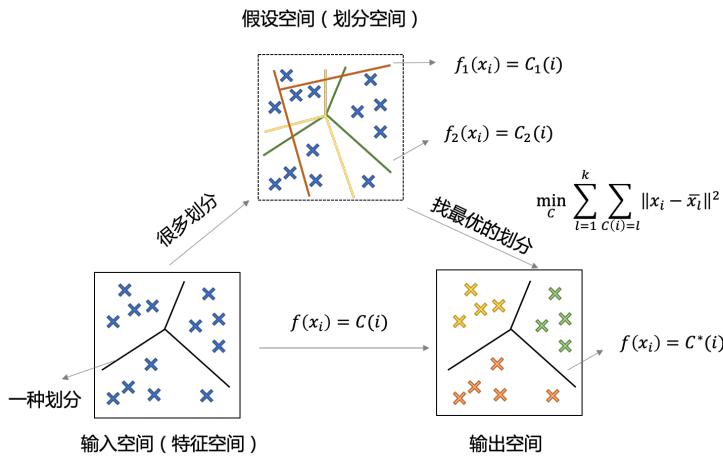
$$C^* = \arg \min_C W(C) = \arg \min_C \sum_{l=1}^k \sum_{C(i)=l} \|V\mathbf{x}_i - \bar{\mathbf{x}}_l\|^2$$

相似的样本被聚到同类时, 损失函数值最小, 这个目标函数的最优化能达到聚类的目的。但是, 这是一个组合优化问题, n 个样本分到 k 类, 所有可能分法的数目是:

$$S(n, k) = \frac{1}{k!} \sum_{l=1}^k (-1)^{k-l} \binom{k}{l} k^n$$

这个数字是指数级的。事实上, k 均值聚类的最优解求解问题是 NP 困难问题。现实中采用迭代的方法求解。

从空间的角度来理解 k 均值聚类中的函数关系, 如下图所示。

图 6.9: k 均值聚类

6.3 深度神经网络中的函数构造

上一节我们已经介绍了统计机器学习中的各种模型函数、损失函数和目标函数的构造。接下来我们介绍深度神经网络中的函数构造。我们首先来回顾一下 2.4 节 MNIST 数字识别这个任务。

例 6.3.1. 在 MNIST 数字识别的任务中，假设我们把训练图像数据集看作 28×28 维向量空间 \mathbb{R}^{784} ，图片向量为 x ；把标签不再看作一个数字，如果标签为 i ，那么我们把它看作只有第 i 个分量为 1，其余分量为 0 的 10 维向量 y ，则所有标签向量在 10 维向量空间 \mathbb{R}^{10} 中。对于训练集中的每个 x ，已知它所代表的数字。我们想要找到一个函数 f （也即分类规则，位于假设空间中），

$$f : \mathbb{R}^{784} \rightarrow \mathbb{R}^{10}$$

$$y' = f(x)$$

将 \mathbb{R}^{784} 维向量空间中的输入，映射到 10 维向量空间中去，每个输入对应的输出在 0 到 9 之间，其中 y' 也是 \mathbb{R}^{10} 中的向量。机器学习试图学习到这个函数，使其适用于（大部分）训练图像，并且在测试集中也能获得好的表现，这一基本要求称为泛化。我们可以通过使 $\|y' - y\|$ 尽可能小，也即求解最优化问题

$$\min \|y' - y\|$$

来找到这个函数。

首先，我们想到这个函数 $f(\mathbf{x})$ 应是 \mathbb{R}^{784} 到 \mathbb{R}^{10} 上的线性函数（一个 $10 \times p$ 矩阵）。十个输出是数字 0 到 9 的概率，我们将通过 10^p 个条目和 M 个训练样本来得到近似正确的结果。

1. 线性函数和线性函数的复合分类手写数字

如果我们令 $f(\mathbf{x}) = \mathbf{Ax}$ 或者令 $f(\mathbf{x}) = f_2(f_1(\mathbf{x})) = \mathbf{A}_2\mathbf{A}_1\mathbf{x} = \mathbf{Ax}$ ，则优化问题变为

$$\min_{\mathbf{A}} \|\mathbf{Ax} - \mathbf{y}\|$$

其中 \mathbf{A} 是参数矩阵，复合函数 $f_2(f_1(\mathbf{x}))$ 表示先用 f_1 将图像映射成 50 维的向量，再用 f_2 将 50 维的向量映射为 10 维的向量。最终我们看到线性映射的复合并不能提高分类的准确性。

2. 仿射函数和仿射函数的复合分类手写数字

如果我们令 $f(\mathbf{x}) = \mathbf{Ax} + \mathbf{b}$ 或者令 $f(\mathbf{x}) = f_2(f_1(\mathbf{x})) = \mathbf{A}_2(\mathbf{A}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2 = \mathbf{Ax} + \mathbf{b}$ ，则优化问题变为

$$\min_{\mathbf{A}, \mathbf{b}} \|\mathbf{Ax} + \mathbf{b} - \mathbf{y}\|$$

其中 \mathbf{A}, \mathbf{b} 是参数矩阵。

但是，线性函数的泛化能力是十分受限的。从艺术角度上看，两个 0 可以构成 8，1 和 0 可以组合成手写体的 9 或是 6，而图像不具有可加性，因而它的输入-输出规则远不是线性的。因此我们考虑用非线性函数以及其复合来分类手写数字。

3. 非线性函数分类手写数字

如果我们令 $f(\mathbf{x}) = \text{ReLU}(\mathbf{Ax} + \mathbf{b})$ ，其中 \mathbf{A}, \mathbf{b} 是参数矩阵， $\text{ReLU}(x) = x_+ = \max(x, 0)$ 是非线性函数，则优化问题变为

$$\min_{\mathbf{A}, \mathbf{b}} \|\text{ReLU}(\mathbf{Ax} + \mathbf{b}) - \mathbf{y}\|$$

- 非线性函数复合分类手写数字

如果我们令 $f(\mathbf{x}) = f_2(f_1(\mathbf{x})) = \text{ReLU}(\mathbf{A}_2\text{ReLU}(\mathbf{A}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2)$ ，其中 $\mathbf{A}_1, \mathbf{b}_1, \mathbf{A}_2, \mathbf{b}_2$ 是参数矩阵， $\text{ReLU}(x) = x_+ = \max(x, 0)$ 是非线性函数，则优化问题变为

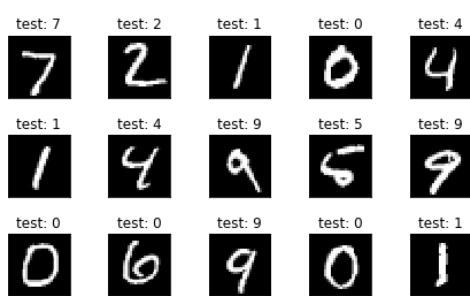
$$\min_{\mathbf{A}_1, \mathbf{A}_2, \mathbf{b}_1, \mathbf{b}_2} \|\text{ReLU}(\mathbf{A}_2\text{ReLU}(\mathbf{A}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) - \mathbf{y}\|$$

从图 6.10 可以看出，线性模型和仿射模型及其复合并不能提高模型的准确率，而引入非线性函数则可以大幅提高模型的准确率。接下来，我们将介绍在深度神经网络模型中，非线性函数的一般构造方法。

6.3.1 深度神经网络模型函数的构造过程

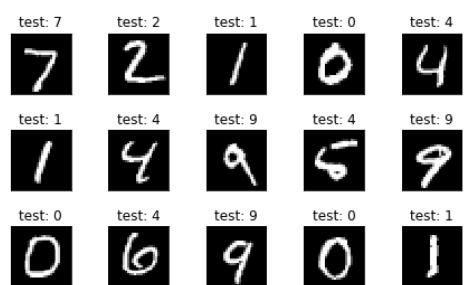
我们在双层非线性网络中使用了 ReLU 函数是一个连续分片线性 (CPL) 函数，这是一个超越预期的成功发现，它把浅层学习转化为深度学习。这里线性是为了保持简单起见，连续性是为了建模一条未知但合理的规则，而分段用于实现真实图像和数据必然要求的非线性。

test acc: 0.8457



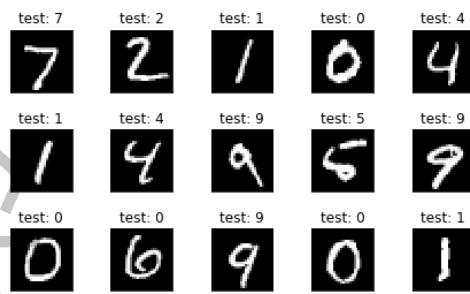
(a) 线性映射分类准确率

test acc: 0.8513



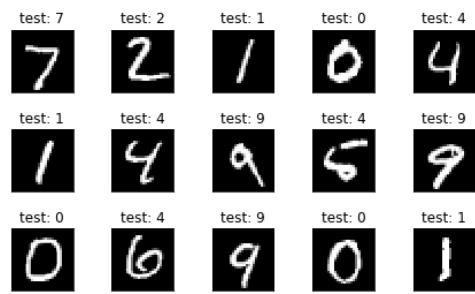
(b) 双层线性网络准确率

test acc: 0.8503



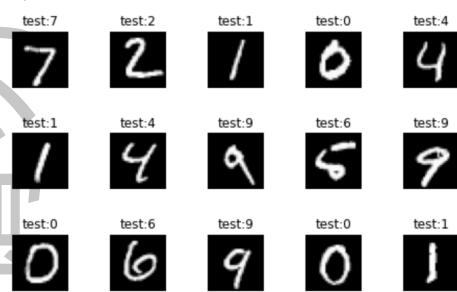
(c) 仿射映射模型分类准确率

test acc: 0.8552



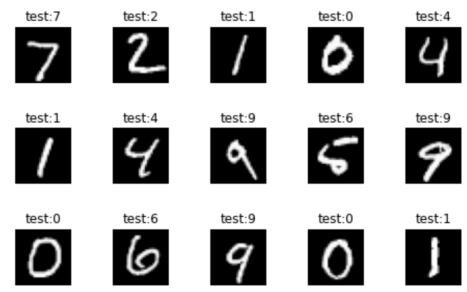
(d) 双层仿射映射复合模型分类准确率

test acc: 0.8894



(e) 非线性函数分类准确率

test acc: 0.9089



(f) 双层非线性网络准确率

图 6.10: 线性映射、双层线性映射、仿射映射、双层仿射映射、非线性函数和双层非线性网络准确率的对比

CPL 函数所在的假设空间是连续分段线性函数空间。这带来了可计算性中的一个关键问题：什么参数能够快速描述一大族 CPL 函数？

定义 6.3.1. 如果函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$, 对于任意一个点 $x \in \mathbb{R}^n$, 存在一个无洞的子集 $\mathbb{I} \subset \mathbb{R}^n$ 包含 x 使得 f 在 \mathbb{I} 上是一个一次函数。则称 f 为分片线性函数。

我们首先来看看连续分片线性 (CPL) 函数的构造。

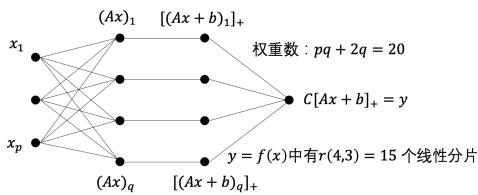


图 6.11: 数据向量 v 的分段线性函数的神经网络架构

上图是数据向量 v 的分段线性函数的初步构造：

首先确定矩阵 A 和向量 b ；

接着将 $Av + b$ 中所有的负分量设为 0(此步是非线性的)。

随后乘上矩阵 C , 得到输出 $w = F(v) = C(Av + b)_+$ 。向量 $(Av + b)_+$ 形成了在输入 v 和输出 w 间的“隐藏层”。

分片线性函数 $ReLU(x) = x_+ = \max(x, 0)$ 与 $\frac{1}{1+e^{-x}}$ 的 Logistic 曲线有类似平滑，通常认为连续导数将有助于优化 A, b, C 的权值，这种想法是合理的，但它被证明是错误的。

在上图中, $(Av + b)_+$ 的每个分量都是双半平面的(由于 $Av + b$ 中负分量处的 0, 其中一个半平面是水平的)。若 A 是 $q \times p$ 的矩阵, 输入空间 \mathbb{R}^P 将被 q 个超平面分割成 r 个部分, 这些分块是可数的, 它度量了整个函数 $F(v)$ 的“表达性”, 其中

$$r(p, q) = C_q^0 + C_q^1 + \cdots + C_q^p$$

这个数字给出了 F 的图像的一个描述, 但是 F 的形式还没有明确给出。

例 6.3.2. 令 $A = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}, C = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, 考虑 $y = F(x) = CReLU(Ax + b)$ 的图像。

可以看到函数的输入空间 \mathbb{R}^2 被两个超平面 $2x_1 - x_2 + 1 = 0, -x_1 + x_2 + 2 = 0$ 划分成了四个区域。函数 $y = F(x)$ 在每一个区域中约束为一个线性函数。

要想获得对数据更好的表达能力, 我们需要更复杂的函数 F 。构造一个更加复杂的 F 最好的方法是通过复合运算, 从简单函数中创造复杂函数。每个 F_i 都是对线性的(或仿射的)函数施

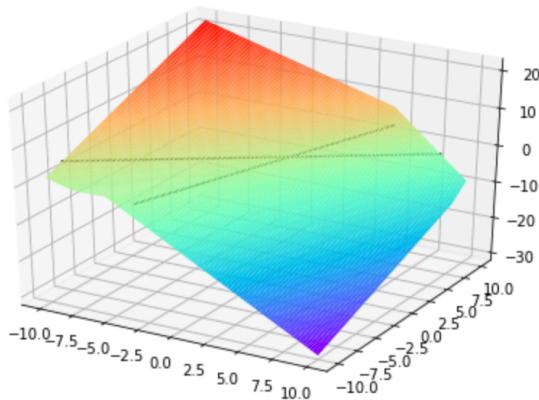


图 6.12: 一层的神经网络

加 ReLU，即 $F_i(\mathbf{x}) = (\mathbf{A}_i \mathbf{x} + \mathbf{b}_i)_+$ 是非线性的，它们的复合是 $F(\mathbf{x}) = \mathbf{C}F_L(F_{L-1}(\dots F_2(F_1(\mathbf{x}))))$ ，在最终输出层之前，得到了 L 个隐藏层。随着 L 的增加，网络将会变得更深。

例 6.3.3. 考虑一个具有三个隐藏层的神经网络，其中

$$F_1 = \text{ReLU} \left(\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 1 \\ -2 \end{pmatrix} \right),$$

$$F_2 = \text{ReLU} \left(\begin{pmatrix} 1 & 2 \\ -2 & -3 \end{pmatrix} \mathbf{x} + \begin{pmatrix} -1 \\ 2 \end{pmatrix} \right),$$

$$F_3 = \text{ReLU} \left(\begin{pmatrix} 2 & 4 \\ -2 & 3 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 1 \\ 2 \end{pmatrix} \right)$$

复合后得：

$$F(\mathbf{x}) = \begin{pmatrix} -1 & 1 \end{pmatrix} F_3(F_2(F_1(\mathbf{x}))),$$

其图像如图所示。

6.3.2 激活函数

神经网络是一个非线性模型。其中非线性是通过激活函数来提供。记神经网络中每一层的函数为 $F_1, F_2, F_3, \dots, F_n$ ，权重 \mathbf{W} 是连接各层，并且将在训练 F 的时候被更新。向量 $\mathbf{x} = \mathbf{x}_0$ 来自训练集，函数 F_k 在第 k 层产生了向量 \mathbf{x}_k 。通常 F_k 由两部分组成，首先是线性部分，比如 $\mathbf{Ax} + \mathbf{b}$ 或者卷积，然后再通过激活函数或者池化函数作用变成一个非线性函数。在训练神经网络过程中，我们通常使用随机梯度下降。为了做到这一点，我们就需要链式法则和对向量函数或者矩阵函数求梯度，后者我们将在后面一节详细讲述。

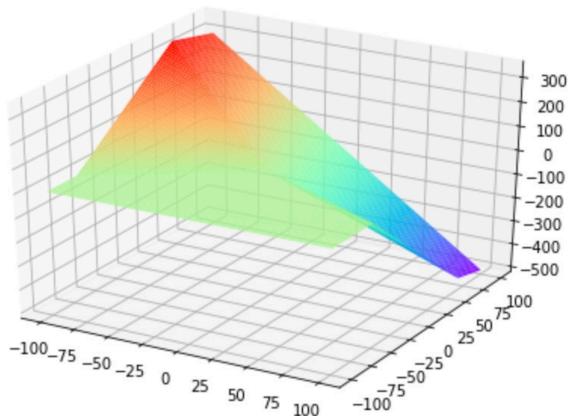


图 6.13: 三个隐藏层的神经网络

定义 6.3.2. 激活函数是一类非线性函数，其满足以下性质：

- 连续并可导(允许少数点上不可导)的非线性函数。
- 激活函数本身及其导数计算简单。
- 激活函数的导函数的值域要在一个合适的区间内。

常见的激活函数：ReLU 型函数

ReLU 函数是目前最常用的激活函数，它有多种不同的变体。

- ReLU(Rectified Linear Unit, 修正线性单元):

$$\text{ReLU}(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} = \max(0, x)$$

- 带泄露的 ReLU(LeakyReLU):

$$\text{LeakyReLU}(x) = \begin{cases} x & x > 0 \\ \gamma x & x \leq 0 \end{cases} = \max(0, x) + \gamma \min(0, x)$$

- 带参数的 ReLU(Parametric ReLU, PReLU):

$$\text{PReLU}_i(x) = \begin{cases} x & x > 0 \\ \gamma_i x & x \leq 0 \end{cases} = \max(0, x) + \gamma_i \min(0, x)$$

上面三种激活函数都是分片线性函数。所以如果一个神经网络中只用这类激活函数，那么最终得到的模型函数也是分片线性函数。他们只需要进行加、乘和比较的操作，计算上非常高效。ReLU 函数被认为有生物上的解释性，比如单侧抑制、宽兴奋边界（即兴奋程度也可以非常

高)。ReLU 函数的缺点是输出是非零中心化的, 给后一层的神经网络引入偏置偏移, 会影响梯度下降的效率。ReLU 神经元指采用 ReLU 作为激活函数的神经元。

此外, ReLU 神经元在训练时比较容易“死亡”。在训练时, 如果参数在一次不恰当的更新后, 第一个隐藏层中的某个 ReLU 神经元在所有的训练数据上都不能被激活, 那么这个神经元自身参数的梯度永远都会是 0。在实际使用中, 为了避免上述情况, 我们就可以使用 LeakyReLU 和 PReLU。LeakyReLU 在输入 $x < 0$ 时, 保持一个很小的梯度 λ 。这样当神经元非激活时也能有一个非零的梯度可以更新参数, 避免永远不能被激活。而 PReLU 则引入一个可学习的参数, 可以使得不同神经元可以有不同的参数。

常见的激活函数: Sigmoid 型函数

- **Logistic 函数**, 其具有形式:

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

- **Tanh 函数**, 其具有形式:

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)},$$

Tanh 函数可以看作是放大并平移的 Logistic 函数, 其值域为 $(-1, 1)$, 它与 Logistic 函数满足如下关系:

$$\tanh(x) = 2\sigma(2x) - 1$$

Logistic 函数也叫 Sigmoid 函数, 因此我们把它和 Tanh 函数统称为 Sigmoid 型函数。

定义 6.3.3. 对于函数 $f(x)$, 若 $x \rightarrow -\infty$ 时, 其导数 $f'(x) \rightarrow 0$, 则称其为左饱和。若 $x \rightarrow +\infty$ 时, 其导数 $f'(x) \rightarrow 0$, 则称其为右饱和。当同时满足左、右饱和时, 就称为两端饱和。

定理 6.3.1. Sigmoid 型函数具有饱和性。

Sigmoid 型激活函数会导致一个非稀疏的神经网络, 但是 ReLU 具有很好的稀疏性。相对于 ReLU, Logistic 有更好的光滑性, 通常认为连续导数将有助于优化模型, 这种想法是合理的, 但它被证明是错误的, 因为饱和性容易导致梯度消失。

“挤压”函数

Logistic 函数可以看成是一个“挤压”函数, 把一个实数域的输入“挤压”到 $(0, 1)$ 。当输入值在 0 附近时, Sigmoid 型函数近似为线性函数; 当输入值靠近两端时, 对输入进行抑制。输入越小, 越接近于 0; 输入越大, 越接近于 1。

因为 Logistic 函数的性质, 使得装备了 Logistic 激活函数的神经元具有以下两点性质:

- 其输出直接可以看作是概率分布, 使得神经网络可以更好地和统计学习模型进行结合。
- 其可以看作是一个软性门 (Soft Gate), 用来控制其他神经元输出信息的数量。

Logistic 函数和 Tanh 函数计算开销较大。因为这两个函数都是在中间（0 附近）近似线性，两端饱和，因此这两个函数可以通过分段函数来近似。

Logistic 函数的近似

因为 Logistic 函数的导数为 $\sigma'(x) = \sigma(x)(1-\sigma(x))$ ，所以 Logistic 函数在 0 附近的一阶泰勒展开（Taylor expansion）为

$$g_l(x) = \sigma(0) + x \times \sigma'(0) = 0.25x + 0.5$$

这样 Logistic 函数可以用分段函数 hard-logistic(x) 来近似

$$\begin{aligned} \text{hard-logistic}(x) &= \begin{cases} 1 & g_l(x) \geq 1 \\ g_l(x) & 0 < g_l(x) < 1 \\ 0 & g_l(x) \leq 0 \end{cases} \\ &= \max(\min(g_l(x), 1), 0) \quad = \max(\min(0.25x + 0.5, 1), 0) \end{aligned}$$

Tanh 函数的近似

同样，Tanh 函数在 0 附近的一阶泰勒展开为

$$g_t(x) = \tanh(0) + x \times \tanh'(0) = x$$

这样 Tanh 函数也可以用分段函数 hard-tanh(x) 来近似。

$$\text{hard-tanh}(x) = \max(\min(x, 1), -1)$$

其他一些激活函数

我们再列举一些其他的激活函数。

- ELU (Exponential Linear Unit, 指数线性单元) :

$$\text{ELU}(x) = \begin{cases} x & x > 0 \\ \gamma(\exp(x) - 1) & x \leq 0 \end{cases} = \max(0, x) + \min(0, \gamma(\exp(x) - 1))$$

- Softplus 函数

$$\text{Softplus}(x) = \log(1 + \exp(x))$$

Softplus 函数其导数刚好是 Logistic 函数。Softplus 函数虽然也具有单侧抑制、宽兴奋边界的特点，却没有稀疏激活性。

- Swish 函数

$$\text{Swish}(x) = x\sigma(\beta x)$$

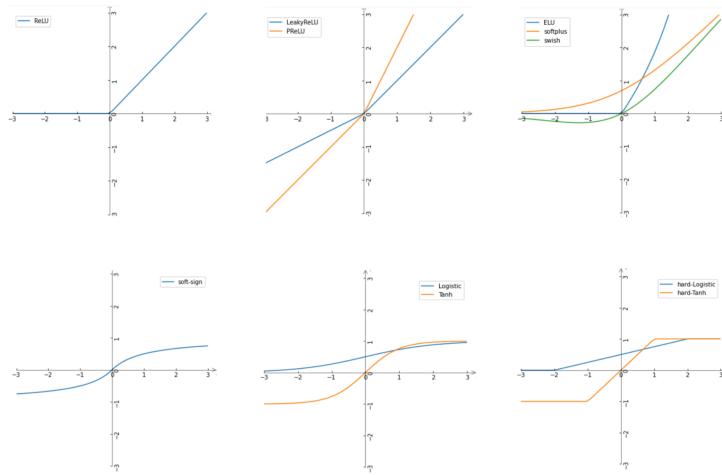


图 6.14: 激活函数一览

6.3.3 跨步下采样与池化

我们除了使用激活函数来捕获数据的非线性，有时我们还可以通过跨步和下采样来实现对数据降维。假设我们从长度为 128 的一维信号开始，我们希望过滤该信号，也即将向量乘以权重矩阵 A ，将长度减小为 64。为达到这个目标，可以通过：

1. 下采样过滤：将 128 维向量 v 乘以 A ，然后丢弃输出的奇数分量，这样输出为 $(\downarrow 2)Av$ 。
2. 跨步过滤：丢弃奇数行矩阵 A ，得到又短又宽的新矩阵 A_2 : 64 行和 128 列。此时，过滤的“步幅”为 2。现在将 128 分量向量 v 乘以 A_2 ，然后得到 A_2v ，这和 $(\downarrow 2)Av$ 是一样的。如果步幅为 3，将在每 3 个分量中保留一个分量。

显然，跨步方法更为有效。如果步幅为 4，则将每一个维度总分量除以 4。在两个维度（对于图像）的数表中，尺寸将减小 16 倍。而下采样则清楚地表明，丢失了一半或者四分之三的信息。跨步和下采样能降低数据的维度，但是会丢失数据的重要信息。

还有一种将维数从 128 减少到 64 的方法，但是却能减少破坏重要信息的风险：池化（pooling），它是某一种形式的下采样。

- 对于图像（二维信号），我们可以在每 2×2 平方像素上使用池化。那么在每个维度上元素减少一半，而对于图像来说总的像素减少 4 倍。从而导致隐藏层上神经元的数量会变为原来的 1/4，这可以加快训练速度。
- 池化使得尺寸减小，除了减少计算量，还减少了过拟合的可能性。池化函数会不断地减小数据的空间大小，因此参数的数量和计算量也会下降，这在一定程度上控制了过拟合。

- 池化包括最大池化和平均池化。

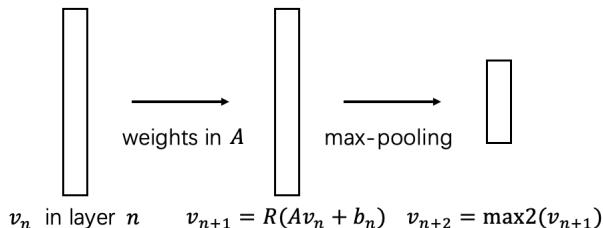


图 6.15: 最大池化

如果将向量 \mathbf{v} 乘以 A , 然后进行下采样时, 是从每若干个元素取出其中的最大值作为结果的方式叫做最大池化。注意: 最大合并简单而又快速, 但不是线性操作, 这是减少尺寸, 纯粹和简单的明智方法。如果将向量 \mathbf{v} 乘以 A , 然后进行下采样时, 是将每若干个元素的平均值作

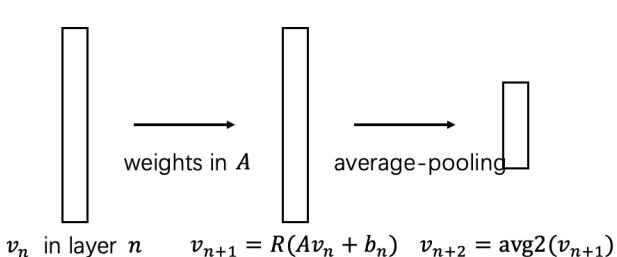


图 6.16: 平均池化

为结果的方式叫做平均池化。平均池化将保持每个池中的平均值, 并且平均池化是线性的。

最后我们简单总结一下一个一般的神经网络构造方式:

记神经网络中每一层的函数为 $F_1, F_2, F_3, \dots, F_n$, 权重 \mathbf{W} 是连接各层, 并且将在训练 F 的时候被更新。向量 $\mathbf{x} = \mathbf{x}_0$ 来自训练集, 函数 F_k 在第 k 层产生了向量 \mathbf{x}_k 。通常 F_k 由两部分组成, 首先是线性部分, 比如 $\mathbf{Ax} + \mathbf{b}$ 或者卷积, 然后再通过激活函数作用变成一个非线性函数。神经网络中最核心的操作就是函数的复合, 我们最终得到的模型 F 就是一系列函数的复合 $F(\mathbf{x}) = F_n(\dots F_2(F_1(\mathbf{x})))$ 。

神经网络模型是一个非线性模型, 其中非线性主要是通过激活函数来提供。在训练神经网络过程中, 我们通常使用随机梯度下降。为了做到这一点, 我们就需要链式法则和对向量函数或者矩阵函数求梯度, 我们将在下面详细讲述。

如果从空间的角度来理解神经网络中的函数关系，则可以总结为下图。

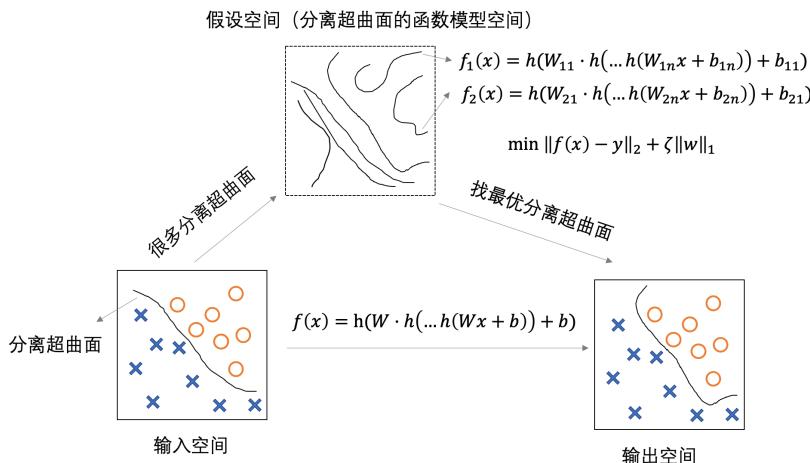


图 6.17: 神经网络

6.4 向量值函数和矩阵微分

在机器学习中，一个机器学习模型的求解通常会转变成一个优化问题：

- 例 6.4.1. • 逻辑回归对应的优化问题：

$$\min_{\mathbf{w}} \sum_{i=1}^N [y_i(\mathbf{w}^T \mathbf{x}_i) - \log(1 + \exp(\mathbf{w}^T \mathbf{x}_i))]$$

- 线性可分支持向量机模型对应的优化问题：

$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 \\ & s.t. y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 > 0 \end{aligned}$$

- PCA 对应的优化问题：

$$\begin{aligned} & \min_{\mathbf{W}} -\text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}) \\ & s.t. \mathbf{W}^T \mathbf{W} = \mathbf{I} \end{aligned}$$

- 例 6.4.2. 在深度学习中我们可能会构造一个两层的神经网络

$$\mathbf{h} = \text{ReLU}(\mathbf{A}_1 \mathbf{x} + \mathbf{b}_1)$$

草稿请勿外传

$$\mathbf{y}' = \text{ReLU}(\mathbf{A}_2 \mathbf{h} + \mathbf{b}_2)$$

并且我们有关于数据集的标签向量 \mathbf{y} , 那么我们需要求解以下优化问题:

$$\min \|\mathbf{y} - \mathbf{y}'\|_2^2$$

上述例子中优化的目标函数都是向量函数或者矩阵函数, 优化问题的求解通常都需要利用到函数的梯度信息, 对于像牛顿法这种二阶方法还需要知道函数的 Hessian 矩阵, 而且这些函数都是多元函数, 含有的变量非常多。

例如在深度学习领域, 2019 年 OpenAI 开放了一个文本生成模型 GPT-2, 有 7.74 亿个参数, 而完整模型则有 15 亿的参数, 这就意味着我们需要求解同等规模的梯度, 如果要一个一个去计算他们的偏导数是不可能的。

本节将主要介绍如何使用一些较为方便的方法来求解梯度或者 Hessian 矩阵。

6.4.1 向量函数的梯度

我们先回顾一下一元函数的导数的相关概念:

定义 6.4.1. 函数 $f : \mathbb{R} \rightarrow \mathbb{R}$ 关于 x 的导数定义为

$$\frac{f}{x} := \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

定义 6.4.2. 函数 $f : \mathbb{R} \rightarrow \mathbb{R}$ 在 x_0 的 n 阶泰勒多项式为

$$T_n(x) := \sum_{k=0}^n \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

定义 6.4.3. 光滑函数 $f : \mathbb{R} \rightarrow \mathbb{R}$, $f \in C^\infty$ 在 x_0 处的泰勒级数为

$$T_\infty(x) := \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

定义 6.4.4. 函数 $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ 关于 \mathbf{x} 的 n 个分量的偏导为

$$\frac{\partial f}{\partial x_1} = \lim_{h \rightarrow 0} \frac{f(x_1 + h, x_2, \dots, x_n) - f(x)}{h}$$

⋮

$$\frac{\partial f}{\partial x_n} = \lim_{h \rightarrow 0} \frac{f(x_1, x_2, \dots, x_n + h) - f(x)}{h}$$

多元函数的梯度可以看作一元函数的导数的推广。

相对于 $n \times 1$ 向量 \mathbf{x} 的梯度算子记作 $\nabla_{\mathbf{x}}$, 定义为

$$\nabla_{\mathbf{x}} \stackrel{\text{def}}{=} \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_n} \right]^T = \frac{\partial}{\partial \mathbf{x}} \quad (6.5)$$

因此, 以 $n \times 1$ 实向量 \mathbf{x} 为变元的实值函数 $f(\mathbf{x})$ 相对于 x 的梯度为一 $n \times 1$ 列向量, 定义为

禁
止
复
制
外
传

定义 6.4.5. 若 $\mathbf{x} \in \mathbb{R}^n, f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ 是一实值函数, 其中 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, 则定义

$$\frac{\partial}{\partial \mathbf{x}} f = \nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

梯度方向的负方向称为变元 \mathbf{x} 的梯度流 (gradient flow), 记作

$$\dot{\mathbf{x}} = -\nabla_{\mathbf{x}} f(\mathbf{x}) \quad (6.6)$$

从梯度的定义式可以看出:

- (1) 一个以向量为变元的标量函数的梯度为一向量。
- (2) 梯度的每个分量给出标量函数在分量方向上的变化率。

梯度向量最重要的性质之一是, 它指出了当变元增大时函数 f 的最大增大率。相反, 梯度的负值 (简称负梯度) 指出了当变元增大时函数 f 的最大减小率。根据这样一种性质, 即可设计出求一函数极小值的迭代算法, 这将在后面详细讨论。

例 6.4.3. 假设函数 $f(\mathbf{x}) : \mathbb{R}^2 \rightarrow \mathbb{R}$ 为

$$f(\mathbf{x}) = \sin x_1 + 2x_1 x_2 + x_2^2$$

其中 $\mathbf{x} = (x_1, x_2)^T$, 则 f 的偏导数分别为

$$\begin{aligned} \frac{\partial f}{\partial x_1}(\mathbf{x}) &= \cos x_1 + 2x_2 \\ \frac{\partial f}{\partial x_2}(\mathbf{x}) &= 2x_1 + 2x_2 \end{aligned}$$

因此梯度为

$$\nabla f(\mathbf{x}) = (\cos x_1 + 2x_2, 2x_1 + 2x_2)^T$$

例 6.4.4. 设 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n, \mathbf{a} = (a_1, a_2, \dots, a_n)^T \in \mathbb{R}^n, \mathbf{b} = (b_1, b_2, \dots, b_n)^T \in \mathbb{R}^n$ 以及 $f(x_1, x_2, \dots, x_n) = f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} + \mathbf{b}$, 求 $f(\mathbf{x})$ 的梯度 $\nabla f(\mathbf{x})$ 。将 $f(\mathbf{x})$ 写成分量的形式:

$$f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} + \mathbf{b} = \sum_{i=1}^n a_i x_i + b_i$$

那么 $f(\mathbf{x})$ 对第 h 个分量的偏导数为

$$\frac{\partial(\mathbf{a}^T \mathbf{x} + \mathbf{b})}{\partial x_h} = a_h$$

从而就有

$$\nabla f = \mathbf{a}$$

例 6.4.5. 设 $\mathbf{p} \in \mathbb{R}^n$ 是 \mathbb{R}^n 中的一个点, 函数 $f(\mathbf{x})$ 表示点 \mathbf{x} 和 \mathbf{p} 的距离:

$$f(\mathbf{x}) = \|\mathbf{x} - \mathbf{p}\|_2 = \sqrt{\sum_{i=1}^n (x_i - p_i)^2}$$

函数 $f(\mathbf{x})$ 在 $\mathbf{x} \neq \mathbf{p}$ 处处可微, 并且梯度为

$$\nabla f(\mathbf{x}) = \frac{1}{\|\mathbf{x} - \mathbf{p}\|_2} (\mathbf{x} - \mathbf{p})$$

标量对向量求梯度的通用做法是对每一个变元求导数。但是, 对于一些简单的, 通过矩阵向量乘积运算得到的标量函数, 有更为简单更为形式化的运算公式。

例 6.4.6. 若 A 和 y 均与向量 \mathbf{x} 无关, 则

$$\frac{\partial \mathbf{x}^T A \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^T}{\partial \mathbf{x}} A \mathbf{y} = A \mathbf{y}$$

例 6.4.7. 注意到 $\mathbf{y}^T A \mathbf{x} = \langle A^T \mathbf{y}, \mathbf{x} \rangle = \langle \mathbf{x}, A^T \mathbf{y} \rangle = \mathbf{x}^T A^T \mathbf{y}$, 故

$$\frac{\partial \mathbf{y}^T A \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^T A^T \mathbf{y}}{\partial \mathbf{x}} = A^T \mathbf{y}$$

例 6.4.8. 由于

$$\mathbf{x}^T A \mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n A_{ij} x_i x_j$$

可求出梯度 $\frac{\partial \mathbf{x}^T A \mathbf{x}}{\partial \mathbf{x}}$ 的第 k 个分量为

$$\left[\frac{\partial \mathbf{x}^T A \mathbf{x}}{\partial \mathbf{x}} \right]_k = \frac{\partial}{\partial x_k} \sum_{i=1}^n \sum_{j=1}^n A_{ij} x_i x_j = \sum_{i=1}^n A_{ik} x_i + \sum_{j=1}^n A_{kj} x_j$$

即有

$$\frac{\partial \mathbf{x}^T A \mathbf{x}}{\partial \mathbf{x}} = A \mathbf{x} + A^T \mathbf{x}$$

特别地, 若 A 为对称矩阵, 则

$$\frac{\partial \mathbf{x}^T A \mathbf{x}}{\partial \mathbf{x}} = 2A \mathbf{x}$$

例 6.4.8 通过将矩阵计算展开, 求得标量函数对各分量的导数。但对更为复杂的函数计算时将非常困难, 在之后, 我们会学习迹微分法, 可以让求导运算非常简单。归纳以上三个例子的结果及其他结果, 便得到实值函数 $f(\mathbf{x})$ 相对于列向量 \mathbf{x} 的下述几个常用的梯度公式。

(1) 若 $f(\mathbf{x}) = c$ 为常数, 则梯度 $\frac{\partial c}{\partial \mathbf{x}} = \mathbf{0}$ 。

(2) 线性法则: 若 $f(\mathbf{x})$ 和 $g(\mathbf{x})$ 分别是向量 \mathbf{x} 的实值函数, c_1 和 c_2 为实常数, 则

$$\frac{\partial[c_1f(\mathbf{x}) + c_2g(\mathbf{x})]}{\partial \mathbf{x}} = c_1\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} + c_2\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \quad (6.7)$$

(3) 乘积法则:

□ 若 $f(\mathbf{x})$ 和 $g(\mathbf{x})$ 都是向量 \mathbf{x} 的实值函数, 则

$$\frac{\partial f(\mathbf{x})g(\mathbf{x})}{\partial \mathbf{x}} = g(\mathbf{x})\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} + f(\mathbf{x})\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \quad (6.8)$$

□ 若 $f(\mathbf{x}), g(\mathbf{x})$ 和 $h(\mathbf{x})$ 都是向量 \mathbf{x} 的实值函数, 则

$$\frac{\partial f(\mathbf{x})g(\mathbf{x})h(\mathbf{x})}{\partial \mathbf{x}} = g(\mathbf{x})h(\mathbf{x})\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} + f(\mathbf{x})h(\mathbf{x})\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} + f(\mathbf{x})g(\mathbf{x})\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \quad (6.9)$$

(4) 商法则: 若 $g(\mathbf{x}) \neq 0$, 则

$$\frac{\partial f(\mathbf{x})/g(\mathbf{x})}{\partial \mathbf{x}} = \frac{1}{g^2(\mathbf{x})} \left[g(\mathbf{x})\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} - f(\mathbf{x})\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \right] \quad (6.10)$$

(5) 链式法则: 若 $\mathbf{y}(\mathbf{x})$ 是 \mathbf{x} 的向量值函数, 则

$$\frac{\partial f(\mathbf{y}(\mathbf{x}))}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}^T(\mathbf{x})}{\partial \mathbf{x}} \frac{\partial f(\mathbf{y})}{\partial \mathbf{y}} \quad (6.11)$$

式中, $\frac{\partial \mathbf{y}^T(\mathbf{x})}{\partial \mathbf{x}}$ 为 $n \times n$ 矩阵。

(6) 若 $n \times 1$ 向量 \mathbf{a} 与 \mathbf{x} 是无关的常数向量, 则

$$\frac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} = \mathbf{a}, \quad \frac{\partial \mathbf{x}^T \mathbf{a}}{\partial \mathbf{x}} = \mathbf{a}$$

(7) 若 $n \times 1$ 向量 \mathbf{a} 与 \mathbf{x} 是无关的常数向量, 则

$$\frac{\partial \mathbf{a}^T \mathbf{y}(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}^T(\mathbf{x})}{\partial \mathbf{x}} \mathbf{a}, \quad \frac{\partial \mathbf{y}^T(\mathbf{x}) \mathbf{a}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y}^T(\mathbf{x})}{\partial \mathbf{x}} \mathbf{a}$$

(8) 若 A 和 \mathbf{y} 均与向量 \mathbf{x} 无关, 则

$$\frac{\partial \mathbf{x}^T A \mathbf{y}}{\partial \mathbf{x}} = A \mathbf{y}, \quad \frac{\partial \mathbf{y}^T A \mathbf{x}}{\partial \mathbf{x}} = A^T \mathbf{y}$$

(9) 令 A 是一个与向量 \mathbf{x} 无关的矩阵, 则

$$\frac{\partial \mathbf{x}^T A}{\partial \mathbf{x}} = A, \quad \frac{\partial \mathbf{x}^T A \mathbf{x}}{\partial \mathbf{x}} = A \mathbf{x} + A^T \mathbf{x} = (A + A^T) \mathbf{x}$$

特别地, 若 A 为对称矩阵, 则有 $\frac{\partial \mathbf{x}^T A \mathbf{x}}{\partial \mathbf{x}} = 2A \mathbf{x}$ 。

(10) 若 A 与向量 \mathbf{x} 无关, 而 $\mathbf{y}(\mathbf{x})$ 是与向量 \mathbf{x} 的元素有关, 则

$$\frac{\partial[\mathbf{y}(\mathbf{x})]^T \mathbf{A} \mathbf{y}(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial[\mathbf{y}(\mathbf{x})]^T}{\partial \mathbf{x}} (\mathbf{A} + \mathbf{A}^T) \mathbf{y}(\mathbf{x})$$

(11) 若 \mathbf{A} 是一个与向量 \mathbf{x} 无关的矩阵, 而 $\mathbf{y}(\mathbf{x})$ 和 $\mathbf{z}(\mathbf{x})$ 是与向量 \mathbf{x} 的元素有关的列向量, 则

$$\frac{\partial[\mathbf{y}(\mathbf{x})]^T \mathbf{A} \mathbf{z}(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial[\mathbf{y}(\mathbf{x})]^T}{\partial \mathbf{x}} \mathbf{A} \mathbf{z}(\mathbf{x}) + \frac{\partial[\mathbf{z}(\mathbf{x})]^T}{\partial \mathbf{x}} \mathbf{A}^T \mathbf{y}(\mathbf{x})$$

(1),(2),(3),(4),(5) 容易从导数的运算法则推出.(6) 是一个容易获得的结论. 综合 (5),(6) 便可得出 (7).(8) 是 (6) 的一个特例.(9) 在前面已经证明过了. 综合 (5),(9) 便可得出 (10).(11) 与 (9) 的证明过程类似.

6.4.2 矩阵函数的梯度

定义 6.4.6. 若 $\mathbf{A} \in \mathbb{R}^{n \times m}$, $f(\mathbf{A}) : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$ 是一实值函数, 其中 $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}$,

则定义矩阵函数的梯度为

$$\frac{\partial}{\partial \mathbf{A}} f = \begin{pmatrix} \frac{\partial f}{\partial a_{11}} & \frac{\partial f}{\partial a_{12}} & \dots & \frac{\partial f}{\partial a_{1m}} \\ \frac{\partial f}{\partial a_{21}} & \frac{\partial f}{\partial a_{22}} & \dots & \frac{\partial f}{\partial a_{2m}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f}{\partial a_{n1}} & \frac{\partial f}{\partial a_{n2}} & \dots & \frac{\partial f}{\partial a_{nm}} \end{pmatrix}$$

例 6.4.9. 令 $f : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$, $f(\mathbf{A}) = \sum_{i,j} a_{ij}$, 其中 a_{ij} 为矩阵 \mathbf{A} 的第 ij 个元素, 求 $\frac{\partial f}{\partial \mathbf{A}}$ 。

解 我们对每一分量进行求导可得

$$\frac{\partial f}{\partial a_{ij}} = 1$$

故根据定义 6.4.6, 则有

$$\frac{\partial f}{\partial \mathbf{A}} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}$$

注意在向量函数梯度定义 6.4.5 中 \mathbf{x} 是一列向量。若将行向量和列向量均看做矩阵的特殊情况, 则我们只需给出矩阵函数梯度的定义 6.4.6, 由此可导出向量函数梯度定义 6.4.5。通过定义 6.4.6 我们可以自然地导出对 \mathbf{x}^T 求偏导的结果。

- 注意在向量函数梯度定义 6.4.5 中 \mathbf{x} 是一列向量。

- 若将行向量和列向量均看做矩阵的特殊情况，则我们只需给出矩阵函数梯度的定义6.4.6，由此可导出向量函数梯度定义6.4.5。
- 通过定义6.4.6我们可以自然地导出对 \mathbf{x}^T 求偏导的结果。

定理 6.4.1. 若 $\mathbf{x} \in \mathbb{R}^n, f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ 是一实值函数，则有

$$\frac{\partial}{\partial \mathbf{x}^T} f = \left(\frac{\partial}{\partial \mathbf{x}} f \right)^T$$

证明. 通过定义6.4.6，有

$$\frac{\partial}{\partial \mathbf{x}^T} f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}^T = \left(\frac{\partial}{\partial \mathbf{x}} f \right)^T$$

□

例 6.4.10. 在例6.4.4中我们考虑了一个非常简单的多元线性函数 $f(\mathbf{x}) = \mathbf{a}^T \mathbf{x} + \mathbf{b}$ ，我们知道

$$\frac{\partial f}{\partial \mathbf{x}} = \mathbf{a}$$

利用上述定理我们有

$$\frac{\partial f}{\partial \mathbf{x}^T} = \left(\frac{\partial f}{\partial \mathbf{x}} \right)^T = \mathbf{a}^T$$

注意我们在这个例子中实际上仅仅使用了定义。之后我们将使用矩阵性质来展示相同的结果，并且不需要使用 $\frac{\partial f}{\partial \mathbf{x}}$ 作为桥梁。

例 6.4.11. 对于一个可分的支持向量机，相应的优化问题为

$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{s.t. } y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 > 0 \end{aligned}$$

我们考虑其目标函数的梯度

$$\frac{\partial}{\partial \mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2$$

我们逐分量地求其偏导数有

$$\frac{\partial}{\partial w_i} \frac{1}{2} \|\mathbf{w}\|^2 = \frac{\partial}{\partial w_i} \frac{1}{2} \sum_{i=1}^n w_i^2 = w_i$$

所以

$$\frac{\partial}{\partial \mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 = \mathbf{w}$$

实值函数相对于矩阵变元的梯度具有以下性质。

- (1) 若 $f(\mathbf{A}) = c$ 为常数，其中 \mathbf{A} 为 $m \times n$ 矩阵，则梯度 $\frac{\partial c}{\partial \mathbf{A}} = \mathbf{O}_{m \times n}$ 。
- (2) 线性法则：若 $f(\mathbf{A})$ 和 $g(\mathbf{A})$ 分别是矩阵 \mathbf{A} 的实值函数， c_1 和 c_2 为实常数，则

$$\frac{\partial[c_1f(\mathbf{A}) + c_2g(\mathbf{A})]}{\partial \mathbf{A}} = c_1\frac{\partial f(\mathbf{A})}{\partial \mathbf{A}} + c_2\frac{\partial g(\mathbf{A})}{\partial \mathbf{A}}$$

(3) 乘积法则: 若 $f(\mathbf{A}), g(\mathbf{A})$ 和 $h(\mathbf{A})$ 分别是矩阵 \mathbf{A} 的实值函数, 则

$$\frac{\partial f(\mathbf{A})g(\mathbf{A})}{\partial \mathbf{A}} = g(\mathbf{A})\frac{\partial f(\mathbf{A})}{\partial \mathbf{A}} + f(\mathbf{A})\frac{\partial g(\mathbf{A})}{\partial \mathbf{A}}$$

$$\frac{\partial f(\mathbf{A})g(\mathbf{A})h(\mathbf{A})}{\partial \mathbf{A}} = g(\mathbf{A})h(\mathbf{A})\frac{\partial f(\mathbf{A})}{\partial \mathbf{A}} + f(\mathbf{A})h(\mathbf{A})\frac{\partial g(\mathbf{A})}{\partial \mathbf{A}} + f(\mathbf{A})g(\mathbf{A})\frac{\partial h(\mathbf{A})}{\partial \mathbf{A}}$$

(4) 商法则: 若 $g(\mathbf{A}) \neq 0$, 则

$$\frac{\partial f(\mathbf{A})/g(\mathbf{A})}{\partial \mathbf{A}} = \frac{1}{g^2(\mathbf{A})} \left[g(\mathbf{A})\frac{\partial f(\mathbf{A})}{\partial \mathbf{A}} - f(\mathbf{A})\frac{\partial g(\mathbf{A})}{\partial \mathbf{A}} \right]$$

(5) 链式法则: 令 \mathbf{A} 为 $m \times n$ 矩阵, 且 $y = f(\mathbf{A})$ 和 $g(y)$ 分别是以矩阵 \mathbf{A} 和标量 y 为变元的实值函数, 则

$$\frac{\partial g(f(\mathbf{A}))}{\partial \mathbf{A}} = \frac{dg(y)}{dy} \frac{\partial f(\mathbf{A})}{\partial \mathbf{A}}$$

(6) 若 $\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{x} \in \mathbb{R}^{m \times 1}, \mathbf{y} \in \mathbb{R}^{n \times 1}$, 则

$$\frac{\partial \mathbf{x}\mathbf{A}\mathbf{y}}{\partial \mathbf{A}} = \mathbf{x}\mathbf{y}^T$$

(7) 若 $\mathbf{A} \in \mathbb{R}^{n \times n}$, 非奇异, $\mathbf{x} \in \mathbb{R}^{n \times 1}, \mathbf{y} \in \mathbb{R}^{n \times 1}$ 则

$$\frac{\partial \mathbf{x}^T \mathbf{A}^{-1} \mathbf{y}}{\partial \mathbf{A}} = -\mathbf{A}^{-T} \mathbf{x} \mathbf{y}^T \mathbf{A}^{-T}, \mathbf{A}^{-T} = \mathbf{A}^{-1 T}$$

(8) 若 $\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^{n \times 1}$, 则

$$\frac{\partial \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{y}}{\partial \mathbf{A}} = \mathbf{A}(\mathbf{x}\mathbf{y}^T + \mathbf{y}\mathbf{x}^T)$$

(9) 若 $\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^{m \times 1}$, 则

$$\frac{\partial \mathbf{x}^T \mathbf{A} \mathbf{A}^T \mathbf{y}}{\partial \mathbf{A}} = (\mathbf{x}\mathbf{y}^T + \mathbf{y}\mathbf{x}^T)\mathbf{A}$$

(10) 指数函数的梯度

$$\frac{\partial \exp(\mathbf{x}^T \mathbf{A} \mathbf{y})}{\partial \mathbf{A}} = \mathbf{x}\mathbf{y}^T \exp(\mathbf{x}^T \mathbf{A} \mathbf{y})$$

(1),(2),(3),(4),(5) 通过导数的运算法则容易得出这些结论。(6),(7),(8),(9),(10) 均可以通过本节最后一部分内容迹微分法推导出来。其中 (6),(7) 两个结论均在本节最后一部分内容中给出了证明。

6.4.3 对矩阵微分

尽管大多数时候我们想要的是矩阵导数, 但是因为微分形式不变性, 将问题转化为求矩阵微分会更容易求解。

定义 6.4.7. 设 $A \in \mathbb{R}^{m \times n}$, 矩阵 A 的微分定义为

$$dA = \begin{pmatrix} da_{11} & da_{12} & \dots & da_{1n} \\ da_{21} & da_{22} & \dots & da_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ da_{m1} & da_{m2} & \dots & da_{mn} \end{pmatrix}$$

与上面类似, 我们也可以将矩阵微分的定义推广到向量上。

定义 6.4.8. 设 $x \in \mathbb{R}^n$, 向量 x 的微分定义为

$$dx = \begin{pmatrix} dx_1 \\ dx_2 \\ \vdots \\ dx_n \end{pmatrix}; dx^T = (dx_1, dx_2, \dots, dx_n)$$

性质 6.4.1. 矩阵微分有如下性质

- $d(cA) = cdA$ 其中 $A \in \mathbb{R}^{n \times m}$
- $d(A + B) = dA + dB$ 其中 $A, B \in \mathbb{R}^{n \times m}$
- $d(AB) = dAB + AdB$ 其中 $A \in \mathbb{R}^{n \times m}, B \in \mathbb{R}^{m \times k}$
- $dA^T = (dA)^T$ 其中 $A \in \mathbb{R}^{n \times m}$

证明. 这些性质都能通过矩阵微分的定义自然推出, 我们只在这里证明第 3 个性质。注意等式成立需要两边每一个对应元素都相等, 我们考虑两边的第 ij 个元素, 并记 A, B 的第 ij 个元素分别为 a_{ij}, b_{ij} 。

$$\begin{aligned} \text{左边}_{ij} &= d \left(\sum_k a_{ik} b_{kj} \right) \\ &= \sum_k (da_{ik} b_{kj} + a_{ik} db_{kj}) \end{aligned}$$

$$\begin{aligned} \text{右边}_{ij} &= (dAB)_{ij} + (AdB)_{ij} \\ &= \sum_k da_{ik} b_{kj} + \sum_k a_{ik} db_{kj} \\ &= \text{左边}_{ij} \end{aligned}$$

□

定理 6.4.2. 微分运算和迹运算可交换, 即设 $A \in \mathbb{R}^{n \times n}$, 则

$$d\text{Tr}(A) = \text{Tr}(dA)$$

证明.

$$\begin{aligned} \text{左边} &= d \left(\sum_i a_{ii} \right) = \sum_i da_{ii} \\ \text{右边} &= \text{Tr} \begin{bmatrix} da_{11} & da_{12} & \dots & da_{1n} \\ da_{21} & da_{22} & \dots & da_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ da_{n1} & da_{n2} & \dots & da_{nn} \end{bmatrix} = \sum_i da_{ii} = \text{左边} \end{aligned}$$

□

矩阵微分与偏导数的联系

要弄清矩阵微分与偏导数的联系，我们首先回顾一下迹函数的性质：

- 性质 6.4.2.**
1. $\text{Tr}(A + B) = \text{Tr}(A) + \text{Tr}(B)$ 其中 $A, B \in \mathbb{R}^{n \times n}$
 2. $\text{Tr}(cA) = c\text{Tr}(A)$ 其中 $A \in \mathbb{R}^{n \times n}, c \in \mathbb{R}$
 3. $\text{Tr}(AB) = \text{Tr}(BA)$ 其中 $A \in \mathbb{R}^{n \times m}, B \in \mathbb{R}^{m \times n}$
 4. $\text{Tr}(A_1 A_2 \dots A_n) = \text{Tr}(A_n A_1 \dots A_{n-1})$ 其中 $A_1 \in \mathbb{R}^{c_n, c_1}; A_i \in \mathbb{R}^{c_{i-1} \times c_i}, i = 2, 3, \dots, n$
 5. $\text{Tr}(A^T B) = \sum_i \sum_j A_{ij} B_{ij}$ 其中 $A, B \in \mathbb{R}^{n \times m}$
 6. $\text{Tr}(A) = \text{Tr}(A^T)$ 其中 $A \in \mathbb{R}^{n \times n}$

多元函数的微分和偏导的关系如下

$$df(x_1, x_2, \dots, x_n) = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 + \dots + \frac{\partial f}{\partial x_n} dx_n$$

这里 df 是一个标量，从分量的角度来看， df 就是将 $\frac{\partial f}{\partial x}$ 与 $d\mathbf{x}$ 相同位置的元素相乘后再求和。我们希望对于矩阵微分与偏导数能够得到一个类似的形式。此时我们可以利用迹函数第 5 条性质来给出下面这个定理：

定理 6.4.3. 对于实值函数 $f: \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$ 和 $A \in \mathbb{R}^{n \times m}$ 有

$$df = \text{Tr} \left[\left(\frac{\partial f}{\partial A} \right)^T dA \right]$$

证明.

$$\text{左边} = df = \sum_{ij} \frac{\partial f}{\partial x_{ij}} dx_{ij}$$

$$\text{右边} = \text{Tr} \left[\left(\frac{\partial f}{\partial A} \right)^T dA \right]$$

$$= \sum_{ij} \left(\frac{\partial f}{\partial A} \right)_{ij} (dA)_{ij}$$

$$= \sum_{ij} \frac{\partial f}{\partial x_{ij}} dx_{ij} = \text{左边}$$

初
等
传
授
讲
稿
草

□

注意对于向量也有类似的结果。这里不再叙述。

6.5 迹函数和行列式的微分

迹函数在处理矩阵微分的问题中具有很重要的地位。下面我们将给出一种利用迹函数和矩阵微分来求解实值函数的梯度的方法——迹微分法。我们知道对于一个标量 c 来说 $c = \text{Tr}(c)$, 这也就意味着对于一个实值函数 $f(\mathbf{A})$ 有 $f(\mathbf{A}) = \text{Tr}(f(\mathbf{A}))$ 。从而就有 $\text{d}f(\mathbf{A}) = \text{d}\text{Tr}(f(\mathbf{A})) = \text{Tr}(\text{d}f(\mathbf{A}))$ 。通过矩阵微分与迹运算的交换性、迹函数性质、矩阵微分的性质以及定理6.4.3我们可以总结出如下迹微分法:

1. $\text{d}f = \text{d}\text{Tr}(f) = \text{Tr}(\text{d}f)$
2. 使用迹函数的性质和矩阵微分的性质来得到如下形式

$$\text{d}f = \text{Tr}(\mathbf{A}^T \text{d}\mathbf{x})$$

3. 应用定理6.4.3得到结果

$$\frac{\partial f}{\partial \mathbf{x}} = \mathbf{A}$$

下面先举几个例子说明如何求迹的梯度。

例 6.5.1. 对于 $n \times n$ 矩阵 \mathbf{A} , 由于 $\text{tr}(\mathbf{A}) = \sum_{i=1}^n A_{ii}$, 故梯度 $\frac{\partial \text{tr}(\mathbf{A})}{\partial \mathbf{A}}$ 的 (i, j) 元素为

$$\left[\frac{\partial \text{tr}(\mathbf{A})}{\partial \mathbf{A}} \right] = \frac{\partial \sum_{k=1}^n A_{kk}}{\partial A_{ij}} = \begin{cases} 1, & i = j \\ 0, & j \neq 0 \end{cases}$$

即有 $\frac{\partial \text{tr}(\mathbf{A})}{\partial \mathbf{A}} = \mathbf{I}_n$ 。

例 6.5.2. 考查目标函数 $f(\mathbf{A}) = \text{Tr}(\mathbf{AB})$, 其中, \mathbf{A} 和 \mathbf{B} 分别为 $m \times n$ 和 $n \times m$ 实矩阵。首先, 矩阵乘积的元素为 $[\mathbf{AB}]_{ij} = \sum_{l=1}^n A_{il}B_{lj}$, 故矩阵乘积的迹 $\text{Tr}(\mathbf{AB}) = \sum_{p=1}^m \sum_{l=1}^n A_{pl}B_{lp}$ 。于是, 梯度 $\frac{\partial \text{Tr}(\mathbf{AB})}{\partial \mathbf{A}}$ 是 $m \times n$ 矩阵, 其元素为

$$\left[\frac{\partial \text{Tr}(\mathbf{AB})}{\partial \mathbf{A}} \right]_{ij} = \frac{\partial}{\partial A_{ij}} \left(\sum_{p=1}^m \sum_{l=1}^n A_{pl}B_{lp} \right) = \mathbf{B}_{ji}$$

即有

$$\frac{\partial \text{Tr}(\mathbf{AB})}{\partial \mathbf{A}} = \Delta_{\mathbf{A}} \text{Tr}(\mathbf{AB}) = \mathbf{B}^T$$

又由于 $\text{Tr}(\mathbf{BA}) = \text{Tr}(\mathbf{AB})$, 故

$$\frac{\partial \text{tr}(\mathbf{AB})}{\partial \mathbf{A}} = \frac{\partial \text{tr}(\mathbf{BA})}{\partial \mathbf{A}} = \mathbf{B}^T$$

例 6.5.3. 由于 $\text{Tr}(\mathbf{x}\mathbf{y}^T) = \text{Tr}(\mathbf{y}\mathbf{x}^T) = \mathbf{x}^T\mathbf{y}$, 易知

$$\frac{\partial \text{Tr}(\mathbf{x}\mathbf{y}^T)}{\partial \mathbf{x}} = \frac{\partial \text{Tr}(\mathbf{y}\mathbf{x}^T)}{\partial \mathbf{x}} = \mathbf{y}$$

例 6.5.4. 给定函数 $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$, 其中 \mathbf{A} 是一方阵, \mathbf{x} 是一列向量, 我们计算

$$\begin{aligned} df &= d \text{Tr} (\mathbf{x}^T \mathbf{A} \mathbf{x}) \\ &= \text{Tr} (d (\mathbf{x}^T \mathbf{A} \mathbf{x})) \\ &= \text{Tr} (d (\mathbf{x}^T) \mathbf{A} \mathbf{x} + \mathbf{x}^T d(\mathbf{A} \mathbf{x})) \\ &= \text{Tr} (d (\mathbf{x}^T) \mathbf{A} \mathbf{x} + \mathbf{x}^T d\mathbf{A} \mathbf{x} + \mathbf{x}^T \mathbf{A} d\mathbf{x}) \\ &= \text{Tr} (d\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{x}^T \mathbf{A} d\mathbf{x}) \\ &= \text{Tr} (d\mathbf{x}^T \mathbf{A} \mathbf{x}) + \text{Tr} (\mathbf{x}^T \mathbf{A} d\mathbf{x}) \\ &= \text{Tr} (\mathbf{x}^T \mathbf{A}^T d\mathbf{x}) + \text{Tr} (\mathbf{x}^T \mathbf{A} d\mathbf{x}) \\ &= \text{Tr} (\mathbf{x}^T \mathbf{A}^T d\mathbf{x} + \mathbf{x}^T \mathbf{A} d\mathbf{x}) \\ &= \text{Tr} ((\mathbf{x}^T \mathbf{A}^T + \mathbf{x}^T \mathbf{A}) d\mathbf{x}) \end{aligned}$$

我们可以得到

$$\frac{\partial f}{\partial \mathbf{x}} = (\mathbf{x}^T \mathbf{A}^T + \mathbf{x}^T \mathbf{A})^T = \mathbf{A} \mathbf{x} + \mathbf{A}^T \mathbf{x}$$

如果 \mathbf{A} 是对称矩阵, 我们还可以将其简化为

$$\frac{\partial f}{\partial \mathbf{x}} = 2\mathbf{A} \mathbf{x}$$

令 $\mathbf{A} = \mathbf{I}$ 我们则有

$$\frac{\partial (\mathbf{x}^T \mathbf{x})}{\partial \mathbf{x}} = 2\mathbf{x}$$

例 6.5.5. 根据上面的推导可以知道, 在谱聚类中我们要求解的优化问题

$$\begin{aligned} &\min_{\mathbf{x}} \mathbf{x}^T \mathbf{L} \mathbf{x} \\ &s.t. \mathbf{x}^T \mathbf{1} = 0 \end{aligned}$$

中目标函数的梯度为

$$\nabla_{\mathbf{x}} \mathbf{x}^T \mathbf{L} \mathbf{x} = 2\mathbf{L} \mathbf{x}$$

我们再看一个关于矩阵函数的例子。

例 6.5.6. 在 PCA 中, 我们需要求解优化问题

$$\begin{aligned} &\min_{\mathbf{W}} -\text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}) \\ &s.t. \mathbf{W}^T \mathbf{W} = \mathbf{I} \end{aligned}$$

我们现在考虑求梯度 $\nabla_{\mathbf{W}} -\text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W})$ 。

解. 利用迹微分法有

$$\begin{aligned} d(-\text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W})) &= -\text{Tr}(d(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W})) \\ &= -2 \text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T d\mathbf{W}) \end{aligned}$$

所以

$$\nabla_{\mathbf{W}} - \text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}) = -2 \mathbf{X} \mathbf{X}^T \mathbf{W}$$

关于 F 范数的函数的梯度

我们可以使用迹微分法来处理含 F 范数的函数。

例 6.5.7. 设 $\mathbf{A} \in \mathbb{R}^{n \times m}$, 求 $\frac{\partial ||\mathbf{A}||_F^2}{\partial \mathbf{A}}$, 其中 $||\mathbf{A}||_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m a_{ij}^2}$

解.

$$\begin{aligned} d||\mathbf{A}||_F^2 &= d\text{Tr}(\mathbf{A}^T \mathbf{A}) \\ &= \text{Tr}(d(\mathbf{A}^T \mathbf{A})) \\ &= \text{Tr}((d\mathbf{A})^T \mathbf{A}) + \text{Tr}(\mathbf{A}^T d\mathbf{A}) \\ &= \text{Tr}(2\mathbf{A}^T d\mathbf{A}) \end{aligned}$$

$$\frac{\partial ||\mathbf{A}||_F^2}{\partial \mathbf{A}} = 2\mathbf{A}$$

通过以上几个例子，并综合 Langville A N,Horn R A(1991),Kumar R(1985) 的有关结果，可得到关于迹的梯度矩阵的一些常见公式。

(1) 单个矩阵的迹的梯度矩阵

(a) \mathbf{W} 是 $m \times m$ 矩阵时, 有

$$\frac{\partial \text{Tr}(\mathbf{W})}{\partial \mathbf{W}} = \mathbf{I}_m \quad (6.12)$$

(b) $m \times m$ 矩阵 \mathbf{W} 可逆时, 有

$$\frac{\partial \text{Tr}(\mathbf{W}^{-1})}{\partial \mathbf{W}} = -(\mathbf{W}^{-1})^T \quad (6.13)$$

(c) 对于两个向量的外积, 有

$$\frac{\partial \text{Tr}(\mathbf{x}\mathbf{y}^T)}{\partial \mathbf{x}} = \frac{\partial \text{Tr}(\mathbf{y}\mathbf{x}^T)}{\partial \mathbf{x}} = \mathbf{y} \quad (6.14)$$

(2) 两个矩阵乘积的迹的梯度

(a) 若 $\mathbf{W} \in \mathbb{R}^{m \times n}$, $\mathbf{A} \in \mathbb{R}^{n \times m}$, 则

$$\frac{\partial \text{Tr}(\mathbf{WA})}{\partial \mathbf{W}} = \frac{\partial \text{Tr}(\mathbf{AW})}{\partial \mathbf{W}} = \mathbf{A}^T \quad (6.15)$$

特别地, 有

$$\frac{\partial \text{Tr}(\mathbf{W}\mathbf{A})}{\partial \mathbf{W}} = \frac{\partial \text{Tr}(\mathbf{A}\mathbf{W})}{\partial \mathbf{W}} = \mathbf{A} + \mathbf{A}^T - \text{diag}(\mathbf{A}), \quad \mathbf{W} \text{为对称矩阵} \quad (6.16)$$

(b) 若 $\mathbf{W} \in \mathbb{R}^{m \times n}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, 有

$$\frac{\partial \text{Tr}(\mathbf{W}^T \mathbf{A})}{\partial \mathbf{W}} = \frac{\partial \text{Tr}(\mathbf{A}\mathbf{W}^T)}{\partial \mathbf{W}} = \mathbf{A} \quad (6.17)$$

(c) 对于 $\mathbf{W} \in \mathbb{R}^{m \times m}$, 有

$$\frac{\partial \text{Tr}(\mathbf{W}\mathbf{W}^T)}{\partial \mathbf{W}} = \frac{\partial \text{Tr}(\mathbf{W}^T \mathbf{W})}{\partial \mathbf{W}} = 2\mathbf{W} \quad (6.18)$$

(d) 令 $\mathbf{W} \in \mathbb{R}^{m \times m}$, 则

$$\frac{\partial \text{Tr}(\mathbf{W}^2)}{\partial \mathbf{W}} = \frac{\partial \text{Tr}(\mathbf{W}\mathbf{W})}{\partial \mathbf{W}} = 2\mathbf{W}^T \quad (6.19)$$

(e) 若 $\mathbf{W}, \mathbf{A} \in \mathbb{R}^{m \times m}$, 并且 \mathbf{W} 非奇异, 则

$$\frac{\partial \text{Tr}(\mathbf{A}\mathbf{W}^{-1})}{\partial \mathbf{W}} = -(\mathbf{W}^{-1}\mathbf{A}\mathbf{W}^{-1})^T \quad (6.20)$$

(3) 三个矩阵乘积的迹的梯度

(a) 若 $\mathbf{W} \in \mathbb{R}^{m \times n}$, $\mathbf{A} \in \mathbb{R}^{m \times m}$, 则

$$\frac{\partial \text{Tr}(\mathbf{W}^T \mathbf{A}\mathbf{W})}{\partial \mathbf{W}} = (\mathbf{A} + \mathbf{A}^T)\mathbf{W} \quad (6.21)$$

特别地, 当 \mathbf{A} 为对称矩阵时, 有 $\frac{\partial \text{Tr}(\mathbf{W}^T \mathbf{A}\mathbf{W})}{\partial \mathbf{W}} = 2\mathbf{A}\mathbf{W}$ 。

(b) 若 $\mathbf{W} \in \mathbb{R}^{m \times n}$, $\mathbf{A} \in \mathbb{R}^{n \times n}$, 则

$$\frac{\partial \text{Tr}(\mathbf{W}\mathbf{A}\mathbf{W}^T)}{\partial \mathbf{W}} = \mathbf{W}(\mathbf{A} + \mathbf{A}^T) \quad (6.22)$$

特别地, 当 \mathbf{A} 为对称矩阵时, 有 $\frac{\partial \text{Tr}(\mathbf{W}\mathbf{A}\mathbf{W}^T)}{\partial \mathbf{W}} = 2\mathbf{W}\mathbf{A}$ 。

(c) 若 $\mathbf{W}, \mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times m}$, 并且 \mathbf{W} 非奇异, 则

$$\frac{\partial \text{Tr}(\mathbf{A}\mathbf{W}^{-1}\mathbf{B})}{\partial \mathbf{W}} = -(\mathbf{W}^{-1}\mathbf{B}\mathbf{A}\mathbf{W}^{-1})^T \quad (6.23)$$

(4) 四个矩阵乘积的迹的梯度

(a) 若 $\mathbf{W} \in \mathbb{R}^{m \times n}$ 和 $\mathbf{A} \in \mathbb{R}^{p \times m}$ 时:

$$\frac{\partial \text{Tr}(\mathbf{A}\mathbf{W}\mathbf{W}^T \mathbf{A}^T)}{\partial \mathbf{W}} = 2\mathbf{A}^T \mathbf{A}\mathbf{W} \quad (6.24)$$

(b) 若 $\mathbf{W} \in \mathbb{R}^{m \times n}$ 和 $\mathbf{A} \in \mathbb{R}^{p \times n}$ 时:

$$\frac{\partial \text{Tr}(\mathbf{A}\mathbf{W}^T \mathbf{W}\mathbf{A}^T)}{\partial \mathbf{W}} = 2\mathbf{W}\mathbf{A}^T \mathbf{A} \quad (6.25)$$

(c) 若 $\mathbf{W} \in \mathbb{R}^{m \times n}$, $\mathbf{A} \in \mathbb{R}^{p \times m}$, $\mathbf{B} \in \mathbb{R}^{m \times p}$ 时, 有

$$\frac{\partial \text{Tr}(\mathbf{A}\mathbf{W}\mathbf{W}^T \mathbf{B})}{\partial \mathbf{W}} = (\mathbf{B}\mathbf{A} + \mathbf{A}^T \mathbf{B}^T)\mathbf{W} \quad (6.26)$$

(d) 若 $\mathbf{W} \in \mathbb{R}^{m \times n}$, $\mathbf{A} \in \mathbb{R}^{p \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times p}$ 时, 有

$$\frac{\partial \text{Tr}(\mathbf{A}\mathbf{W}^T \mathbf{W}\mathbf{B})}{\partial \mathbf{W}} = \mathbf{W}(\mathbf{B}\mathbf{A} + \mathbf{A}^T \mathbf{B}^T) \quad (6.27)$$

(e) 若 $\mathbf{W} \in \mathbb{R}^{m \times n}$, $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{m \times m}$ 时, 有

$$\frac{\partial \text{Tr}(\mathbf{W}\mathbf{A}^T \mathbf{W}^T \mathbf{B})}{\partial \mathbf{W}} = \mathbf{B}\mathbf{W}\mathbf{A} + \mathbf{B}^T \mathbf{W}\mathbf{A}^T \quad (6.28)$$

(f) 若 $\mathbf{W} \in \mathbb{R}^{m \times n}$, $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$ 时, 有

$$\frac{\partial \text{Tr}(\mathbf{W}\mathbf{A}\mathbf{W}\mathbf{B})}{\partial \mathbf{W}} = \mathbf{B}^T \mathbf{W}^T \mathbf{A}^T + \mathbf{A}^T \mathbf{W}^T \mathbf{B}^T \quad (6.29)$$

6.5.1 关于逆矩阵的函数的微分

$$0 = d\mathbf{I} = d(\mathbf{XX}^{-1}) = d\mathbf{XX}^{-1} + \mathbf{X}d(\mathbf{X}^{-1})$$

$$d(\mathbf{X}^{-1}) = -\mathbf{X}^{-1}d\mathbf{XX}^{-1}$$

这样我们就得到了关于逆矩阵微分的一个结论。

例 6.5.8. 若 $\mathbf{A} \in \mathbb{R}^{n \times n}$, 非奇异, $\mathbf{x} \in \mathbb{R}^{n \times 1}, \mathbf{y} \in \mathbb{R}^{n \times 1}$ 求

$$\frac{\partial \mathbf{x}^T \mathbf{A}^{-1} \mathbf{y}}{\partial \mathbf{A}}$$

解.

$$\begin{aligned} d(\mathbf{x}^T \mathbf{A}^{-1} \mathbf{y}) &= \text{Tr}(d(\mathbf{x}^T \mathbf{A}^{-1} \mathbf{y})) \\ &= \text{Tr}(\mathbf{x}^T d\mathbf{A}^{-1} \mathbf{y}) \\ &= \text{Tr}(-\mathbf{x}^T \mathbf{A}^{-1} d\mathbf{A} \mathbf{A}^{-1} \mathbf{y}) \\ &= \text{Tr}(-\mathbf{A}^{-1} \mathbf{y} \mathbf{x}^T \mathbf{A}^{-1} d\mathbf{A}) \end{aligned}$$

所以
传
外
解。
故
高
请
不
要
忘
记
以
上
的
方
法
对
于
求
解
问
题
非
常
有
用
！

$$\frac{\partial \mathbf{x}^T \mathbf{A}^{-1} \mathbf{y}}{\partial \mathbf{A}} = -\mathbf{A}^{-T} \mathbf{y} \mathbf{x}^T \mathbf{A}^{-T}$$

例 6.5.9. 设函数 $f(\mathbf{X}) = \|\mathbf{AX}^{-1}\|_F^2$, 其中 $\mathbf{A} \in \mathbb{R}^{n \times m}, \mathbf{X} \in \mathbb{R}^{m \times m}$ 且 \mathbf{X} 可逆, 求 $\frac{\partial f}{\partial \mathbf{X}}$

解.

$$\begin{aligned} f(\mathbf{X}) &= \text{Tr}(\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1}) \\ df(\mathbf{X}) &= \text{Tr}[d(\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1})] \\ &= \text{Tr}(d\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1} + \mathbf{X}^{-T} \mathbf{A}^T \mathbf{AdX}^{-1}) \\ &= \text{Tr}(2\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AdX}^{-1}) \\ &= \text{Tr}(-2\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1} d\mathbf{X}^{-1}) \\ &= \text{Tr}(-2\mathbf{X}^{-1} \mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1} d\mathbf{X}) \end{aligned}$$

$$\frac{\partial f}{\partial \mathbf{X}} = -2\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1} \mathbf{X}^{-T}$$

6.5.2 关于行列式函数的梯度

行列式也是关于矩阵的一个实值函数, 有时我们会面临求 $\frac{\partial |\mathbf{A}|}{\partial \mathbf{A}}$ 。我们首先回顾一下行列式相关的一些概念, 假设矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$, 则:

- 余子式 M_{ij} 是矩阵 \mathbf{A} 划去第 i 行 j 列元素组成的矩阵的行列式

- 第 ij 个元素的代数余子式定义为 $A_{ij} = (-1)^{i+j} M_{ij}$
- 如果我们将行列式按第 i 行展开，则有 $|A| = \sum_j a_{ij} A_{ij}$
- A 的伴随矩阵被定义为 $A_{ij}^* = A_{ji}$
- 对于非奇异矩阵 A 有 $A^{-1} = \frac{A^*}{|A|}$

定理 6.5.1. 设矩阵 $A \in \mathbb{R}^{n \times n}$ 则有

$$\frac{\partial |A|}{\partial A} = (A^*)^T$$

证明. 为了计算 $\frac{\partial |A|}{\partial A}$ ，我们利用定义6.4.6逐元素进行求导。将行列式按第 i 行展开，则有

$$\frac{\partial |A|}{\partial a_{ij}} = \frac{\partial \left(\sum_j a_{ij} A_{ij} \right)}{\partial a_{ij}} = A_{ij}$$

使用定义6.4.6来组织元素就有

$$\frac{\partial |A|}{\partial A} = (A^*)^T$$

□

如果矩阵 A 非奇异，则可以进一步推出

$$\frac{\partial |A|}{\partial A} = (|A| A^{-1})^T = |A| (A^{-1})^T$$

通过上述偏导的结果和定理6.4.3，我们还能够给出对应的微分关系

定理 6.5.2. 设矩阵 $A \in \mathbb{R}^{n \times n}$ 则有

$$d|A| = \text{Tr}(A^* dA)$$

当 A 可逆时有

$$d|A| = \text{Tr}(|A| A^{-1} dA)$$

证明.

$$\begin{aligned} d|A| &= \text{Tr} \left(\left(\frac{\partial |A|}{\partial A} \right)^T dA \right) \\ &= \text{Tr} \left(((A^*)^T)^T dA \right) \\ &= \text{Tr}(A^* dA) \end{aligned}$$

当 A 可逆时有

$$d|A| = \text{Tr}(A^* dA) = \text{Tr}(|A| A^{-1} dA)$$

□

例 6.5.10. 设矩阵 $A \in \mathbb{R}^{n \times n}$ 是一可逆矩阵。求

$$\frac{\partial |A^{-1}|}{\partial A}$$

解. 应用定理6.5.2有

$$\begin{aligned} d|\mathbf{A}^{-1}| &= \text{Tr}(|\mathbf{A}^{-1}| \mathbf{A} d\mathbf{A}^{-1}) \\ &= \text{Tr}(-|\mathbf{A}^{-1}| \mathbf{A} \mathbf{A}^{-1} d\mathbf{A} \mathbf{A}^{-1}) \\ &= \text{Tr}(-|\mathbf{A}^{-1}| \mathbf{A}^{-1} d\mathbf{A}) \end{aligned}$$

故

$$\frac{\partial |\mathbf{A}^{-1}|}{\partial \mathbf{A}} = -|\mathbf{A}^{-1}| \mathbf{A}^{-T} = -|\mathbf{A}|^{-1} \mathbf{A}^{-T}$$

矩阵的行列式的梯度（矩阵）具有以下性质。

性质 6.5.1. 单个非奇异矩阵的行列式的梯度

$$\begin{aligned} \frac{\partial |\mathbf{W}|}{\partial \mathbf{W}} &= |\mathbf{W}| (\mathbf{W}^{-1})^T = (\mathbf{W}^\#)^T \\ \frac{\partial |\mathbf{W}^{-1}|}{\partial \mathbf{W}} &= -|\mathbf{W}|^{-1} (\mathbf{W}^{-1})^T \end{aligned} \tag{6.30}$$

式中， $\mathbf{W}^\#$ 是矩阵 \mathbf{A} 的伴随矩阵。

性质 6.5.2. 行列式对数的梯度

$$\begin{aligned} \frac{\partial}{\partial \mathbf{W}} \log |\mathbf{W}| &= \frac{1}{|\mathbf{W}|} \frac{\partial |\mathbf{W}|}{\partial \mathbf{W}}, \quad \mathbf{W} \text{ 非奇异} \\ &= (\mathbf{W}^{-1})^T, \quad \mathbf{W} \text{ 的元素相互独立} \\ &= 2\mathbf{W}^{-1} - \text{diag}(\mathbf{W}^{-1}), \quad \text{若 } \mathbf{W} \text{ 为对称矩阵} \end{aligned} \tag{6.31}$$

性质 6.5.3. 两个矩阵乘积的行列式的梯度

$$\begin{aligned} \frac{\partial |\mathbf{WW}^T|}{\partial \mathbf{W}} &= 2|\mathbf{WW}^T| (\mathbf{WW}^T)^{-1} \mathbf{W}, \quad \text{rank}(\mathbf{W}_{m \times n}) = m \\ \frac{\partial |\mathbf{W}^T \mathbf{W}|}{\partial \mathbf{W}} &= 2|\mathbf{W}^T \mathbf{W}| \mathbf{W} (\mathbf{W}^T \mathbf{W})^{-1}, \quad \text{rank}(\mathbf{W}_{m \times n}) = n \\ \frac{\partial |\mathbf{W}^2|}{\partial \mathbf{W}} &= 2|\mathbf{W}|^2 (\mathbf{W}^{-1})^T, \quad \text{rank}(\mathbf{W}_{m \times m}) = m \end{aligned} \tag{6.32}$$

性质 6.5.4. 三个矩阵乘积的行列式的梯度

$$\begin{aligned} \frac{\partial |\mathbf{AWB}|}{\partial \mathbf{W}} &= |\mathbf{AWB}| \mathbf{A}^T (\mathbf{B}^T \mathbf{W}^T \mathbf{A}^T)^{-1} \mathbf{B}^T \\ \frac{\partial |\mathbf{W}^T \mathbf{AW}|}{\partial \mathbf{W}} &= 2\mathbf{AW} (\mathbf{W}^T \mathbf{AW})^{-1}, \quad |\mathbf{W}^T \mathbf{AW}| > 0 \\ \frac{\partial |\mathbf{WAW}^T|}{\partial \mathbf{W}} &= [(\mathbf{WAW}^T)^{-1}]^T \mathbf{W} (\mathbf{A}^T + \mathbf{A}) \\ &= 2(\mathbf{WAW}^T)^{-1} \mathbf{WA}, \quad \mathbf{A} \text{ 为对称矩阵} \end{aligned}$$

传
外
清
稿
草

因为

$$\frac{\partial |\mathbf{W}|}{\partial \mathbf{W}} = |\mathbf{W}|(\mathbf{W}^{-1})^T, \quad \mathbf{W} \text{的元素相互独立} \quad (6.33)$$

所以

$$\begin{aligned} d|\mathbf{W}| &= \text{Tr}(|\mathbf{W}| \mathbf{W}^{-1} d\mathbf{W}) \\ &= |\mathbf{W}| \text{Tr}(\mathbf{W}^{-1} d\mathbf{W}) \end{aligned}$$

6.6 向量值函数和矩阵值函数的梯度

6.6.1 向量值函数的梯度

我们上面已经讨论函数实值函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 的偏导和梯度。接下来，我们将给出向量值函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}^m, (n, m \geq 1)$ 的梯度的概念。对于一个函数 $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 和一个向量 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ ，那么对应的函数值为

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^m$$

这样写能够更好地展示一个向量值函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ，它就相当于一个函数的向量 $(f_1, f_2, \dots, f_m)^T, f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ 。

因此，应用前面已经讨论过了的关于其中任一个 f_i 的微分法则，我们可得向量值函数 \mathbf{f} 关于 x_i 的偏导数：

$$\frac{\partial \mathbf{f}}{\partial x_i} = \begin{pmatrix} \frac{\partial f_1}{\partial x_i} \\ \vdots \\ \frac{\partial f_m}{\partial x_i} \end{pmatrix} = \begin{pmatrix} \lim_{h \rightarrow 0} \frac{f_1(x_1, \dots, x_{i-1}, x_i + h, x_{i+1}, \dots, x_m) - f_1(\mathbf{x})}{h} \\ \vdots \\ \lim_{h \rightarrow 0} \frac{f_m(x_1, \dots, x_{i-1}, x_i + h, x_{i+1}, \dots, x_m) - f_m(\mathbf{x})}{h} \end{pmatrix}$$

在上式中，每一个偏导都是一个列向量。因此，我们按照如下组织得到一个向量值函数的偏导：

$$\frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}^T} = \left[\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_n} \right] = \begin{pmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

定义 6.6.1. 向量值函数 $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 的所有一阶导数组成的矩阵称为 **Jacobian 矩阵**，它是一个 $m \times n$ 的矩阵，具体定义如下：

$$\mathbf{J} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}^T} = \begin{pmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

并且我们定义

$$\frac{\partial \mathbf{f}(\mathbf{x})^T}{\partial \mathbf{x}} = \mathbf{J}^T = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}^T} \right)^T$$

注意，这里我们没有去定义 $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ 以及 $\frac{\partial f(\mathbf{x})^T}{\partial \mathbf{x}^T}$ ，所以在后面的讨论中不会出现这两种情况。在计算中也需要注意所计算的形式是否已经被定义。

6.6.2 矩阵值函数的梯度

求矩阵关于向量或其它矩阵的梯度，通常会导致一个多元张量。例如，我们计算一个 $m \times n$ 矩阵关于 $p \times q$ 矩阵的梯度，相应的 Jacobian 是 $(p \times q) \times (m \times n)$ ，这是一个四维的张量。

定义 6.6.2. 函数 $vec : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{nm}$ 将一个矩阵按列重排成一个列向量。设 $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m) \in \mathbb{R}^{n \times m}$ 则

$$vec(\mathbf{A}) = \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_m \end{pmatrix}$$

有了这样一类函数之后，我们就可以定义矩阵关于矩阵梯度的 Jacobian 矩阵。

定义 6.6.3. 设矩阵函数 $\mathbf{F}(\mathbf{X}) : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{q \times p}$ 则其 Jacobian 矩阵定义为

$$\mathbf{J} = \frac{\partial vec(\mathbf{F}(\mathbf{X}))}{\partial vec(\mathbf{X})^T} = \begin{pmatrix} \frac{\partial f_{11}}{\partial x_{11}} & \frac{\partial f_{11}}{\partial x_{12}} & \cdots & \frac{\partial f_{11}}{\partial x_{nm}} \\ \frac{\partial f_{12}}{\partial x_{11}} & \frac{\partial f_{12}}{\partial x_{12}} & \cdots & \frac{\partial f_{12}}{\partial x_{nm}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_{pq}}{\partial x_{11}} & \frac{\partial f_{pq}}{\partial x_{12}} & \cdots & \frac{\partial f_{pq}}{\partial x_{nm}} \end{pmatrix}$$

定义 6.6.4. 设矩阵 \mathbf{J} 是一 Jacobian 矩阵，则其行列式 $J = |\mathbf{J}|$ 称为 Jacobian 行列式。

6.6.3 向量值函数微分

定理 6.6.1. 设函数 $f(\mathbf{x}) : \mathbb{R}^m \rightarrow \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^m$ 则有

$$df = \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}} \right)^T d\mathbf{x} = \mathbf{J}\mathbf{x}$$

证明。显然， df 有 n 个分量，所以我们从分量的角度来证明。考虑第 j 个分量。

$$\text{左边}_j = df_j = \sum_{i=1}^m \frac{\partial f_j}{\partial x_i} dx_i$$

$$\begin{aligned} \text{右边}_j &= \left(\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T d\mathbf{x} \right)_j \\ &= \sum_{i=1}^m \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)_{ij} dx_i \\ &= \sum_{i=1}^m \left(\frac{\partial f_j}{\partial x_i} \right) dx_i = \text{左边}_j \end{aligned}$$

勿外传
草稿请

□

注意这个式子在形式上与之前我们推得的定理6.4.3是很像的。

利用定理6.6.1，仿照求解实值函数梯度的步骤，我们可以简化求解向量对向量的导数。

例 6.6.1. 考虑向量变换 $\mathbf{x} = \sigma \Lambda^{-0.5} \mathbf{W}^T \boldsymbol{\eta}$, \mathbf{x} 和 $\boldsymbol{\eta}$ 的维数是 d , 其中 σ 是一个实变量, Λ 是一个满秩对角矩阵, \mathbf{W} 是正交矩阵(即 $\mathbf{W}\mathbf{W}^T = \mathbf{W}^T\mathbf{W} = \mathbf{I}$), 计算 Jacobian 行列式的绝对值。

$$d\mathbf{x} = d(\sigma \Lambda^{-0.5} \mathbf{W}^T \boldsymbol{\eta}) = \sigma \Lambda^{-0.5} \mathbf{W}^T d\boldsymbol{\eta}$$

应用定理6.6.1我们有

$$\mathbf{J} = \left(\frac{\partial \mathbf{x}^T}{\partial \boldsymbol{\eta}} \right)^T = \sigma \Lambda^{-0.5} \mathbf{W}^T$$

接着我们利用行列式的性质来计算 Jacobian 行列式 $J = |\mathbf{J}| = \det(\mathbf{J})$ 的绝对值。

$$\begin{aligned} |J| &= |\det(\mathbf{J})| \\ &= \sqrt{|\det(\mathbf{J})| |\det(\mathbf{J})|} \\ &= \sqrt{|\det(\mathbf{J})| |\det(\mathbf{J}^T)|} \\ &= \sqrt{|\det(\mathbf{J}^T \mathbf{J})|} \\ &= \sqrt{|\det(\mathbf{W} \Lambda^{-0.5} \sigma \sigma \Lambda^{-0.5} \mathbf{W}^T)|} \\ &= \sqrt{|\det(\sigma^2 \mathbf{W} \Lambda^{-1} \mathbf{W}^T)|} \end{aligned}$$

我们令 $\Sigma = \mathbf{W} \Lambda \mathbf{W}^T$ 。我们就能得到一个优美的结果

$$|J| = |\sigma|^d |\Sigma|^{-1/2}$$

这个结论我们可以应用到多元正态分布的推广中。

定理 6.6.2. 如果 \mathbf{f} 和 \mathbf{x} 维数相同, 则

$$\left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}} \right)^{-1} = \frac{\partial \mathbf{x}^T}{\partial \mathbf{f}}$$

证明. 利用定理6.6.1

$$d\mathbf{f} = \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}} \right)^T d\mathbf{x} \implies \left(\left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}} \right)^T \right)^{-1} d\mathbf{f} = d\mathbf{x} \implies d\mathbf{x} = \left(\left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}} \right)^{-1} \right)^T d\mathbf{f}$$

所以, 我们就有

$$\frac{\partial \mathbf{x}^T}{\partial \mathbf{f}} = \left(\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}} \right)^{-1}$$

□

这个结果和标量导数是一致的。这个结论对于变量替换很有用。

6.7 链式法则

回顾对于一元复合函数，设 $y = f(x)$, $z = g(y)$, 则我们知道

$$\frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx}$$

而对于多元复合函数，设 $z = f(y_1, y_2, \dots, y_n)$, $y_i = g_i(x_1, x_2, \dots, x_m)$, $i = 1, 2, \dots, n$, 则有

$$\frac{\partial z}{\partial x_j} = \sum_{i=1}^n \frac{\partial z}{\partial y_i} \frac{\partial y_i}{\partial x_j} = \sum_{i=1}^n \frac{\partial y_i}{\partial x_j} \frac{\partial z}{\partial y_i}$$

即

$$\frac{\partial z}{\partial x_j} = \left(\frac{\partial z}{\partial y_1}, \frac{\partial z}{\partial y_2}, \dots, \frac{\partial z}{\partial y_n} \right) \begin{pmatrix} \frac{\partial y_1}{\partial x_j} \\ \frac{\partial y_2}{\partial x_j} \\ \vdots \\ \frac{\partial y_n}{\partial x_j} \end{pmatrix} = \left(\frac{\partial y_1}{\partial x_j}, \frac{\partial y_2}{\partial x_j}, \dots, \frac{\partial y_n}{\partial x_j} \right) \begin{pmatrix} \frac{\partial z}{\partial y_1} \\ \frac{\partial z}{\partial y_2} \\ \vdots \\ \frac{\partial z}{\partial y_n} \end{pmatrix}$$

例 6.7.1. 考虑函数 $z = f(y_1, y_2) = e^{y_1 y_2^2}$, $y_1 = g_1(x) = x \cos x$, $y_2 = g_2(x) = x \sin x$. 那么

$$\begin{aligned} \frac{\partial z}{\partial x} &= \left(\frac{\partial y_1}{\partial x}, \frac{\partial y_2}{\partial x} \right) \begin{pmatrix} \frac{\partial z}{\partial y_1} \\ \frac{\partial z}{\partial y_2} \end{pmatrix} \\ &= (\cos x - x \sin x, \sin x + x \cos x) \begin{pmatrix} y_2^2 e^{y_1 y_2^2} \\ 2y_1 y_2 e^{y_1 y_2^2} \end{pmatrix} \\ &= (y_2^2(\cos x - x \sin x) + 2y_1 y_2(\sin x + x \cos x))e^{y_1 y_2^2} \end{aligned}$$

当我们把 $\mathbf{y} = \mathbf{g}(x)$ 看做一个向量值函数时，我们就可以将上述例子看做是求复合函数 $z = f(\mathbf{g}(x))$ 关于 x 的导数，并且可以得到公式

$$\frac{\partial z}{\partial x} = \frac{\partial \mathbf{y}}{\partial x}^T \frac{\partial z}{\partial \mathbf{y}}$$

一般地，我们可以对多个向量值函数（或标量值函数）复合的函数求偏导，有以下定理：

定理 6.7.1. 假设我们有 n 个列向量 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$, 它们各自的长度为 l_1, l_2, \dots, l_n , 假设 $\mathbf{x}^{(i)}$ 是 $\mathbf{x}^{(i-1)}$ 的一个函数，则对于所有的 $i = 2, 3, \dots, n$ 有

$$\frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(1)}} = \frac{\partial(\mathbf{x}^{(2)})^T}{\partial \mathbf{x}^{(1)}} \frac{\partial(\mathbf{x}^{(3)})^T}{\partial \mathbf{x}^{(2)}} \cdots \frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(n-1)}}$$

此定理即（多元）链式法则。

证明. 根据向量值函数梯度的定义6.6.1和向量值函数微分定理6.6.1，我们应用这个定理在每一对相关向量上，则有

$$d\mathbf{x}^{(2)} = \left(\frac{\partial(\mathbf{x}^{(2)})^T}{\partial \mathbf{x}^{(1)}} \right)^T d\mathbf{x}^{(1)}, d\mathbf{x}^{(3)} = \left(\frac{\partial(\mathbf{x}^{(3)})^T}{\partial \mathbf{x}^{(2)}} \right)^T d\mathbf{x}^{(2)}, \dots, d\mathbf{x}^{(n)} = \left(\frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(n-1)}} \right)^T d\mathbf{x}^{(n-1)}$$

将它们合并起来则有

$$\begin{aligned}\mathrm{d}\boldsymbol{x}^{(n)} &= \left(\frac{\partial(\boldsymbol{x}^{(n)})^T}{\partial \boldsymbol{x}^{(n-1)}} \right)^T \cdots \left(\frac{\partial(\boldsymbol{x}^{(3)})^T}{\partial \boldsymbol{x}^{(2)}} \right)^T \left(\frac{\partial(\boldsymbol{x}^{(2)})^T}{\partial \boldsymbol{x}^{(1)}} \right)^T \mathrm{d}\boldsymbol{x}^{(1)} \\ &= \left(\frac{\partial(\boldsymbol{x}^{(2)})^T}{\partial \boldsymbol{x}^{(1)}} \frac{\partial(\boldsymbol{x}^{(3)})^T}{\partial \boldsymbol{x}^{(2)}} \cdots \frac{\partial(\boldsymbol{x}^{(n)})^T}{\partial \boldsymbol{x}^{(n-1)}} \right)^T \mathrm{d}\boldsymbol{x}^{(1)}\end{aligned}$$

再次应用定理6.6.1可得

$$\frac{\partial(\boldsymbol{x}^{(n)})^T}{\partial \boldsymbol{x}^{(1)}} = \frac{\partial(\boldsymbol{x}^{(2)})^T}{\partial \boldsymbol{x}^{(1)}} \frac{\partial(\boldsymbol{x}^{(3)})^T}{\partial \boldsymbol{x}^{(2)}} \cdots \frac{\partial(\boldsymbol{x}^{(n)})^T}{\partial \boldsymbol{x}^{(n-1)}}$$

□

例 6.7.2. 考虑线性回归中的优化问题：

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^n (\boldsymbol{\theta}^T \boldsymbol{x}_i - y_i)^2$$

我们将其目标函数改写成 $\|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2$ 并关于 $\boldsymbol{\theta}$ 求梯度，其中 $\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)^T$, $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$ ，由链式法则我们有

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\|_2^2 \\ &= \frac{\partial(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^T}{\partial \boldsymbol{\theta}} \frac{\partial \|\mathbf{z}\|_2^2}{\partial \mathbf{z}}, \quad \text{其中 } \mathbf{z} = \mathbf{X}^T \boldsymbol{\theta} - \mathbf{y} \\ &= \mathbf{X}^T \frac{\partial \mathbf{z}^T \mathbf{z}}{\partial \mathbf{z}} \\ &= 2\mathbf{X}^T \mathbf{z} \\ &= 2\mathbf{X}^T \mathbf{X}\boldsymbol{\theta} - \mathbf{X}^T \mathbf{y}\end{aligned}$$

例 6.7.3. 计算 $(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})$ 关于 $\boldsymbol{\mu}$ 的导数，其中 $\boldsymbol{\Sigma}^{-1}$ 是对称矩阵。由链式法则，我们有

$$\begin{aligned}&\frac{\partial [(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})]}{\partial \boldsymbol{\mu}} \\ &= \frac{\partial[(\boldsymbol{x} - \boldsymbol{\mu})^T]}{\partial \boldsymbol{\mu}} \frac{\partial [(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})]}{\partial [\boldsymbol{x} - \boldsymbol{\mu}]} \\ &= \frac{\partial[(\boldsymbol{x} - \boldsymbol{\mu})^T]}{\partial \boldsymbol{\mu}} 2\boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \\ &= -I2\boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \\ &= -2\boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\end{aligned}$$

6.8 反向传播和自动微分

草稿请勿外传

6.8.1 反向传播

在许多机器学习应用中，通过执行梯度下降来找到好的模型参数，这取决于我们可以根据模型参数计算学习目标的梯度。对于给定的目标函数，可以使用微积分和链式法则获得模型参数的梯度。

考虑这个函数

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos(x^2 + \exp(x^2)). \quad (6.34)$$

应用链式法则，注意微分是线性的，计算梯度。

$$\begin{aligned} \frac{df}{dx} &= \frac{2x + 2x \exp(x^2)}{2\sqrt{x^2 + \exp(x^2)}} - \sin(x^2 + \exp(x^2))(2x + 2x \exp(x^2)) \\ &= 2x\left(\frac{1}{2\sqrt{x^2 + \exp(x^2)}} - \sin(x^2 + \exp(x^2))(1 + \exp(x^2))\right). \end{aligned}$$

用这种明确的方式写出梯度通常是不切实际的，因为它常常导致导数的表达式非常冗长。在实践中，这意味着，梯度的实现可能比计算函数要昂贵得多，这是不必要的开销。对于深层神经网络模型的训练，反向传播算法（Kelley, 1960; Bryson, 1961; Dreyfus, 1962; Rumelhart 等人, 1986）是计算与模型参数相关的误差函数梯度的有效方法。

在机器学习中，链式法则在选择层次模型参数（例如，最大似然估计）时起着重要作用。将链式法则用到极致的领域是深度学习，其中函数 y 是函数深度复合来进行计算的。

$$y = (f_K \circ f_{K-1} \circ \cdots \circ f_1)(x) = f_K(f_{K-1}(\cdots(f_1(x))\cdots)), \quad (6.35)$$

其中 x 是输入（例如，图像）， y 是观察值（例如，类标签）且每一个函数 $f_i, i = 1, \dots, K$ 拥有自己的参数。在多层神经网络中，在第 i 层我们有函数 $f_i(x_{i-1}) = \sigma(A_i x_{i-1} + b_i)$ 。其中 x_{i-1} 是第 $i-1$ 层的输出， σ 是激活函数，如 sigmoid, tanh 或一个整流线性单元（ReLU）。为了训练这些模型，我们需要损失函数 L 相对于所有模型参数 $A_j, b_j, j = 1, \dots, K$ 的梯度。这也要求我们计算 L 相对于每层输入的梯度。例如，如果我们有输入 x 和观测 y ，那么网络结构定义为

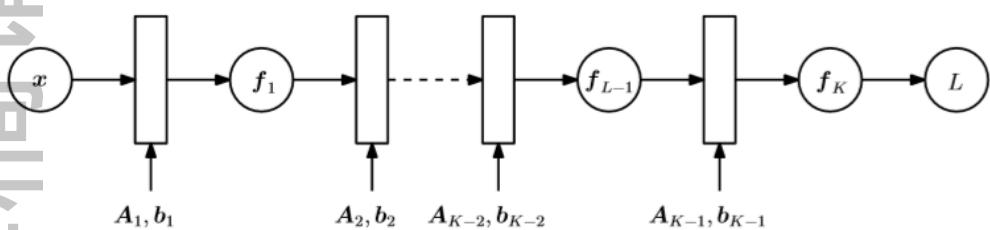


图 6.18: 多层神经网络中的正向传播，用于计算作为输入 x 和参数 A_i, b_i 的函数的损失 L .

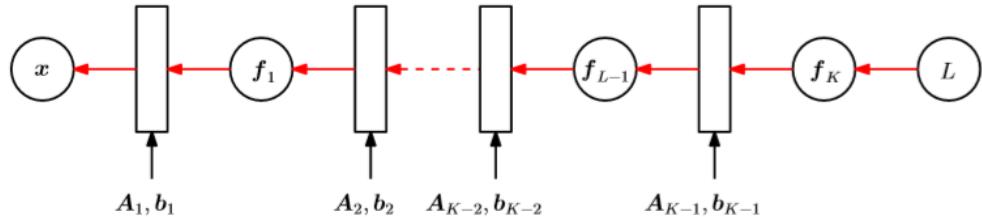


图 6.19: 三阶张量的 3-模式向量积的原理图

观察 y 和由网络结构定义

$$f_0 := x \quad (6.36)$$

$$f_i := \delta_i(\mathbf{A}_{i-1}f_{i-1} + b_{i-1}), i = 1, \dots, K, \quad (6.37)$$

另见图6.18的可视化，我们可能有兴趣发现 \mathbf{A}_j , b_j 为 $j = 0, \dots, K - 1$, 这样的平方损失

$$L(\theta) = \|y - f_K(\theta, x)\|^2 \quad (6.38)$$

最小化，其中 $\theta = \{\mathbf{A}_0, b_0, \dots, \mathbf{A}_{K-1}, b_{K-1}\}$

为了获得相对于参数集 θ 的梯度，我们需要 L 关于每层参数 $\theta_j = \{\mathbf{A}_j, b_j\}$ 的 ($j = 0, \dots, K - 1$) 偏导数。由链式法则可得

$$\frac{\partial L}{\partial \theta_{K-1}} = \frac{\partial L}{\partial f_K} \frac{\partial f_K}{\partial f_{K-1}} \quad (6.39)$$

$$\frac{\partial L}{\partial \theta_{K-2}} = \frac{\partial L}{\partial f_K} \frac{\partial f_K}{\partial f_{K-1}} \frac{\partial f_{K-1}}{\partial f_{K-2}} \quad (6.40)$$

$$\frac{\partial L}{\partial \theta_{K-3}} = \frac{\partial L}{\partial f_K} \frac{\partial f_K}{\partial f_{K-1}} \frac{\partial f_{K-1}}{\partial f_{K-2}} \frac{\partial f_{K-2}}{\partial f_{K-3}} \quad (6.41)$$

$$\frac{\partial L}{\partial \theta_i} = \frac{\partial L}{\partial f_K} \frac{\partial f_K}{\partial f_{K-1}} \dots \frac{\partial f_{i+2}}{\partial f_{i+1}} \frac{\partial f_{i+1}}{\partial \theta_i} \quad (6.42)$$

假设我们已经准备好计算偏导数 $\frac{\partial L}{\partial \theta_{i+1}}$ ，那么大部分计算可以重复使用来计算 $\frac{\partial L}{\partial \theta_i}$ 。图6.19显示了通过网络向后传播的过程。

注记

- 在神经网络的每一层上都有许多节点(神经元)，所以神经网络模型函数是多变量函数，因此上述使用的链式法则是多元链式法则。

2. 使用链式法则以及反向传播，除了大部分计算可以重复使用外，事实上这里还有一个问题就是：链式法则求梯度公式中都是一些 Jacobian 矩阵或梯度向量的乘积，因此这些矩阵之间、矩阵和向量乘法的哪个顺序（沿着链向前或向后）更快？

假设链式法则中有三个因子乘积 $\mathbf{M}_1 \mathbf{M}_2 \mathbf{w}$ ，两个矩阵和一个向量。我们要先做矩阵乘积 $\mathbf{M}_1 \mathbf{M}_2$ 还是先做矩阵向量乘积 $\mathbf{M}_2 \mathbf{w}$ ？对于 $N \times N$ 矩阵， $\mathbf{M}_1 \mathbf{M}_2$ 包含 N^3 个独立的乘法，而 $\mathbf{M}_2 \mathbf{w}$ 有 N^2 个独立的乘法。因此 $(\mathbf{M}_1 \mathbf{M}_2) \mathbf{w}$ 需要 $N^3 + N^2$ 次乘法，而 $\mathbf{M}_1(\mathbf{M}_2 \mathbf{w})$ 仅需要 $N^2 + N^2$ 。这是一个重要的区别。如果我们在神经网络中有来自 L 个层的 L 个矩阵链，则差异本质上是 N 的一个因子：

- 正向 $((\mathbf{M}_1 \mathbf{M}_2) \mathbf{M}_3) \dots \mathbf{M}_L) \mathbf{w}$ 需要 $(L - 1)N^3 + N^2$ 个乘法
- 反向 $\mathbf{M}_1(\mathbf{M}_2(\dots(\mathbf{M}_L \mathbf{w})))$ 需要 LN^2 个乘法

正向和反向顺序之间的选择也出现在矩阵乘法中。如果要求我们将 \mathbf{A} 乘以 \mathbf{B} 乘以 \mathbf{C} ，则结合律为乘法顺序提供了两种选择：

- 首先计算 \mathbf{AB} 还是 \mathbf{BC} ？
- 计算 $(\mathbf{AB})\mathbf{C}$ 还是 $\mathbf{A}(\mathbf{BC})$ ？

他们的结果相同，但单个乘法的数量可能非常不同。假设矩阵 \mathbf{A} 为 $m \times n$, \mathbf{B} 为 $n \times p$ 以及 \mathbf{C} 为 $p \times q$ 。

- 第一种方式 $\mathbf{AB} = (m \times n)(n \times p)$ 具有 mnp 个乘法 $(\mathbf{AB})\mathbf{C} = (m \times p)(p \times q)$ 具有 mpq 个乘法
- 第二种方式 $\mathbf{BC} = (n \times p)(p \times q)$ 具有 njq 个乘法 $\mathbf{A}(\mathbf{BC}) = (m \times n)(n \times q)$ 有 mjq 个乘法

因此我们比较 $mp(n+q)$ 和 $njq(m+p)$ ，将两个数除以 $mnpq$ 就会有结论：当 $\frac{1}{q} + \frac{1}{n} < \frac{1}{m} + \frac{1}{p}$ 时，则第一种方式更快；反之，第二种方式更快。

在深度神经网络中，我们会定义类似如下网络：

$$\mathbf{f}(\mathbf{v}) = \mathbf{A}_L \mathbf{v}_{L-1} = \mathbf{A}_L(R\mathbf{A}_{L-1}(\dots(R\mathbf{A}_2(R\mathbf{A}_1 \mathbf{v}))))$$

我们的目的是优化其中的参数，所以当我们决定了损失函数 $L(\mathbf{f})$ ，我们所要求的就是 L 关于各参数的梯度，即

$$\frac{\partial L}{\partial \mathbf{A}_i} = \frac{\partial \mathbf{v}_i^T}{\partial \mathbf{A}_i} \frac{\partial \mathbf{v}_{i+1}^T}{\partial \mathbf{v}_i} \dots \frac{\partial \mathbf{v}_{L-1}^T}{\partial \mathbf{v}_{L-2}} \frac{\partial \mathbf{f}^T}{\partial \mathbf{v}_{L-1}} \frac{\partial L}{\partial \mathbf{f}}$$

等式的右边恰好为若干个矩阵相乘，并且最后乘以了一个向量。根据前面的结论，我们可以知道按照反向计算可以大大减少计算梯度时的计算量。注： $\frac{\partial \mathbf{v}_i^T}{\partial \mathbf{A}_i}$ 即为 $\frac{\partial \mathbf{v}_i^T}{\partial \text{vec}(\mathbf{A}_i)}$ 。

6.8.2 自动微分

事实证明，反向传播是自动微分的数值分析中的特例。我们可以将自动微分视为一组数字技术（与符号相反），该技术通过使用中间变量和应用链式法则来评估函数的精确（达到机器精

度)梯度。自动微分应用一系列基本算术运算,例如加法和乘法以及基函数,例如 \sin, \cos, \exp, \log 。通过将链式法则应用于这些操作,可以自动计算相当复杂的函数的梯度。自动微分适用于通用计算机程序,具有正向和反向模式。

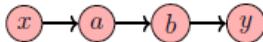


图 6.20: 一个简单的计算图, 显示了数据从 x , 经过中间变量, 最终到 y

图6.20显示计算图, 在该计算图中, 输入数据 x 经过中间变量 a, b 得到输出 y . 如果我们想要去计算梯度 $\frac{dy}{dx}$, 我们将应用链式法则, 最终得到:

$$\frac{dy}{dx} = \frac{dy}{db} \frac{db}{da} \frac{da}{dx} \quad (6.43)$$

直观地, 正向和反向模式在乘法的顺序上是不同。由于矩阵乘法的相关性, 我们可以选择等式(6.44)或(6.45)。

$$\frac{dy}{dx} = \left(\frac{dy}{db} \frac{db}{da} \right) \frac{da}{dx} \quad (6.44)$$

$$\frac{dy}{dx} = \frac{dy}{db} \left(\frac{db}{da} \frac{da}{dx} \right) \quad (6.45)$$

等式(6.44)是反向模式, 因为梯度通过图向后传播, 即与数据流相反。公式(6.45)将是正向模式, 其中梯度随着数据从左到右流过图。

在下文中, 我们将重点关注反向模式自动微分, 即反向传播。在神经网络的背景下, 输入维度通常远高于标签维数, 反向模式在计算上比正向模式容易得多。让我们从一个有教育意义的例子开始。

从(6.34)中考虑函数

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos(x^2 + \exp(x^2)) \quad (6.46)$$

如果我们要在计算机上实现一个函数 f , 我们可以通过使用中间变量来节省一些计算:

$$a = x^2 \quad (6.47)$$

$$b = \exp(a) \quad (6.48)$$

$$c = a + b \quad (6.49)$$

$$d = \sqrt{c} \quad (6.50)$$

$$e = \cos(c) \quad (6.51)$$

深度学习
从入门到实践

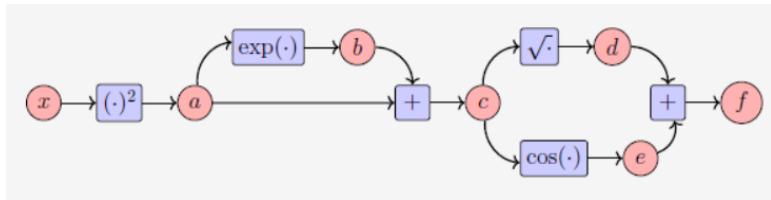


图 6.21: 输入 x, 函数 f 以及中间变量为 a,b,c,d,e 的计算图.

$$f = d + e \quad (6.52)$$

这与应用链式法则时所发生的思考过程是一样的。注意，上述方程组所需的操作比直接实现(6.34)中定义的函数 $f(x)$ 所需的操作要少。图6.21中相应的计算图显示了获取函数值 f 所需的数据流和计算。

包含中间变量的方程组可以看作是一个计算图，一种广泛应用于神经网络软件库实现的表示形式。通过回顾初等函数导数的定义，我们可以直接计算中间变量对其相应输入的导数，可得：

$$\frac{\partial a}{\partial x} = 2x \quad (6.53)$$

$$\frac{\partial b}{\partial a} = \exp(a) \quad (6.54)$$

$$\frac{\partial c}{\partial a} = 1 = \frac{\partial c}{\partial b} \quad (6.55)$$

$$\frac{\partial d}{\partial c} = \frac{1}{2\sqrt{c}} \quad (6.56)$$

$$\frac{\partial e}{\partial c} = -\sin(c) \quad (6.57)$$

$$\frac{\partial f}{\partial d} = 1 = \frac{\partial f}{\partial e} \quad (6.58)$$

通过看图6.21中的计算图，我们通过输出的反向传播计算出 $\frac{\partial f}{\partial x}$ ，并且我们可以得到下面的关系：

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial d} \frac{\partial d}{\partial c} + \frac{\partial f}{\partial e} \frac{\partial e}{\partial c} \quad (6.59)$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} \frac{\partial c}{\partial b} \quad (6.60)$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} \frac{\partial b}{\partial a} + \frac{\partial f}{\partial c} \frac{\partial c}{\partial a} \quad (6.61)$$

传外勿请稿草

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a} \frac{\partial a}{\partial x} \quad (6.62)$$

注意，我们隐含地应用了链式法则来获得 $\frac{\partial f}{\partial x}$ ，通过替换初等函数的导数，我们得到

$$\frac{\partial f}{\partial c} = 1 \cdot \frac{1}{2\sqrt{c}} + 1 \cdot (-\sin(c)) \quad (6.63)$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} \cdot 1 \quad (6.64)$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} \exp(a) + \frac{\partial f}{\partial c} \cdot 1 \quad (6.65)$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a} \cdot 2x \quad (6.66)$$

通过把上面的每一个导数看作一个变量，我们观察到计算导数所需要的计算与函数本身的计算具有相似的复杂性。这是非常违反直觉的，因为导数的数学表达式要比函数 $f(x)$ 的数学表达式复杂得多。

自动微分是上述示例的形式化描述。令 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。然后将计算图表示为方程：

$$\text{For } i = d+1, \dots, D : \quad x_i = g_i(x_{Pa(x_i)}) \quad (6.67)$$

其中 $g_i(\cdot)$ 是基函数， $x_{Pa(x_i)}$ 是图中变量 x_i 的父节点。给定以这种方式定义的函数，我们可以使用链式法则以逐步的方式计算函数的导数。回想一下，根据定义 $f = x_D$ 因此

$$\frac{\partial x}{\partial x_D} = 1 \quad (6.68)$$

对于其他变量 x_i ，我们应用链式法则

$$\frac{\partial f}{\partial x_i} = \sum_{x_j: x_j \in Pa(x_j)} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i} = \sum_{x_j: x_j \in Pa(x_j)} \frac{\partial f}{\partial x_j} \frac{\partial g_j}{\partial x_i} \quad (6.69)$$

其中 $Pa(x_j)$ 是计算图中 x_j 的父节点集。公式(6.67)是函数的前向传播，而(6.69)是通过计算图的梯度的反向传播。对于神经网络训练，我们反向传播关于标签的预测误差。

只要我们有一个可以表示为计算图的函数，上面的自动微分方法就可以工作，其中基函数是可微的。实际上，该函数甚至可能不是数学函数而是计算机程序。然而，并非所有计算机程序都可以自动微分，例如，如果我们找不到微分基函数：编程结构，例如 for 循环和 if 语句也需要更多的关注。

例 6.8.1. 我们考虑一个两层的全连接神经网络：

$$y = f(\mathbf{x}) = \text{ReLU}(\mathbf{A}_2(\text{ReLU}(\mathbf{A}_1\mathbf{x} + \mathbf{b}_1)) + \mathbf{b}_2)$$

其中

$$\mathbf{A}_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -2 & 1 \end{pmatrix}, \mathbf{A}_2 = \begin{pmatrix} 1 & -2 & 1 \\ 2 & -1 & 0 \end{pmatrix}, \mathbf{b}_1 = \begin{pmatrix} 1 \\ -2 \\ -1 \end{pmatrix}, \mathbf{b}_2 = \begin{pmatrix} -2 \\ -3 \end{pmatrix}$$

假设输入为 $\mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, 并且对应的真实输出为 $\mathbf{y}' = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, 采用平方损失 $L = \frac{1}{2}\|\mathbf{y} - \mathbf{y}'\|_2^2$ 。试计算函数 L 关于 $\mathbf{b}_1, \mathbf{b}_2$ 的梯度。

解. 先计算前项过程:

$$\mathbf{A}_1 \mathbf{x} + \mathbf{b}_1 = \begin{pmatrix} 2 \\ -2 \\ -2 \end{pmatrix}, \mathbf{A}_2(\text{ReLU}(\mathbf{A}_1 \mathbf{x} + \mathbf{b}_1)) + \mathbf{b}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

故 $\mathbf{y} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ 从而 $L = 1$ 。记

$$\mathbf{k} = \text{ReLU}(\mathbf{A}_1 \mathbf{x} + \mathbf{b}_1)$$

然后分别计算

$$\frac{\partial L}{\partial \mathbf{y}} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \frac{\partial \mathbf{y}^T}{\partial \mathbf{b}_2} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \frac{\partial \mathbf{y}^T}{\partial \mathbf{k}} = \begin{pmatrix} 1 & 2 \\ -2 & -1 \\ 1 & 0 \end{pmatrix}, \frac{\partial \mathbf{k}^T}{\partial \mathbf{b}_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

所以有

$$\frac{\partial L}{\partial \mathbf{b}_1} = \frac{\partial \mathbf{k}^T}{\partial \mathbf{b}_1} \frac{\partial \mathbf{y}^T}{\partial \mathbf{k}} \frac{\partial L}{\partial \mathbf{y}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial L}{\partial \mathbf{b}_2} = \frac{\partial \mathbf{y}^T}{\partial \mathbf{b}_2} \frac{\partial L}{\partial \mathbf{y}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

大多数深度学习算法都涉及某种形式的优化。优化指的是改变 \mathbf{x} 以最小化或最大化某个函数 $f(\mathbf{x})$ 的任务。我们通常以最小化 $f(\mathbf{x})$ 指代大多数最优化问题。最大化可经由最小化算法最小化 $-f(\mathbf{x})$ 来实现。

我们把要最小化或最大化的函数称为目标函数或准则。当我们对其进行最小化时，我们也把它称为代价函数、损失函数或误差函数。

我们通常使用一个上标 * 表示最小化或最大化函数的 \mathbf{x} 值。如我们记 $\mathbf{x}^* = \arg \min f(\mathbf{x})$ 。

我们假设读者已经熟悉微积分，这里简要回顾微积分概念如何与优化联系。

假设我们有一个函数 $y = f(x)$, 其中 x 和 y 是实数。这个函数的导数记为 $f'(x)$ 或 $\frac{dy}{dx}$ 。导数 $f'(x)$ 代表 $f(x)$ 在点 x 处的斜率。换句话说，它表明如何缩放输入的小变化才能在输出获得相应的变化: $f(x + \epsilon) \approx f(x) + \epsilon f'(x)$ 。

因此导数对于最小化一个函数很有用，因为它告诉我们如何更改 x 来略微地改善 y 。例如，我们知道对于足够小的 ϵ 来说， $f(x - \epsilon \text{sign}(f'(x)))$ 是比 $f(x)$ 小的。因此我们可以将 x 往导数的反方向移动一小步来减小 $f(x)$ 。这种技术被称为梯度下降。

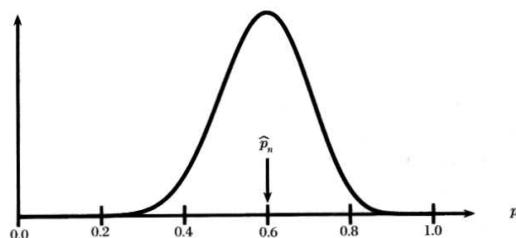


图 6.22: 梯度下降算法的例子

当 $f'(x) = 0$, 导数无法提供往哪个方向移动的信息。 $f'(x) = 0$ 的点称为临界点或驻点。一局部极小点意味着这个点的 $f(x)$ 小于所有邻近点, 因此不可能通过移动无穷小的步长来减小 $f(x)$ 。一个局部最大点意味着这个点的 $f(x)$ 大于所有邻近点, 因此不可能通过移动无穷小的步长来增大 $f(x)$ 。有些临界点既不是最小点也不是最大点。这些点被称为鞍点。

使 $f(x)$ 取得绝对的最小值 (相对所有其他值) 的点是全局最小。函数可能只有一个全局最小或存在多个全局最小, 还可能存在不是全局最优的局部最小。在深度学习的背景下, 我们要优化的函数可能含有许多不是最优的局部最小, 或者还有很多处于非常平坦的区域内的鞍点。尤其是当输入是多维的时候, 所有这些都将使优化变得困难。因此, 我们通常寻找使 f 非常小的点, 但这任何形式意义下并不一定是最小。

我们经常最小化具有多维输入的函数: $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 。为了使“最小化”的概念有意义, 输出必须是一维的 (标量)。

针对具有多维输入的函数, 我们需要用到偏导数的概念。偏导数 $\frac{\partial}{\partial x_i} f(\mathbf{x})$ 衡量点 \mathbf{x} 处只有 x_i 增加时 $f(\mathbf{x})$ 如何变化。梯度是相对一个向量求导的导数: f 的导数是包含所有偏导数的向量, 记为 $\nabla_{\mathbf{x}} f(\mathbf{x})$ 。梯度的第 i 个元素是 f 关于 x_i 的偏导数。在多维情况下, 驻点是梯度中所有元素都为零的点。

在 \mathbf{u} (单位向量) 方向的方向导数是函数 f 在 \mathbf{u} 方向的斜率。换句话说, 方向导数是函数 $f(\mathbf{x} + \alpha \mathbf{u})$ 关于 α 的导数 (在 $\alpha = 0$ 时取得)。使用链式法则, 我们可以看到当 $\alpha = 0$ 时, $\frac{\partial}{\partial \alpha} f(\mathbf{x} + \alpha \mathbf{u}) = \mathbf{u}^\top \nabla_{\mathbf{x}} f(\mathbf{x})$ 。

为了最小化 f , 我们希望找到使 f 下降得最快的方向。计算方向导数:

$$\begin{aligned} & \min_{\mathbf{u}, \mathbf{u}^\top \mathbf{u}=1} \mathbf{u}^\top \nabla_{\mathbf{x}} f(\mathbf{x}) \\ &= \min_{\mathbf{u}, \mathbf{u}^\top \mathbf{u}=1} \|\mathbf{u}\|_2 \|\nabla_{\mathbf{x}} f(\mathbf{x})\|_2 \cos \theta \end{aligned}$$

其中 θ 是 \mathbf{u} 与梯度的夹角。将 $\|\mathbf{u}\|_2 = 1$ 代入, 并忽略与 \mathbf{u} 无关的项, 就能简化得到 $\min_{\mathbf{u}} \cos \theta$ 。这在 \mathbf{u} 与梯度方向相反时取得最小。换句话说, 梯度向量指向上坡, 负梯度向量指向下坡。我们在负梯度方向上移动可以减小 f 。这被称为最速下降法 (method of steepest descent) 或梯度下降法。

最速下降建议新的点为

$$\mathbf{x}' = \mathbf{x} - \epsilon \nabla_{\mathbf{x}} f(\mathbf{x})$$

其中 ϵ 为学习率，是一个确定步长大小的正标量。我们可以通过几种不同的方式选择 ϵ 。普遍的方式是选择一个小常数。有时我们通过计算，选择使方向导数消失的步长。还有一种方法是根据几个 ϵ 计算 $f(\mathbf{x} - \epsilon \nabla_{\mathbf{x}} f(\mathbf{x}))$ ，并选择其中能产生最小目标函数值的 ϵ 。这种策略被称为线性搜索。

最速下降在梯度的每一个元素为零时收敛（或在实践中，很接近零时）。在某些情况下，我们也许能够避免运行该迭代算法，并通过解方程 $\nabla_{\mathbf{x}} f(\mathbf{x}) = 0$ 直接跳到驻点。

虽然梯度下降是连续空间中的优化方法，但不断向更好的情况移动一小步（即近似最佳的小移动）的一般概念可以推广到离散空间。递增带有离散参数的目标函数被称为爬山 (hill climbing) 算法。

6.9 高阶微分和泰勒展开

6.9.1 Hessian 矩阵

我们前面已经讨论过了梯度，即一阶导数。有时我们会对高阶导数感兴趣，比如在优化中使用牛顿法时我们需要二阶导数。在一元的情况下，我们可以使用泰勒展开构造多项式来逼近函数，在多元情况下，我们同样可以这么做。

定义 6.9.1. 设函数 $y = f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ 。 $f(\mathbf{x})$ 的 **Hessian 矩阵**被定义为

$$\mathbf{H} = \frac{\partial}{\partial \mathbf{x}^T} \frac{\partial f}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

记作 $\nabla^2 f$ 。

求函数的 Hessian 矩阵可以用二步法求出：(1) 求实值函数 $f(\mathbf{x})$ 关于向量变元 \mathbf{x} 的偏导数，得到实值函数的梯度 $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ 。(2) 再求梯度 $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ 相对于 $1 \times n$ 行向量 \mathbf{x}^T 的偏导数，得到梯度的梯度即 Hessian 矩阵。根据以上步骤，容易得到 Hessian 矩阵的下列公式。

- 对于 $n \times 1$ 常数向量 \mathbf{a} ，有

$$\frac{\partial^2 \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x} \partial \mathbf{x}^T} = \mathbf{O}_{n \times n} \quad (6.70)$$

- 若 \mathbf{A} 是 $n \times n$ 矩阵，则

$$\frac{\partial^2 \mathbf{x}^T \mathbf{A} \mathbf{x}}{\partial \mathbf{x} \partial \mathbf{x}^T} = \mathbf{A} + \mathbf{A}^T \quad (6.71)$$

- 令 \mathbf{x} 为 $n \times 1$ 向量, \mathbf{a} 为 $m \times 1$ 常数向量, \mathbf{A} 和 \mathbf{B} 分别为 $m \times n$ 和 $m \times m$ 常数矩阵, 且 \mathbf{B} 为对称矩阵, 则

$$\frac{\partial^2 (\mathbf{a} - \mathbf{Ax})^T \mathbf{B} (\mathbf{a} - \mathbf{Ax})}{\partial \mathbf{x} \partial \mathbf{x}^T} = 2\mathbf{A}^T \mathbf{BA} \quad (6.72)$$

- 若 \mathbf{A} 是一个与向量 \mathbf{x} 无关的矩阵, 而 $\mathbf{y}(\mathbf{x})$ 是与向量 \mathbf{x} 的元素有关的列向量, 则

$$\begin{aligned} \frac{\partial^2 [\mathbf{y}(\mathbf{x})]^T \mathbf{Ay}(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} &= \frac{\partial [\mathbf{y}(\mathbf{x})]^T}{\partial \mathbf{x}} (\mathbf{A} + \mathbf{A}^T) \frac{\partial \mathbf{y}(\mathbf{x})}{\partial \mathbf{x}^T} + \\ &([\mathbf{y}(\mathbf{x})]^T (\mathbf{A} + \mathbf{A}^T) \otimes \mathbf{I}_n) \frac{\partial}{\partial \mathbf{x}^T} \left[\text{vec} \left(\frac{\partial [\mathbf{y}(\mathbf{x})]^T}{\partial \mathbf{x}} \right) \right] \end{aligned} \quad (6.73)$$

Hessian 矩阵在机器学习优化中有很多应用。如果 $f(\mathbf{x})$ 是二次 (连续) 可微的函数, 则二阶偏导可交换, 也即二阶偏导与微分的顺序无关, 此时 Hessian 矩阵是对称矩阵。在凸优化的章节中, 我们将会学到在函数的极小点处 Hessian 矩阵为正定矩阵。Hessian 矩阵也被应用于二阶优化算法, 如牛顿法能够快速的收敛到最优点。

6.9.2 线性化和多元泰勒级数

一个函数的梯度 ∇f 通常可以作为 \mathbf{x}_0 附近的局部线性逼近

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + (\nabla f(\mathbf{x}_0))^T (\mathbf{x} - \mathbf{x}_0)$$

这里 $(\nabla f(\mathbf{x}_0))^T$ 是 f 关于 \mathbf{x} 的梯度在 \mathbf{x}_0 处的取值。即通过一条直线来逼近函数 f , 这种逼近是局部准确的, 但是在更大范围内是有很大误差的。上式实际上是函数 f 在 \mathbf{x}_0 处泰勒展开的前两项, 它是 $f(\mathbf{x})$ 在 \mathbf{x}_0 处的高阶多元泰勒级数展开的特殊情形。

定义 6.9.2. 对于多元泰勒展开, 我们考虑函数 $f : \mathbb{R}^D \rightarrow \mathbb{R}, \mathbf{x} \rightarrow f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^D$ 在 \mathbf{x}_0 处光滑。

如果我们定义差分向量 $\Delta := \mathbf{x} - \mathbf{x}_0$, 那么 f 在 \mathbf{x}_0 处的泰勒展开为

$$f(\mathbf{x}) = \sum_{k=0}^{\infty} \frac{D_x^k f(\mathbf{x}_0)}{k!} \Delta^k$$

其中, $D_x^k f(\mathbf{x}_0)$ 是 f 关于 \mathbf{x} 的 k 阶全微分在 \mathbf{x}_0 处的取值。

定义 6.9.3. 函数 f 在 \mathbf{x}_0 处的 n 阶泰勒多项式被定义为泰勒展开的前 $n+1$ 项:

$$T_n = \sum_{k=0}^n \frac{D_x^k f(\mathbf{x}_0)}{k!} \Delta^k$$

注意当 $D > 1, k > 1$ 时, 我们在上面使用的简写记号 Δ^k 并没有在 \mathbb{R}^D 中定义。这里的 $D_x^k f, \Delta^k$ 都是 k 阶张量, $\Delta^k \in \mathbb{R}^{D \times D \times \dots \times D}$ 是通过张量积 (用符号 \otimes) 得到的。例如

$$\Delta^2 = \Delta \otimes \Delta = \Delta \Delta^T, \Delta^2[i, j] = \delta[i]\delta[j]$$

$$\Delta^3 = \Delta \otimes \Delta \otimes \Delta, \Delta^3[i, j, k] = \delta[i]\delta[j]\delta[k]$$

在泰勒展开中，我们得到以下式子

$$D_x^k \mathbf{f}(\mathbf{x}_0) \Delta^k = \sum_a \cdots \sum_k D_x^k \mathbf{f}(\mathbf{x}_0)[a, \dots, k] \delta[a] \dots \delta[k]$$

其中

$$D_x^k \mathbf{f}(\mathbf{x})[i_1, \dots, i_k] = \frac{\partial^k}{\partial x_{i_1} \dots \partial x_{i_k}} \mathbf{f}(\mathbf{x})$$

所以 $D_x^k \mathbf{f}(\mathbf{x}_0) \Delta^k$ 包含了所有 k 次多项式。

$$k = 0 : D_x^0 \mathbf{f}(\mathbf{x}_0) \Delta^0 = \mathbf{f}(\mathbf{x}_0) \in \mathbb{R}$$

$$k = 1 : D_x^1 \mathbf{f}(\mathbf{x}_0) \Delta^1 = \nabla_x \mathbf{f}(\mathbf{x}_0) \Delta = \sum_i \nabla \mathbf{f}(\mathbf{x}_0)[i] \delta[i] \mathbb{R}$$

$$k = 2 : D_x^2 \mathbf{f}(\mathbf{x}_0) \Delta^2 = \Delta^T \mathbf{H} \Delta = \sum_i \sum_j H[i, j] \delta[i] \delta[j] \in \mathbb{R}$$

$$k = 3 : D_x^3 \mathbf{f}(\mathbf{x}_0) \Delta^3 = \sum_i \sum_j \sum_k D_x^3 f(\mathbf{x}_0)[i, j, k] \delta[i] \delta[j] \delta[k] \in \mathbb{R}$$

例 6.9.1. 求函数 $f(\mathbf{x}) = \mathbf{a}^T e^{\mathbf{x}}$ 在 $\mathbf{0}$ 处的 2 阶泰勒多项式。

解：根据泰勒展开我们有

$$T_2 = f(\mathbf{0}) + (\nabla_x f(\mathbf{0}))^T (\mathbf{x} - \mathbf{0}) + \frac{1}{2} (\mathbf{x} - \mathbf{0})^T (\nabla_x^2 f(\mathbf{0})) (\mathbf{x} - \mathbf{0})$$

通过计算可得

$$\nabla_x f(\mathbf{x}) = (\mathbf{a}_1 e^{x_1}, \mathbf{a}_2 e^{x_2}, \dots, \mathbf{a}_n e^{x_n})^T$$

$$\nabla_x^2 f(\mathbf{x}) = \text{diag}(\nabla_x f(\mathbf{x}))$$

$$\text{所以 } f(\mathbf{0}) = \sum_{i=1}^n \mathbf{a}_i, \nabla_x f(\mathbf{0}) = \mathbf{a}, \nabla_x^2 f(\mathbf{0}) = \text{diag}(\mathbf{a})$$

$$\text{故 } T_2 = \sum_{i=1}^n \mathbf{a}_i + \mathbf{a}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \text{diag}(\mathbf{a}) \mathbf{x} = \sum_{i=1}^n \mathbf{a}_i (1 + x_i + \frac{1}{2} x_i^2)$$

6.10 阅读材料

本章主要介绍向量和矩阵微分，包括向量和矩阵函数，以及数据科学中常见的各种函数（包括模型函数、损失函数、目标函数、非线性激活函数等）、深度神经网络中函数的构造，梯度和高阶导数的定义和性质、向量值函数和矩阵函数的梯度求解方法以及用迹微分法求梯度的方法，并引入深度网络中的梯度和自动微分求解方法。这一章介绍的函数模型是数据科学中两大类型的模型之一。这些内容将在优化方法介绍和数据科学中的各种优化问题求解中反复使用。更多矩阵微分的细节和所需要的线性代数的简短回顾可以在 Magnus 和 Neudecker(2007) 中找到。自动微分有很长的一段历史，读者可以参考 Griewank 和 Walther (2003, 2008); Elliott(2009) 和他们的引用。

此外，在数据分析和机器学习领域，我们经常需要计算期望，例如我们需要解这种形式的积分

$$E[f(\mathbf{x})] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \quad (6.74)$$

即使 $p(\mathbf{x})$ 是一种简便的形式（如高斯函数），这个积分通常也不能解析求解。用 f 的泰勒级数展开是找到近似解的一种方法：假设 $p(\mathbf{x}) = \mathcal{N}(\mu, \Sigma)$ 是高斯函数，然后将非线性函数 f 用关于 μ 的一阶泰勒级数展开线性化。对于线性函数，如果 $p(x)$ 是高斯分布，我们可以精确地计算均值（和协方差）。扩展卡尔曼滤波器（Maybeck, 1979）在非线性动力系统（也称为“状态空间模型”）中的在线状态估计中充分利用了这一特性。其他确定性的方法来逼近上述积分的有的 unscented transform（Julier 和 Uhlmann, 1997），这个方法不需要任何梯度信息。或者拉普拉斯近似（Bishop, 2006），它使用 Hessian 在后均值处对 $p(\mathbf{x})$ 进行局部高斯近似。

习题

习题 6.1. 计算导数

$$f(\mathbf{x}) = \log(x^4) \sin(x^3)$$

习题 6.2. 计算导数

$$f(\mathbf{x}) = \frac{1}{1 + \exp(-x)}$$

习题 6.3. 计算导数

$$f(\mathbf{x}) = \exp\left(-\frac{1}{2\sigma^2}(\mathbf{x} - \boldsymbol{\mu})^2\right)$$

这里的 μ, σ 都是常量

习题 6.4. 当 $x_0 = 0$ 时计算泰勒多项式 T_n , $f(\mathbf{x}) = \sin(\mathbf{x}) + \cos(\mathbf{x})$, 其中 $n = 0, \dots, 5$

习题 6.5. 有以下函数

$$f_1(\mathbf{x}) = \sin(\mathbf{x}_1) \cos(\mathbf{x}_2), \mathbf{x} \in \mathbb{R}^2$$

$$f_1(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$$

$$f_1(\mathbf{x}, \mathbf{y}) = \mathbf{x}, \mathbf{x} \in \mathbb{R}^n$$

1. $\frac{\partial f_1}{\partial \mathbf{x}}$ 的维数是多少？

2. 计算雅克比

习题 6.6. f 对 t 求导, g 对 X 求导, 其中

$$f(\mathbf{t}) \sin(\log(\mathbf{t}^T \mathbf{t})), \mathbf{t} \in \mathbb{R}^D$$

$$g(X) = \text{Tr}(\mathbf{A} \mathbf{X} \mathbf{B}), \mathbf{A} \in \mathbb{R}^{D \times E}, \mathbf{X} \in \mathbb{R}^{E \times F}, \mathbf{B} \in \mathbb{R}^{F \times D}$$

Tr 表示迹

习题 6.7. 用链式法则计算下列函数的导数 $\frac{df}{dx}$, 给出每个偏导数的维数, 详细描述你的步骤。
1.

$$f(z) = \log(1 + z), z = \mathbf{x}^T \mathbf{x}, \mathbf{x} \in \mathbb{R}^D$$

2.

$$f(z) = \sin(z), z = \mathbf{A}\mathbf{x} + b, \mathbf{A} \in \mathbb{R}^{E*D}, \mathbf{x} \in \mathbb{R}^D, b \in \mathbb{R}^E$$

其中 $\sin(\cdot)$ 作用于每个 z 元素

习题 6.8. 计算下列函数的导数 $\frac{df}{dx}$, 详细描述你的步骤。

1. 使用链式法则, 计算每个偏导数的维数。

$$f(z) = \exp\left(-\frac{1}{2}z\right)$$

$$z = g(y) = \mathbf{y}^T \mathbf{S}^{-1} \mathbf{y}$$

$$\mathbf{y} = h(\mathbf{x}) = \mathbf{x} - \mu$$

其中 $\mathbf{x}, \mu \in \mathbb{R}^D, \mathbf{S} \in \mathbb{R}^{D*D}$

2.

$$f(\mathbf{x}) = \text{Tr}(\mathbf{x}\mathbf{x}^T + \sigma^2 \mathbf{I}), \mathbf{x} \in \mathbb{R}^D$$

这里 $\text{Tr}(\mathbf{A})$ 是 \mathbf{A} 的迹, 即所有对角元素之和。提示: 需要明确写出外积。

3. 使用链式法则。给出每个偏导数的维数。不需要明确地计算偏导数的乘积。

$$f = \tanh(z) \in \mathbb{R}^M$$

$$z = \mathbf{A}\mathbf{z} + b, \mathbf{x} \in \mathbb{R}^N, \mathbf{A} \in \mathbb{R}^{M*N}, b \in \mathbb{R}^M$$

这里的 \tanh 作用于 z 的每一个分量。

习题 6.9. 构建模型使得预测值与真实值的误差最小常用向量 2-范数度量, 求解模型过程中需要计算梯度, 求梯度:

- $f(\mathbf{A}) = \frac{1}{2} \|\mathbf{Ax} + \mathbf{b} - \mathbf{y}\|_2^2$, 求 $\frac{\partial f}{\partial \mathbf{A}}$
- $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{Ax} + \mathbf{b} - \mathbf{y}\|_2^2$, 求 $\frac{\partial f}{\partial \mathbf{x}}$

习题 6.10. 求 $\frac{\partial \text{Tr}(\mathbf{W}^{-1})}{\partial \mathbf{W}}$, 利用迹微分法求解

习题 6.11. 二次型是数据分析中常用函数, 求 $\frac{\partial \mathbf{x}^T \mathbf{Ax}}{\partial \mathbf{x}}, \frac{\partial \mathbf{x}^T \mathbf{Ax}}{\partial \mathbf{A}}$

习题 6.12. $f(z) = \frac{\exp(z)}{1^T \exp(z)}$ 称为 softmax 函数, $(\exp(z))_i = \exp(z_i)$, 如果 $\mathbf{q} = \frac{\exp(z)}{1^T \exp(z)}, J = -\mathbf{p}^T \log(\mathbf{q})$, 其中 $\mathbf{p}, \mathbf{q}, \mathbf{z} \in \mathbb{R}^n$, 并且 $1^T \mathbf{p} = 1$,

- 证: $\frac{\partial J}{\partial z} = \mathbf{q} - \mathbf{p}$
- 若 $\mathbf{z} = \mathbf{Wx}$, 其中 $\mathbf{W} \in \mathbb{R}^{n*m}, \mathbf{x} \in \mathbb{R}^m$, $\frac{\partial J}{\partial \mathbf{W}} = (\mathbf{q} - \mathbf{p})\mathbf{x}^T$ 是否成立。

习题 6.13. 以下内容是求解正态分布模型的关键步骤: $L = -\frac{Nd}{2} \ln(2\pi) - \frac{N}{2} \ln|\Sigma| - \frac{1}{2} \sum_t (\mathbf{x}_t - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}_t - \boldsymbol{\mu})$

1) 求 $\frac{\partial L}{\partial \boldsymbol{\mu}}$

2) 当 $\boldsymbol{\mu} = \frac{1}{N} \sum_t \mathbf{x}_t$ 时求 $\frac{\partial L}{\partial \Sigma}$, 求 Σ 使 $\frac{\partial L}{\partial \Sigma} = 0, \mathbf{x} \in \mathbb{R}^N, \Sigma \in \mathbb{R}^{N \times N}$

习题 6.14. 求 $\frac{\partial |X^k|}{\partial X}$

习题 6.15. 求 $\frac{\partial \text{Tr}(AXBX^T C)}{\partial X}$

参考文献

- [1] Magnus, Jan R., and Neudecker, Heinz. 2007. Matrix Differential Calculus with Applications in Statistics and Econometrics. 3rd edn. John Wiley & Sons. pages 166
- [2] Griewank, Andreas, and Walther, Andrea. 2003. Introduction to Automatic Differentiation. PAMM, 2(1), 45–49. pages 166
- [3] Griewank, Andreas, and Walther, Andrea. 2008. Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation. second edn. SIAM, Philadelphia. pages 166
- [4] Julier, Simon J., and Uhlmann, Jeffrey K. 1997. A New Extension of the Kalman Filter to Non-linear Systems. Pages 182–193 of: Proceedings of AeroSense: 11th Symposium on Aerospace/Defense Sensing, Simulation and Controls. pages 167
- [5] Maybeck, Peter S. 1979. Stochastic Models, Estimation, and Control. Mathematics in Science and Engineering, vol. 141. Academic Press, Inc. pages 167
- [6] Bishop, Christopher M. 2006. Pattern Recognition and Machine Learning. Information Science and Statistics. Springer-Verlag. pages vii, 2, 90, 167, 171, 173, 184, 206, 210, 258, 259, 263, 279, 294, 346, 347, 371, 372

草稿请勿
修改

第七章 概率基础

在第二章我们引入了向量和矩阵来对数据进行确定性表示。然而，在数据科学中，我们所遇到数据问题充满了大量的不确定性和随机性。

这种不确定性和随机性可能来自多个方面：

- (1) 被建模系统内在的随机性；
- (2) 不完全观测；
- (3) 不完全建模。

粗略的说，概率可以被看作对不确定性的处理研究，它可以被认为是一个事件发生的次数的分数，或者是对一个事件的信任程度。因此我们可以利用这个概率来确定在实验中发生某种事情的可能性。正如第1章中所提到的，我们通常希望量化不确定性：数据中的不确定性、机器学习模型中的不确定性以及模型生成的预测中的不确定性。量化不确定性需要一个随机变量的概念，这是一个将随机实验结果映射到实数的函数。与随机变量相关联的是一个数字，对应于每个可能的结果到实数的映射。这组数字指定发生概率，称为概率分布。

7.1 概率论基本概念回顾：数据不确定性描述的观点

概率是描述不确定性的数学语言。概率论是研究不确定现象统计规律的一门学科。在数据科学中，数据通过采样得到的，具有一定的不确定性，它的结果是通过观测得到的，也具有一定的不确定性，因此使用概率模型对真实数据统计规律进行建模模拟。

本节将回顾概率论的基本概念。

7.1.1 概率论基本概念

人类在自然界的生产实践中，观察到的现象大致分为确定性现象和不确定现象两类。在中学阶段的物理课程中，我们一般学习研究确定性现象，比如：太阳肯定会东方升起；标准气压下，水在 100°C 会沸腾。

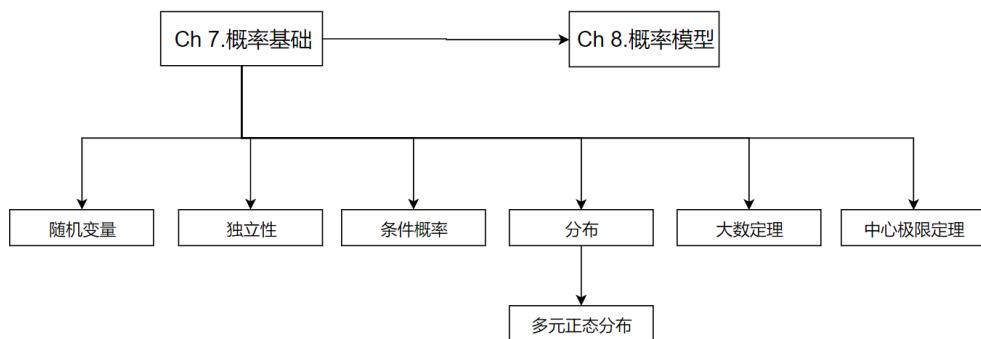


图 7.1: 本章导图

在数据科学中，我们经常研究不确定现象（随机现象），不确定现象在个别试验中呈现不确定结果，大量试验后呈现统计规律：

例 7.1.1. 给定一个查询 Q 和文档集 $D = \{d_1, d_2, d_3, \dots, d_n\}$ ，从 D 随机抽取一篇文 d_i ，查看 d_i 是否与查询相关。

例 7.1.2. 观察 A 股票在中午 12 点时的股价。

例 7.1.3. 从一部电影的众多影评中随机抽取一条影评，观察该影评是正类影评还是负类影评。

以上三个试验都具有可重复、结果多样、结果不可预测这三个特点，因此也被称为随机试验，简称试验。

样本空间、样本点和事件

样本空间是随机试验所有可能结果的集合，记作 Ω 。每一种试验结果称样本空间中的一个样本点。

例 7.1.1 中，给定一个查询 Q 和文档集 D ，从文档集中的随机抽取文档 d_j 的试验的样本空间 $\Omega = \{\text{相关, 不相关}\}$ 。即 d_j 要么与查询相关，要么不相关。”相关”是样本空间中的一个样本点，“不相干”也是一个样本点。

满足某些条件的样本点组成样本空间的子集称为随机事件，简称事件。例 7.1.1 中从文档集 D 中抽取与查询 Q 相关的文档是一随机事件。例 7.1.2 中股票的价格大于 100 是一个随机事件。例 7.1.3 中影评是正类影评是一个随机事件。需要注意的是：

- 一个样本点也属于一个事件。

- 空集 \emptyset 是样本空间 Ω 的子集，称为不可能事件.
- Ω 是它自己的子集，称为必然事件.

事件的关系与运算

事件是样本点的集合，事件之间的关系与运算可以按照集合之间的关系与集运算来规定。

给定一个随机试验， Ω 是试验的样本空间，事件 A, B, C 是 Ω 的子集。下列给出事件之间的 7 种关系。

包含关系 如果 $A \subset B$ 或 $B \supset A$ ，称事件 B 包含事件 A 。它的含义是：若事件 A 发生，则事件 B 必然发生。

相等关系 如果 $A \subset B$ 且 $A \supset B$ ，称事件 B 与事件 A 相等。

事件和 $A \cup B = \{\omega : \omega \in A \text{ 或 } \omega \in B\}$ 称事件 A 与事件 B 的和事件。它的含义是：当且仅当事件 A 与事件 B 中至少一个发生时，事件 $A \cup B$ 发生。

事件积 事件 $A \cap B = \{\omega : \omega \in A \text{ 且 } \omega \in B\}$ 称事件 A 与事件 B 的积事件。它的含义是：当且仅当事件 A 与事件 B 中同时发生时，事件 $A \cap B$ 发生。

事件差 事件 $A - B = \{\omega : \omega \in A \text{ 且 } \omega \notin B\}$ 称事件 A 与事件 B 的差事件。它的含义是：当且仅当事件 A 发生且事件 B 不发生时，事件 $A - B$ 发生。

互斥关系 如果事件 $A \cap B = \emptyset$ 。称事件 A 与事件 B 互斥或不相容。它的含义是：在一次试验后，事件 A 与事件 B 不会同时发生。如果一组事件中任意两个事件互不相容，这组事件两两不相容。

逆事件 事件 $\Omega - A$ 称为事件 A 的逆事件，记作 $\bar{A} = \Omega - A$ 。它的含义是：当且仅当事件 A 不发生时，事件 \bar{A} 发生。于是 $\bar{A} \cap A = \emptyset, \bar{A} + A = \Omega$ 。由于 A 也是 \bar{A} 的对立事件，因此称事件 A 与 \bar{A} 互逆。

事件运算

与集合论中集合的运算一样，事件之间的运算满足下述定律：

- **交换律：** $A \cup B = B \cup A$ $A \cap B = B \cap A$
- **结合律：** $A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$
- **分配律：** $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- **德·摩根法则：** $\overline{A \cap B} = \bar{A} \cup \bar{B}$ $\overline{A \cup B} = \bar{A} \cap \bar{B}$

以上这些定律都可以扩展到任意多个事件。

7.1.2 概率论公理

有事件的定义后，就可以在事件的基础上定义概率。

定义 7.1.1. 设 E 是随机试验, Ω 是它的样本空间. 对于 E 的每一事件 A 赋予一个实数, 记为 $P(A)$, 称为事件 A 的概率, 如果集合函数 $P(\cdot)$ 满足下列条件:

- 非负性 对每一个事件 A , $P(A) \geq 0$.
- 正则性 对必然事件 Ω , $P(\Omega) = 1$.
- 可列可加性 设 A_1, A_2, \dots 是可列个两两互不相容的事件. 即对于 $A_i \cap A_j = \emptyset, i \neq j, i, j = 1, 2, \dots$, 有

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$$

历史上,对概率有两种主要的理解方式,分别是频率派概率(frequentist)和贝叶斯概率(Bayesian probability)。

如果说,抛一枚硬币,硬币正面向上落下的概率为 0.5,对于频率学派,他们认为多次的重复投掷硬币,他们期望正面向上的次数占总实验次数的一半。

而对于贝叶斯学派,他们认为,概率是对事情不确定性的定量描述,与信息有关,而不需要重复试验,因此硬币正面向上概率为 0.5 的解释是:相信下一次试验中,硬币正面向上的可能性为 0.5。两种解释方式各有优劣。

7.1.3 独立事件和条件概率

独立事件

如果连续两次抛一枚均匀的硬币,则两次都出现正面的概率是 $1/2 \times 1/2$,之所以能将二者相乘是因为我们认为这两次抛硬币是独立的,有关独立的正式定义如下:

定义 7.1.2. 如果下式成立,则事件 A 和 B 是独立的:

$$P(AB) = P(A)P(B)$$

如果等式

$$P\left(\bigcap_{i \in J} A_i\right) = \prod_{i \in J} P(A_i)$$

对于所有的 I 的子集 J 都成立,则事件集 $\{A_i : i \in I\}$ 是独立的。

需要注意的是,假定 A 与 B 是互斥事件,并且每个事件都有正的概率,它们可能独立吗?答案是否定的,因为 $P(A)P(B) > 0$ 而 $P(AB) = P() = 0$.除这种情况外,没有别的方法来判断维恩图中集合的独立性。

条件概率

假设 $P(B) > 0$, 定义 B 发生情况下 A 的条件概率如下:

定义 7.1.3. 如果 $P(B) > 0$, 则 A 在 B 下的条件概率为

$$P(A|B) = \frac{P(AB)}{P(B)}$$

从条件概率的定义中可以得到下述引理.

引理 7.1.1. 如果 A 与 B 是相互独立的事件则 $P(AB) = P(A)P(B)$ 。对于任意两个事件 A, B 有

$$P(A|B) = P(A)$$

根据引理, 独立性的另一个解释为: 如果事件 A 和事件 B 相互独立, 那么在知道 B 的情况下不会改变 A 的概率。

7.1.4 贝叶斯理论

在介绍全概率公式前, 引入样本空间划分的定义.

定义 7.1.4. 设 E 是随机试验, Ω 是它的样本空间. S_1, S_2, \dots, S_n , 若:

1) $S_i \cup S_j = \emptyset$, 若 $i \neq j$, 其中 $i, j = 1, 2, \dots, n$

2) $S_1 \cup S_2 \cup, \dots, \cup S_n = \Omega$

则称 S_1, S_2, \dots, S_n 为样本空间 Ω 的一个划分. 若 S_1, S_2, \dots, S_n 为样本空间 Ω 的一个划分. 每次试验 S_1, S_2, \dots, S_n 中必有一个也仅有一个发生.

设试验 E 的样本空间是 Ω , B 是 E 的事件, S_1, S_2, \dots, S_n 为样本空间 Ω 的一个划分, 且 $P(S_i) > 0 (i = 1, 2, \dots, n)$, 则

$$P(B) = P(B|S_1)P(S_1) + P(B|S_2)P(S_2) + \dots + P(B|S_n)P(S_n)$$

称为全概率公式

定义 7.1.5. 贝叶斯定理 令 A_1, \dots, A_k 是 Ω 的一个划分, 对每一个 i 有 $P(A_i) > 0$, 如果 $P(B) > 0$, 则对 $i = 1, 2, \dots, k$ 有

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)}$$

通常称 $P(A_i)$ 为 A 的先验概率, 称 $P(A_i|B)$ 为 A 的后验概率.

在机器学习中, 无论模型如何复杂, 均可以用最基本的加法规则和乘法规则进行概率推理。

$$\text{加法规则} \quad p(x, y) = \sum_y p(x, y)$$

$$\text{乘法规则} \quad p(x, y) = p(x)p(y|x)$$

加法规则与乘法规则本质上就是概率论中的全概率公式与贝叶斯公式。

7.2 随机变量及其分布

统计学与数据挖掘都跟数据有关。怎么将样本空间与事件同数据联系起来呢？这条联系的纽带就是随机变量。

随机变量的定义

定义 7.2.1. 随机变量即映射

$$X : \Omega \rightarrow R$$

该映射对每一个输出 ω 赋予实值 $X(\omega)$ 。

随机变量举例

例 7.2.1. 给定一个查询 Q 和文档集 D ，从文档集中随机抽取一篇文档 d_i ，查看 d_i 是否与查询相关。该试验结果的样本空间是 {不相关, 相关}。构造映射 $X(\omega)$ ：

$$X(\omega) = \begin{cases} 0, & \omega = \text{不相关} \\ 1, & \omega = \text{相关} \end{cases}$$

例 7.2.2. 用随机变量 X 表示股票 A 中午 12 点的股价

$$X(\omega) = x, \omega = \text{股票中午 12 点的股价是 } x \text{ 元}$$

例 7.2.3. 从某部电影的 1000 个影评中抽取 5 个影评，随机变量 X 表示正类影评（对电影持肯定态度）的个数。

$$X(\omega) = \begin{cases} 0, & \omega = \text{没有正类影评} \\ 1, & \omega = 1 \text{ 个正类影评} \\ 2, & \omega = 2 \text{ 个正类影评} \\ 3, & \omega = 3 \text{ 个正类影评} \\ 4, & \omega = 4 \text{ 个正类影评} \\ 5, & \omega = 5 \text{ 个正类影评} \end{cases}$$

累积分布函数

给定随机变量 X ，定义它的累积分布函数（分布函数）如下：

定义 7.2.2. 累积分布函数，或 CDF(Cumulative Distribution Function)，表示函数 $F_X : R \rightarrow [0, 1]$ ，其定义为：

$$F_X(x) = P(X \leq x)$$

CDF 包括了随机变量的所有信息，有时用 F 代替 F_X 来表示 CDF。

概率质量函数和概率密度函数

随机变量分为离散型和连续型随机变量，对于离散型随机变量，我们可以通过概率质量函数刻画随机变量在每个取值的概率；对于连续型随机变量，我们可以利用概率密度函数刻画随机变量的概率密度。

离散型概率质量函数

定义 7.2.3. 如果 X 取可数个值 $\{x_1, x_2, \dots\}$ ，则 X 是离散的，定义 X 的概率函数或概率密度函数为

$$f_X(x) = P(X = x)$$

因此，对于 $x \in R$ 有 $f_X(x) \geq 0$ 并且 $\sum_i f_X(x_i) = 1$. 有时用 f 代替 f_X . X 的累积分布函数 $F_X(x)$ 和 f_X 的关系如下：

$$F_X(x) = P(X \leq x) = \sum_{x_i \leq x} f_X(x_i)$$

连续型概率密度函数

定义 7.2.4. 如果存在某个函数 f_X 对所有 x 有 $f_X(x) \geq 0, \int_{-\infty}^{+\infty} f_X(x)dx = 1$ 并且对任意 $a \leq b$ 有

$$P(a < X < b) = \int_a^b f_X(x)dx,$$

则随机变量 X 是连续型随机变量，函数 f_X 称为概率密度函数(PDF)，且有

$$F_X(x) = \int_{-\infty}^x f_X(t)dt,$$

以及 $f_X(x) = F'_X(x)$ 在 F_X 可微的点均成立.

有时用 $\int f(x)dx$ 或者 $\int f$ 表示 $\int_{-\infty}^{+\infty} f(x)dx$

例 7.2.4. 假设 X 的 PDF 为

$$f_X(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & \text{其他,} \end{cases}$$

显然 $f_X(x) \geq 0$ 且 $\int f_X(x)dx = 1$. 具有这种密度的随机变量称它服从 $(0,1)$ 均分分布. 其含义就是从 0 到 1 之间随机取一点. CDF 为

$$F_X(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x \leq 1 \\ 0, & x > 1 \end{cases}$$

7.2.1 随机变量的常见分布

单变元离散型随机变量

若随机变量 X 为有限个或可列无限多个时, 称 X 为离散型随机变量. 例如??中随机变量 X 只有 3 个取值, 所以是离散型随即变量.

常用的离散型随机变量及其分布

单点分布 仅在一个点 a 上有概率, 记为 $X \sim \delta_a$, 即 $P(X = a) = 1$, 那么

$$F(x) = \begin{cases} 0, & x < a \\ 1, & x \geq a \end{cases}$$

概率密度函数在 $x = a$ 处 $f(x) = 1$, 其他情形下为 0.

离散均匀分布 令 $k > 1$ 为给定的整数, 假设 X 具有如下概率密度函数:

$$f(x) = \begin{cases} \frac{1}{k}, & x = 1, 2, \dots, k. \\ 0, & \text{其他.} \end{cases}$$

则称 X 在 $\{1, 2, \dots, k\}$ 上服从均匀分布.

伯努利分布 随机变量 X 只取两个值, 一般用 0, 1 表示, 且概率密度函数为:

$$f(x) = \begin{cases} p, & x = 1, \\ 1 - p, & x = 0. \end{cases} \quad (7.1)$$

其中 $p \in [0, 1]$, 称 X 服从伯努利分布, 记为 $X \sim Bernoulli(p)$, 概率密度函数可简写为:

$$f(x) = p^x(1 - p)^{1-x}$$

其中 $x \in [0, 1]$. 在统计机器学习中的逻辑回归分类模型假设数据服从伯努利分布, 进而对数据进行建模.

二项式分布 假设从若干影视评论中有放回的抽取 n 次, 令随机变量 X 表示抽取到正类影评的次数. 假设每次取影评都是独立的且取到正类影评的概率是 p . 概率密度函数:

$$f(x) = \begin{cases} \binom{n}{x} p^x (1-p)^{n-x}, & x = 0, 1, \dots, n \\ 0, & \text{其他.} \end{cases}$$

具有上述概率密度函数的随机变量称为伯努利随机变量, 记 $X \sim Binomial(n, p)$.

几何分布 如果在二项式分布的示例中, 不是有放回的取 n 次, 而是直到取到正类影评为止, 令随机变量 X 表示第一次取得正类影评的次数, 则 X 的密度函数为:

$$P(X = k) = p(1 - p)^{k-1}, k = 0, 1, 3, \dots.$$

则 X 服从参数为 $p \in (0, 1)$ 的几何分布, 记为 $X \sim Geom(p)$. 对于几何分布有:

$$\sum_{k=0}^{+\infty} P(X = k) = p \sum_{k=0}^{+\infty} (1 - p)^k = \frac{p}{p} = 1$$

泊松分布 如果

$$f(x=k) = e^{-\lambda} \frac{\lambda^x}{x!}, \quad x \geq 0,$$

则随机变量 X 服从参数为 λ 的泊松分布, 记为 $X \sim Poisson(\lambda)$. 并且有:

$$\sum_{x=0}^{+\infty} f(x) = e^{-\lambda} \sum_{x=0}^{+\infty} \frac{\lambda^x}{x!} = e^{-\lambda} e^\lambda = 1$$

泊松分布常用于罕见事件的计数, 如放射性元素的衰变与交通事故.

常用的连续型随机变量及其分布

单变元均匀分布 设 X 的概率密度函数为:

$$f_X(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & \text{其他} \end{cases}$$

显然 $f_X(x) \geq 0$ 且 $\int_{-\infty}^{\infty} f(x) = 1$. 称具有这种概率密度函数的随机变量 X 服从 $(0, 1)$ 均匀分布.

单变元正态(高斯)分布 如果随机变量 X 的概率密度函数是:

$$f(x) = \frac{1}{\delta \sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma^2} (x - \mu)^2 \right\}$$

则 X 服从参数为 μ 和 σ 的正态分布, 记为 $X \sim N(\mu, \sigma^2)$, 其中 $\mu \in R, \sigma > 0$.

正态分布在概率和统计中扮演者重要角色, 许多自然现象可以用正态分布来近似, 如男性身高. 在深度学习中, 正态分布也是常用的参数初始化方法

拉普拉斯分布 如果随机变量 X 的概率密度函数是:

$$f(x) = \frac{1}{2\lambda} e^{-\frac{|x-\mu|}{\lambda}}$$

其中 $\lambda > 0, \mu$ 为常数, 则 X 服从拉普拉斯分布.

7.2.2 多维随机变量及其分布函数

在实际生产与理论研究中, 都常常会遇到这种情况: 需要同时用几个随机变量才能较好地描绘某一试验或现象. 例如买西瓜时, 根据(色泽, 敲声, 根蒂)来挑选西瓜. 航天飞船返航时的位置需要用(经度, 纬度)来确定.

在数据科学中使用多维随机变量来描述一个数据样本. 例如在金融反欺诈领域中要决定是否给一人贷款时, 要观察此人的(收入, 年龄, 是否结婚, 学历)等等. 称 n 个随机变量 x_1, x_2, \dots, x_n 的总体 $X = (x_1, x_2, \dots, x_n)$ 为 n 元随机变量(或 n 维随机变量). 本节重点讨论二维的情形, n 维情况类似.

边缘分布

定义 7.2.5. 如果 (X, Y) 具有联合质量函数 $f_{X,Y}$, 则 X 的边缘概率质量函数定义为:

$$f_X(x) = P(X = x) = \sum_y P(X = x, Y = y) = \sum_y f(x, y)$$

Y 的边缘概率质量函数定义为:

$$f_Y(y) = P(Y = y) = \sum_x P(X = x, Y = y) = \sum_x f(x, y)$$

定义 7.2.6. 对于连续型随机变量, 边际概率密度函数为

$$f_X(x) = \int_{-\infty}^{+\infty} f(x, y) dy$$

$$f_Y(y) = \int_{-\infty}^{+\infty} f(x, y) dx$$

相应的边际分布函数记为 F_X 和 F_Y

独立的随机变量

定义 7.2.7. 如果对于任意 A 和 B 有

$$P(X \in A, Y \in B) = P(X \in A)P(Y \in B)$$

称随机变量 X 和 Y 是独立的

原则上, 为检验两个随机变量 X 和 Y 是否独立, 需要对所有子集 A 和 B 验证等式 $P(X \in A, Y \in B) = P(X \in A)P(Y \in B)$. 值得庆幸的是, 对于连续型随机变量有如下结论. 事实上, 该结论对离散随机变量也是成立的.

定理 7.2.1. 令 X 和 Y 具有联合概率质量函数或联合概率密度函数 $f_{X,Y}$, 则 X 与 Y 相互独立当前仅当 $f_{X,Y}(x, y) = f_X(x)f_Y(y)$ 对于所有 x 和 y 成立.

条件分布

如果 X 和 Y 是离散的, 则可以计算假设已观察到 $Y = y$ 情况下 X 的条件分布. 特别地, $P(X = x|Y = y) = P(X = x, Y = y)/P(Y = y)$. 从而如下条件概率密度函数的定义:

定义 7.2.8. 如果 $f_Y(y) > 0$, 则条件概率密度函数为

$$f_{X|Y}(x|y) = P(X = x|Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)} = \frac{f_{X,Y}(x, y)}{f_Y(y)}$$

离散的情形下, $f_{X|Y}(x|y)$ 表示 $P(X = x|Y = y)$. 对于连续型随机变量, 采用相同的概念, 当解释不通, 必须通过积分求得概率.

定义 7.2.9. 对于连续情形, 假设 $f_Y(y) > 0$, 则条件概率密度函数为

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

从而

$$P(X \in A|Y = y) = \int_A f_{X|Y}(x|y) dx$$

条件分布在机器学习中的应用 在机器学习中, 监督学习的任务就是学习一个模型, 应用这一模型, 对给定的输入预测相应的输出. 这个模型的一般形式为决策函数:

$$Y = f(X)$$

或者条件概率分布:

$$P(Y|X)$$

监督学习方法又可以分为生成方法 (generative approach) 和判别方法 (discriminative approach). 所学到的模型分别称为生成模型(generative model) 和判别模型(discriminative model).

生成方法由数据学习联合概率分布 $P(X, Y)$, 然后求出条件概率分布 $P(X|Y)$ 作为预测的模型, 即生成模型:

$$P(Y|X) = \frac{P(X, Y)}{P(X)}$$

这样的方法之所以称为生成方法, 是因为模型表示了给定输入 X 产生输出 Y 的生成关系. 典型的生成模型有: 朴素贝叶斯法, 隐马尔可夫模型.

判别方法由数据直接学习决策函数 $f(X)$ 或者条件概率分布 $P(Y|X)$ 作为预测的模型, 即为判别模型. 判别方法关系的是对于给定的输入 X , 应该预测什么样的输出 Y . 典型的判别模型包括:k 近邻法、感知机、决策树、逻辑回归模型、最大熵模型、支持向量机、提升方法和条件随机场等.

在监督学习中, 生成方法和判别方法各有优缺点, 适合于不同条件下的学习问题.

生成方法的特点: 生成方法可以还原出联合概率分布 $P(X, Y)$, 而判别方法则不能; 生成方法的学习收敛速度更快, 即当样本容量增加时, 学到的模型可以更快地收敛于真是模型; 当存在隐变量时, 仍可以用生成方法, 此时判别方法就不能用.

判别方法的特点: 判别方法直接学习的是条件概率 $P(Y|X)$ 或者决策函数 $f(X)$, 直接面对预测, 往往学习的准确率更高; 由于直接学习 $P(Y|X)$, 可以对数据进行各种程度上的抽象、定义特征并使用特征, 因此可以建华学习问题.

独立同分布样本

令 $X = (X_1, \dots, X_n)$, 其中 X_1, \dots, X_n 为随机变量, 则称 X 为随机变量. 令 $f(x_1, \dots, x_n)$ 表示概率密度函数, 同二维情形一样, 可以定义边际分布, 条件分布等.

称 X_1, \dots, X_n 是独立的, 如果对于任意集合 A_1, \dots, A_n 有:

$$P(X_1 \in A_1, \dots, X_n \in A_n) = \prod_{i=1}^n P(X_i \in A_i)$$

容易验证 $f(x_1, \dots, x_n) = \prod_{i=1}^n f_{X_i}(x_i)$ 成立

定义 7.2.10. 如果 X_1, \dots, X_n 独立并且都有相同的累计分布函数 F , 则称 X_1, \dots, X_n 是独立同分布的 (IID), 记为

$$X_1, \dots, X_n \sim F$$

如果 F 的密度函数为 f , 也可记为 $X_1, \dots, X_n \sim f$, 有时也称 X_1, \dots, X_n 是来自 F , 样本量为 n 的随机样本.

许多统计理论和事件都建立在 IID 观测的基础上, 当讨论统计量的时候将对它作详细研究.

7.3 随机变量的数字特征

随机变量的分布函数描述随机变量的统计规律. 但实际中并不知道随机变量的真实分布. 有时只关心随机变量的某些方面的数字特征—期望, 方差, 协方差.

期望 $E(X)$ 反映了随机变量的平均取值水平. 比如一个班级的平均成绩, 某地区人的平均身高等. 在机器学习任务中, 我们的目标是学习到使损失函数 (记为 $L(Y, f(X))$) 的期望最小的模型 f . $R_{exp}(f) = E[L(Y, f(X))]$ 称为期望损失. 我们利用模型在训练集上的期望损失学习模型; 利用模型在测试集上的期望损失衡量模型的泛化性能.

方差 $D(X)$ 反映随机变量与期望的平均偏离程度. 在机器学习中方差常用来度量同样大小的训练集的变动所导致模型性能的变化幅度, 是模型性能的评价指标之一.

协方差 $Cov(X, Y)$ 反映两个随即变量 X, Y 的整体误差. 如果两个变量的变化趋势一致, 也就是说如果当 X 大于自身的 $E(X)$ 且 Y 大于自身的 $E(Y)$ 时, 那么两个变量之间的协方差就是正值; 如果两个变量的变化趋势相反, 那么两个变量之间的协方差就是负值. 通过协方差可以计算相关系数, 而相关系数反映了不同特征间的线性相关程度. 也可作为特征选择的一个指标.

7.3.1 期望

定义

随机变量 X 的均值或者期望表示 X 的平均值, 其定义如下:

定义 7.3.1. 随机变量 X 的期望值或均值或一阶矩定义为

$$E(X) = \int_{-\infty}^{+\infty} x dF(x) = \begin{cases} \sum_x x f(x), & X \text{ 为离散型随机变量}, \\ \int_x x f(x) dx, & X \text{ 为连续型随机变量}. \end{cases}$$

其中 $F(x)$ 是累计分布函数. 加入以上求和(或积分)定义明确, 也可使用如下符号表示 X 的期望:

$$E(X) = EX = \int_{-\infty}^{+\infty} x dF(x) = u = u_X$$

期望是分布的单值概括, 可以将 $E(X)$ 看成是 IID 随机样本 X_1, \dots, X_n 的平均 $\sum_{i=1}^n X_i/n$. 事实上, $E(X) \approx \sum_{i=1}^n X_i/n$ 是正确的而不是主观的推断, 这点将在大数定律章节详细说明.

符号 $\int x dF(x)$ 仅仅用来统一符号, 而不用将离散形式写成 $\sum_x x f(x)$, 将连续形式写成 $\int x f(x) dx$.

为保证 $E(X)$ 定义明确, 如果 $\int_x |x| dF_X(x) < \infty$, 则称 $E(X)$ 存在. 否则称期望不存在.

实际上, 我们经常需要求随机变量函数的数学期望, 例如飞机机翼受到压力 $W = kV^2$ (V 是风速, $k > 0$ 是常数) 的作用, 需要求 W 的数学期望, 这里 W 是随机变量 V 的函数. 这时, 可以通过下面的定理来求 W 的数学期望.

定理 7.3.1. 设 Y 是随机变量 X 的函数: $Y = r(X)$. 则

$$E(Y) = E(r(X)) = \int r(x) dF_X(x)$$

当我们求 $E(r(X))$ 时, 不必算出 $r(X)$ 的概率密度函数, 而只需利用 X 的概率密度函数就可以了.

性质

根据期望的定义, 我们可以得出许多有关期望的性质.

性质 7.3.1. 设 C 是常数, 则有 $E(C) = C$

性质 7.3.2. 设 X 是随机变量, C 是常数, 则有

$$E(CX) = CE(X)$$

性质 7.3.3. 设 X 和 Y 是两个随机变量, 则有:

$$E(X + Y) = E(X) + E(Y)$$

性质 7.3.4. 设 X 和 Y 是相互独立的两个随机变量, 则有:

$$E(XY) = E(X)E(Y)$$

上述性质也可以扩张到 n 个独立的随机变量. 若 X_1, X_2, \dots, X_n 为独立的随机变量, 则

$$E\left(\prod_{i=1}^n X_i\right) = \prod E(X_i)$$

应用举例

期望在机器学习中随处可见。比如在衡量分类模型准确率时我们使用期望来定义：

$$\text{accuracy} = E(I(y = \hat{f}(x))) = \sum I(y = \hat{f}(x))dF(x)$$

其中 $F(x)$ 是随机变量 x 的累积分布函数， $\hat{f}(x)$ 是训练好的模型，且：

$$I(y = \hat{f}(x)) = \begin{cases} 1 & \text{如果 } y = \hat{f}(x) \\ 0 & \text{如果 } y \neq \hat{f}(x) \end{cases}$$

但在实际中，我们并不知道数据真实的概率分布函数，常取的做法是采集 n 个未在训练集中出现过的样本作为测试集，使用：

$$\sum_i^n \frac{1}{n} I(y_i = \hat{f}(x_i))$$

来衡量模型的准确率。

此外强化学习中价值函数（状态价值函数和动作价值函数）也是由数学期望来定义的。

7.3.2 方差

定义 7.3.2. 令随机变量 X 的均值为 μ , X 的方差记为 σ^2 或 $V(X)$ 或 VX , 定义为

$$\begin{aligned} \sigma^2 &= E(X - \mu)^2 \\ &= \int (x - \mu)^2 dF(X) \\ &= \begin{cases} \sum_x (x - \mu)^2 f(x), & X \text{ 为离散型随机变量,} \\ \int_x (x - \mu)^2 f(x) dx, & X \text{ 为连续型随机变量.} \end{cases} \end{aligned}$$

其中假设期望存在. 标准差定义为 $sd(X) = \sqrt{V(X)}$ 也记为 σ 或 σ_X

例 7.3.1. 设随机变量 X 具有数学期望 $E(X) = \mu$, 方差 $D(X) = \sigma^2 \neq 0$, 记

$$X^* = \frac{X - \mu}{\sigma}$$

则

$$E(X^*) = \frac{1}{\sigma} E(X - \mu) = \frac{1}{\sigma} [E(X) - \mu] = 0$$

$$D(X^*) = E((X^*)^2) = \int \left(\frac{x - \mu}{\sigma}\right)^2 f(x) dx = \frac{1}{\sigma^2} \int (x - \mu)^2 f(x) dx = 1$$

因此对于任何一个具有均值和方差的分布，我们总可以通过这样的变换将其变为均值为 0，方差为 1 的分布。

方差的性质

性质 7.3.5. 设 X 是随机变量, 有

$$D(X) = E(X^2) - [E(X)]^2$$

性质 7.3.6. 设 C 是常数, 则有 $D(C) = 0$

性质 7.3.7. 设 X 是随机变量, C 是常数, 则有

$$D(CX) = C^2 D(X) \quad D(X + C) = D(X)$$

性质 7.3.8. 设 X 和 Y 是两个随机变量, 则有:

$$D(X + Y) = D(X) + D(Y) + 2E\{(X - E(X))(Y - E(Y))\}$$

若 X 与 Y 相互独立, 则有:

$$D(X + Y) = D(X) + D(Y)$$

此性质也可扩展到 n 个随机变量的情形. 假设 X_1, X_2, \dots, X_n 是随机变量, a_1, a_2, \dots, a_n 是常数, 则

$$V\left(\sum_{i=1}^n a_i X_i\right) = \sum_{i=1}^n a_i^2 V(X_i)$$

如果 X_1, X_2, \dots, X_n 是随机变量, 则定义样本均值为

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$$

样本方差为

$$S_n^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2$$

定理 7.3.2. 令 X_1, X_2, \dots, X_n 是独立同分布的随机变量且 $\mu = E(X_i), \sigma^2 = V(X_i)$, 则

$$E(\bar{X}_n) = \mu, \quad V(\bar{X}_n) = \frac{\sigma^2}{n}, \quad E(S_n^2) = \sigma^2$$

7.3.3 一些重要的随机变量的期望和方差

下面介绍的随机变量的期望与方差

伯努利分布的期望与方差 设随机变量 X 服从参数为 p 的伯努利分布:

X	1	0
P(X)	p	1-p

则 X 的期望:

$$E(X) = 1 * p + 0 * (1 - p) = p$$

X 的方差:

$$D(X) = (1-p)^2 * p + (0-p)^2 * (1-p) = p(1-p)$$

二项分布的方差 设随机变量 X 服从参数为 n, p 的二项式分布:

X	0	1	2	...	n
P(X)	$C_n^0 p^0 (1-p)^{n-0}$	$C_n^1 p^1 (1-p)^{n-1}$	$C_n^2 p^2 (1-p)^{n-2}$...	$C_n^n p^n (1-p)^0$

则随机变量 X 的期望

$$E(X) = \sum_{k=0}^n k \times P(X=k) = \sum_{k=0}^n k \times C_n^k p^k (1-p)^{n-k} = np$$

方差 $D(X)$:

$$\begin{aligned} D(X) &= \sum_{k=0}^n (k - E(X))^2 \times P(X=k) \\ &= \sum_{k=0}^n (k - np)^2 \times C_n^k p^k (1-p)^{n-k} \\ &= np(1-p) \end{aligned}$$

几何分布的方差 设随机变量 X 服从参数为 p 的几何分布, $0 < p < 1$:

X	1	2	3	...	n	...
P(X)	p	$(1-p)p$	$(1-p)^2 p$...	$(1-p)^{n-1} p$...

即:

$$P(X=k) = (1-p)^{k-1} p$$

随机变量 X 的期望:

$$E(X) = \sum_{k=1}^n k \times P(X=k) = \sum_{k=1}^n k \times (1-p)^{k-1} p = \frac{1}{p}$$

方差 $D(X)$:

$$D(X) = \sum_{k=1}^n (k - E(X))^2 \times P(X=k) = \sum_{k=1}^n (k - \frac{1}{p})^2 \times (1-p)^{k-1} p = \frac{1-p}{p^2}$$

泊松分布的方差 随机变量 X 服从参数为 λ 的泊松分布:

$$F(x=k) = e^{-\lambda} \frac{\lambda^k}{k!}, \quad k = 0, 1, 2, \dots,$$

则随机变量 X 的期望:

$$E(X) = \sum_{k=0}^{\infty} k e^{-\lambda} \frac{\lambda^k}{k!} = \lambda$$

方差:

$$D(X) = \lambda$$

均匀分布的方差 设随机变量 X 服从均匀分布, 记为 $X \sim U(a \square b)$, 其概率密度函数:

$$f(x) = \begin{cases} 0, & X < a \\ \frac{1}{b-a}, & a \leq x \leq b \\ 0, & x > b \end{cases}$$

则随机变量 X 的期望 $E(X)$:

$$E(X) = \int_{-\infty}^{+\infty} xf(x)dx = \int_a^b x \frac{1}{b-a} dx = \frac{b+a}{2}$$

方差 $D(X)$:

$$D(X) = \int_{-\infty}^{+\infty} (x - E(X))^2 f(x)dx = \int_a^b (x - \frac{b+a}{2})^2 \frac{1}{b-a} dx = \frac{(b-a)^2}{12}$$

Laplace 分布的方差 设随机变量 X 服从 Laplace 分布, 其密度函数如下:

$$\text{Laplace}(x; \mu, \gamma) = \frac{1}{2\gamma} \exp(-\frac{|x-\mu|}{\gamma})$$

则 X 的期望 $E(X)$:

$$E(X) = \int_{-\infty}^{+\infty} xf(x)dx = \int_{-\infty}^{+\infty} x \frac{1}{2\gamma} \exp(-\frac{|x-\mu|}{\gamma}) dx = \mu$$

方差 $D(X)$:

$$D(X) = \int_{-\infty}^{+\infty} (x - E(X))^2 f(x)dx = \int_{-\infty}^{+\infty} (x - \mu)^2 \frac{1}{2\gamma} \exp(-\frac{|x-\mu|}{\gamma}) dx = 2\gamma^2$$

高斯分布的方差 设 X 服从参数为 μ 和 σ 的高斯分布, $X \sim N(x; \mu, \sigma)$:

$$N(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

则随机变量的期望 $E(X)$:

$$E(X) = \int_{-\infty}^{+\infty} xf(x)dx = \int_{-\infty}^{+\infty} x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \mu$$

方差 $D(X)$:

$$D(X) = \int_{-\infty}^{+\infty} (x - E(X))^2 f(x)dx = \int_{-\infty}^{+\infty} (x - \mu)^2 \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \sigma^2$$

7.3.4 协方差和相关系数

对于二维随机变量 (X, Y) , 除了讨论期望与方差之外, 还需讨论 X 和 Y 之间的相关关系的数字特征.

定义 7.3.3. 令 X 和 Y 是均值分别为 μ_X 和 μ_Y , 标准差分别是 σ_X 和 σ_Y 的随机变量, 定义 X 和 Y 的协方差:

$$\text{Cov}(X, Y) = E\{(X - \mu_X)(Y - \mu_Y)\}$$

相关系数:

$$\rho_{XY} = \frac{\text{Cov}(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

由协防差定义可知

$$\text{Cov}(X, Y) = \text{Cov}(Y, X) \quad \text{Cov}(X, X) = D(X)$$

以及对于任意两个随机变量 X 和 Y , 下列等式成立:

$$D(X + Y) = D(X) + D(Y) + 2\text{Cov}(X, Y)$$

将 $\text{Cov}(X, Y)$ 的定义展开, 易得

$$\text{Cov}(X, Y) = E(XY) - E(X)E(Y)$$

我们常常利用这一式子得出以下协方差性质:

性质 7.3.9.

$$\text{Cov}(aX, bY) = ab\text{Cov}(X, Y)$$

性质 7.3.10.

$$\text{Cov}(X_1 + X_2, Y) = \text{Cov}(X_1, Y) + \text{Cov}(X_2, Y)$$

同时相关系数具有以下性质:

性质 7.3.11.

$$|\rho_{XY}| \leq 1$$

当 $\rho = 0$ 时, 称随机变量 X 和 Y 不相关.

性质 7.3.12. $|\rho_{XY}| = 1$ 的充要条件是存在 a, b 使 $P(Y = a + bX) = 1$

例 7.3.2. 设 (X, Y) 服从二维正态分布, 它的概率密度函数

$$f(x, y) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\frac{(x-\mu_1)^2}{\sigma_1^2} - 2\rho\frac{(x-\mu_1)(y-\mu_2)}{\sigma_1\sigma_2} + \frac{(y-\mu_2)^2}{\sigma_2^2}\right]\right\}$$

我们可以分别计算出 X 和 Y 的边缘概率分布, 然后分布求出 X 和 Y 的期望, 方差以及相关系数. 由于计算太复杂, 这直接给出结果: $E(X) = \mu_1$, $E(Y) = \mu_2$, $D(X) = \sigma_1^2$, $D(Y) = \sigma_2^2$, $\text{Cov}(X, Y) = \rho$

这也就是说, 二维正态随机变量 (X, Y) 的概率密度中的参数 ρ 就是 X 和 Y 的相关系数, 因而二维正态随机变量的分布完全可由 X, Y 各自的数学期望, 方差以及它们的相关系数所确定.

定理 7.3.3. 若 (X, Y) 服从二维正态分布, 那么 X 和 Y 相互独立的充要条件是 $\rho = 0$

7.3.5 矩和协方差矩阵

本节先介绍随机变量的另外几个数字特征. 设 (X, Y) 是二维随机变量.

定义 7.3.4. 设 X 和 Y 是随机变量, 若

$$E(X^k), \quad k = 1, 2, \dots$$

存在, 称它为 X 的 k 阶原点矩, 简称 k 阶矩.

若

$$E\{[X - E(X)]^k\}, k = 2, 3, \dots$$

存在, 称它为 X 的 k 阶中心矩.

若

$$E\{X^k Y^l\}, k, l = 1, 2, 3, \dots$$

存在, 称它为 X 和 Y 的 $k+l$ 阶混合矩.

若

$$E\{[X - E(X)]^k [Y - E(Y)]^l\}, k, l = 1, 2, 3, \dots$$

存在, 称它为 X 和 Y 的 $(k+l)$ 阶混合中心矩.

显然 X 的数学期望 $E(X)$ 是 X 的一阶原点矩, 方差 $D(X)$ 是 X 的二阶中心矩, 协方差 $Cov(X, Y)$ 是 X 和 Y 的二阶混合中心矩.

下面介绍 n 维随机变量的协方差矩阵. 先从二维随机变量讲起.

二维随机变量 (X_1, X_2) 有 4 个二阶中心矩(假设它们都存在), 分别记为

$$c_{11} = E\{[X_1 - E(X_1)]^2\}$$

$$c_{12} = E\{[X_1 - E(X_1)][X_2 - E(X_2)]\}$$

$$c_{21} = E\{[X_2 - E(X_2)][X_1 - E(X_1)]\}$$

$$c_{22} = E\{[X_2 - E(X_2)]^2\}$$

将它们排成矩阵的形式

$$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

这个矩阵称为随机变量 (X, Y) 的协方差矩阵.

设 n 维随机变量 (X_1, X_2, \dots, X_n) 的二阶混合中心矩

$$c_{ij} = Cov(X_i, Y_j) = E[X_i - E(X_i)][X_j - E(X_j)], i, j = 1, 2, \dots, n$$

都存在，则称矩阵：

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}$$

为 n 维随机变量 (X_1, X_2, \dots, X_n) 的协方差矩阵。由于 $c_{ij} = c_{ji}$ ($i \neq j; i, j = 1, 2, \dots, n$)，因此上述矩阵是一个对称矩阵。

一般情况下 n 维随机变量的分布是不知道的，或者太过复杂，以致在数学上不易处理，因此在实际应用中协方差矩阵就显得重要了。我们以 n 维正态分布为例来介绍 n 维随机变量。在介绍 n 维正态分布的概率密度函数之前，我们先将二维正态分布的概率密度函数改成另外一种形式，以便将它推广到 n 维随机变量的场合中去。二维正态随机变量 (X_1, X_2) 的概率密度函数为

$$f(x, y) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp \left\{ \frac{-1}{2(1-\rho^2)} \left[\frac{(x-\mu_1)^2}{\sigma_1^2} - 2\rho \frac{(x-\mu_1)(y-\mu_2)}{\sigma_1\sigma_2} + \frac{(y-\mu_2)^2}{\sigma_2^2} \right] \right\}$$

现将上式中花括号内的式子写成矩阵形式，为此引入下面的列矩阵

$$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

(X_1, X_2) 的协方差矩阵为

$$C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

它的行列式 $\det C = \sigma_1^2\sigma_2^2(1-\rho^2)$ ， C 的逆矩阵为

$$C^{-1} = \frac{1}{\det C} \begin{pmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{pmatrix}$$

经计算可知（这里矩阵 $(X - \mu)^T$ 是 $(X - \mu)$ 的转置矩阵）

$$(X - \mu)^T C^{-1} (X - \mu) = \frac{1}{\det C} (x_1 - u_1 x_2 - u_2) \begin{bmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}$$

$$(X - \mu)^T C^{-1} (X - \mu) = \frac{1}{1-\rho^2} \left[\frac{(x-\mu_1)^2}{\sigma_1^2} - 2\rho \frac{(x-\mu_1)(y-\mu_2)}{\sigma_1\sigma_2} + \frac{(y-\mu_2)^2}{\sigma_2^2} \right]$$

于是 (X_1, X_2) 的概率密函可写成

$$f(x_1, x_2) = \frac{1}{(2\pi)^{2/2}(\det C)^{1/2}} \exp \left\{ -\frac{1}{2} (X - \mu)^T C^{-1} (X - \mu) \right\}$$

上式容易推广到 n 维正态随机变量 (X_1, X_2, \dots, X_n) 的情况。引入列矩阵

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} E(X_1) \\ E(X_2) \\ \vdots \\ E(X_n) \end{pmatrix}$$

草稿
请勿外传

n 维正态随机变量 (X_1, X_2, \dots, X_n) 的概率密度函数定义为:

$$f(x_1, x_2, \dots, x_n) = \frac{1}{(2\pi)^{n/2}(\det C)^{1/2}} \exp \left\{ -\frac{1}{2}(X - \mu)^T C^{-1}(X - \mu) \right\}$$

其中 C 是 (X_1, X_2, \dots, X_n) 的协方差矩阵.

n 维正态随机变量具有以下四条重要的性质 (证略):

1. n 维正态随机变量 (X_1, X_2, \dots, X_n) 的每一个分量 $X_i, i = 1, 2, \dots, n$ 都是 n 维正态随机变量; 反之, 若 X_1, X_2, \dots, X_n 都是正态随机变量, 且相互独立, 则 X_1, X_2, \dots, X_n 是 n 维正态随机变量.

2. n 维随机变量 X_1, X_2, \dots, X_n 服从 n 维正态分布的充要条件是 X_1, X_2, \dots, X_n 的任意线性组合

$$l_1 X_1 + l_2 X_2 + \dots + l_n X_n$$

服从一维正态分布 (其中 l_1, l_2, \dots, l_n 不全为 0)

3. 若 X_1, X_2, \dots, X_n 服从 n 维正态分布, 设 Y_1, Y_2, \dots, Y_k 是 $X_j (j = 1, 2, \dots, n)$ 的线性函数, 则 (Y_1, Y_2, \dots, Y_k) 也服从多维正态分布.

这一性质称为正态变量的线性变换不变性.

4. 设 (X_1, X_2, \dots, X_n) 服从 n 维正态分布, 则“ X_1, X_2, \dots, X_n ”相互独立与“ X_1, X_2, \dots, X_n ”两两不相关是等价的.

协方差性质: 正定和半正定

本节以二元离散随机变量为例, 来介绍协方差矩阵的半正定性质, 对于 n 元随即变量类似. 对于二元随机变量 (x, y) , 其协方差矩阵为:

$$V = \sum_{i,j} p_{ij} V_{i,j} = p_{ij} \begin{bmatrix} (x_i - \mu_x)^2 & (x_i - \mu_x)(y_j - \mu_y) \\ (x_i - \mu_x)(y_j - \mu_y) & (y_j - \mu_y)^2 \end{bmatrix}$$

其中 $p_{ij} = F(X = i, Y = j)$ 的概率. 注意矩阵 $V_{i,j}$:

$$V_{i,j} = \begin{bmatrix} x_i - \mu_x \\ y_j - \mu_y \end{bmatrix} \begin{bmatrix} x_i - \mu_x & x_i - \mu_x \end{bmatrix} = U^T U$$

由于 $p_{ij} > 0$ 且 $V_{i,j}$ 是秩为 1 的半正定矩阵, 故 $p_{ij} V_{i,j}$ 也是半正定矩阵. 根据半正定矩阵之和仍是半正定举证的性质, 那么 V 是半正定矩阵.

注意: 这仅证明了离散型随机变量的协方差矩阵是半正定了, 对于连续型随机变量的协方差矩阵半正定的性质见 7.3.5 节.

协方差矩阵的分解

目前可以对于二元离散型随机变量 (x, y) , 其协方差矩阵等于多个矩阵之和, 即:

$$V = \sum_{i,j} p_{ij} V_{i,j} = \sum_{i,j} p_{ij} U U^T = \sum_{i,j} p_{ij} \begin{bmatrix} x - \mu_x \\ y - \mu_y \end{bmatrix} \begin{bmatrix} x - \mu_x & y - \mu_y \end{bmatrix}$$

其中 p_{ij} 是二元离散随机变量的概率分布函数, μ_x 和 μ_y 分别是 x 和 y 的均值, 且:

$$U = \begin{bmatrix} x - \mu_x \\ y - \mu_y \end{bmatrix}$$

对于连续型随机变量, 我们使用概率密度函数以及积分来对协方差矩阵 V 进行分解:

$$V = \int_x \int_y f(x, y) U U^T dxdy$$

其中 $f(x, y)$ 是二元连续型随机变量的概率密度函数, μ_x 和 μ_y 分别是 x 和 y 的均值:

$$U = \begin{bmatrix} x - \mu_x \\ y - \mu_y \end{bmatrix}$$

这里借助协方差矩阵分解来给出协方差矩阵是半正定的另一种证明: 对于随机变量:

$$z = \mathbf{c}^T \mathbf{x} = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

其中 $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$, 其方差:

$$\begin{aligned} V(z) &= E [(\mathbf{c}^T \mathbf{x} - \mathbf{c}^T \bar{\mathbf{x}})(\mathbf{c}^T \mathbf{x} - \mathbf{c}^T \bar{\mathbf{x}})^T] \\ &= \mathbf{c}^T E [(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T] \mathbf{c} \\ &= \mathbf{c}^T V(\mathbf{x}) \mathbf{c} \end{aligned}$$

故随机变量 z 的方差可以有 \mathbf{x} 的协方差矩阵以及相关系数 \mathbf{c} 得到, 并且由于随机变量的方差恒大于等于 0, 故 $\mathbf{c}^T V(\mathbf{x}) \mathbf{c} \geq 0$, 故 V 是半正定的.

若此时

$$\mathbf{c} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

且 $\mathbf{z} = \mathbf{c}\mathbf{x}$

$$\mathbf{z} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

则 \mathbf{z} 的协方差矩阵:

$$V(\mathbf{z}) = A V(\mathbf{x}) A^T$$

7.3.6 条件期望

假设 X 和 Y 为随机变量, 当 $Y = y$ 时 X 的均值是多少? 方法跟前面计算 X 的均值一样, 只不过将期望定义中的 $f_X(x)$ 用 $f_{X|Y}(x|y)$ 代替就可以了.

定义 7.3.5. 给定 $Y = y$ 情况下 X 的条件期望为

$$E(X|Y = y) = \begin{cases} \sum x f_{X|Y}(x|y), & \text{离散情形,} \\ \int x f_{X|Y}(x|y) dx, & \text{连续情形} \end{cases}$$

如果 $r(x, y)$ 为 x, y 的函数, 则

$$E(r(X, Y)|Y = y) = \begin{cases} \sum r(x, y) f_{X|Y}(x|y), & \text{离散情形,} \\ \int r(x, y) f_{X|Y}(x|y) dx, & \text{连续情形} \end{cases}$$

注意! 条件期望与期望有一些区别, 期望 $E(X)$ 是一个值, 而 $E(X|Y = y)$ 是 y 的函数. 在观察 y 之前, 并不知道 $E(X|Y = y)$ 的值, 所以它是一个随机变量, 记为 $E(X|Y)$. 换句话说, $E(X|Y)$ 是随机变量, 当 $Y = y$ 时, 其值为 $E(X|Y = y)$. 类似的, $E(r(X, Y)|Y = y)$ 是随机变量, 当 $Y = y$ 时, 其值为 $E(r(X, Y)|Y = y)$. 这一点很容易引起混淆, 下面举一个例子来说明.

例 7.3.3. 假设 $X \sim Uniform(0, 1)$, 当观察到 $X = x$ 后, 假设 $Y|X = x \sim Uniform(x, 1)$, 凭直觉 $E(Y|X = x) = (1+x)/2$, 事实上 $f_{Y|X}(y|x=1) = 1/(1-x)$, 其中 $x < y < 1$, 故

$$E(Y|X = x) = \int_0^1 y f_{Y|X}(y|x) dy = \frac{1}{1-x} \int_x^1 y dy = \frac{1+x}{2}$$

因此 $E(Y|X) = (1+X)/2$, 它是一个随机变量. 当观察到 $X = x$ 后, 其值为 $E(Y|X = x) = (1+x)/2$

定理 7.3.4. (期望迭代法则) 对随机变量 X 和 Y , 假设期望均存在, 则有

$$E[E(Y|x)] = E(Y), \quad E[E(X|Y)] = E(X)$$

更一般的, 对任意函数 $r(x, y)$ 有

$$E[E(r(X, Y))|X] = E(r(X, Y))$$

证明. 下面证明第一个等式, 利用条件期望的定义和 $f(x, y) = f(x)f(y|x)$

$$\begin{aligned} E[E(Y|X)] &= \int E(Y|X=x) f_X(x) dx = \int \int y f(y|x) dy f(x) dx \\ &= \int \int y f(y|x) f(x) dx dy = \int \int y f(x, y) dx dy = E(Y) \end{aligned}$$

□

例 7.3.4. 回到例 7.3.3 中, 试问怎么计算 $E(Y)$? 一种方法是求出联合密度函数 $f(x, y)$, 然后计算 $E(Y) = \int \int y f(x, y) dx dy$. 令一种更简单的方式可以分两步来实现. 首先计算 $E(Y|X) = (1+X)/2$, 从而

$$\begin{aligned} E(Y) &= E[E(Y|x)] = E((1+X)/2) \\ &= \frac{1+E(X)}{2} = \frac{(1+(1/2))}{2} = \frac{3}{4} \end{aligned}$$

7.3.7 方差的应用: 过拟合与偏差-方差分解

在机器学习领域, 我们将现有的数据划分为训练集和测试集, 在训练集上训练一个模型 \hat{f} . 如果模型在训练集上表现很好 (如对于分类问题, 表现好意味着分类准确率高), 而在测试集上表现很差, 则此时的模型输入过拟合状态. 为了避免过拟合, 我们需要在模型的拟合能力与复杂度之间进行权衡. 拟合能力强的模型一般复杂度会比较高, 容易导致过拟合. 相反, 如果限制模型的复杂度, 降低其拟合能力, 又可能会导致欠拟合. 因此, 如何在模型的拟合能力和复杂度之间取得一个较好的平衡, 对一个机器学习算法来讲十分重要. 偏差-方差分解 (Bias-Variance Decomposition) 为我们提供一个很好的分析和指导工具.

以回归问题为例. 假设样本的真实分布是 $p_r(x, y)$, 并采用平方损失函数, 模型 $f(x)$ 的期望误差为:

$$\mathcal{R}(f) = E_{(x,y) \sim p_r(x,y)} [(y - f(x))^2]$$

那么最优模型为:

$$f^*(x) = E_{y \sim p_r(y|x)} [y]$$

其中 $p_r(y|x)$ 为样本的真实条件分布, $f^*(x)$ 为使用平方损失作为优化目标的最优模型, 其损失为:

$$\epsilon = E_{(x,y) \sim p_r(x,y)} [(y - f^*(x))^2]$$

通常损失 ϵ 是由样本分布以及噪声引起的, 无法通过优化模型来减少.

期望错误可以分解为:

$$\begin{aligned} \mathcal{R} &= E_{(x,y) \sim p_r(x,y)} [(y - f^*(x) + f^*(x) - f(x))^2] \\ &= E_{(x,y) \sim p_r(x,y)} [(y - f^*(x))^2] + E_{(x,y) \sim p_r(x,y)} [(f^*(x) - f(x))^2] \\ &= \epsilon + E_{x \sim p_r(x)} [(f^*(x) - f(x))^2] \end{aligned} \quad (7.2)$$

其中

$$\begin{aligned} 2E_{(x,y) \sim p_r(x,y)} [(y - f^*(x))(f^*(x) - f(x))] &= 2 \int_x \int_y p_r(x, y)(y - f^*(x))(f^*(x) - f(x)) dx dy \\ &= 2 \int_x (f^*(x) - f(x)) dx \int_y p_r(x, y)(y - f^*(x)) dy \end{aligned}$$

对于给定的 x_0 :

$$\begin{aligned} \int_y p_r(x_0, y)((y - f^*(x_0)) dy &= \int_y p_r(x_0, y)(y - f^*(x_0)) dy \\ &= p_r(x_0) \int_y p_r(y|x_0)(y - f^*(x_0)) dy \end{aligned}$$

由于

$$f^*(x_0) = E_{y \sim p_r(y|x_0)} [y] = \int_y p_r(y|x_0) y dy$$

故

$$\int_y p_r(x_0, y)((y - f^*(x_0))(f^*(x_0) - f(x_0))dy = 0$$

$$2E_{(x,y) \sim p_r(x,y)} [(y - f^*(x))(f^*(x) - f(x))] = 0$$

式(7.4)中的第一项是当前训练出的模型与最优模型之间的差距，是机器学习算法可以优化的正是目标.

在实际训练一个模型 $f(x, y)$ 时，训练集 D 是从真实分布 $p_r(x, y)$ 上独立同分布地采样出来的有限样本集合. 不同的训练集会得到不同的模型. 令 $f_D(x)$ 表示在训练集 D 学习到的模型，一个机器学习算法（包括模型以及优化算法）的能力可以用不同训练集上的模型的平均性能来评价.

对于单个样本 x ，不同训练集 D 得到模型 $f_D(x)$ 和最优模型 $f^*(x)$ 的期望差距为

$$\begin{aligned} E_D [(f_D(x) - f^*(x))^2] &= E_D [(f_D(x) - E_D [f_D(x)] + E_D [f_D(x)] - f^*(x))^2] \\ &= (E_D [f_D(x)] - f^*(x))^2 + E_D [(f_D(x) - E_D [f_D(x)])^2] \end{aligned} \quad (7.3)$$

式(7.3)中第一项 $(E_D [f_D(x)] - f^*(x))^2$ 称为偏差 (Bias) 的平方，记为 $(bias.x)^2$ ，是指一个模型在不同训练集上的平均性能和最优模的差异. 第二项 $E_D [(f_D(x) - E_D [f_D(x)])^2]$ 称为方差 (Variance)，记为 $variance.x$ ，是指一个模型在不同训练集上的差异，可以用来衡量一个模型是否容易过拟合.

用 $E_D [(f_D(x) - f^*(x))^3]$ 来代替式(7.4)中的 $(f(x) - f^*(x))^2$ ，则期望错误可写成:

$$\begin{aligned} \mathcal{R}(f) &= E_{x \sim p_r(x)} [E_D [(f_D(x) - f^*(x))^2]] + \epsilon \\ &= (bias)^2 + variance + \epsilon \end{aligned} \quad (7.4)$$

其中:

$$(bias)^2 = E_x [(E_D [f_D(x)] - f^*(x))^2]$$

$$variance = E_x [E_D [(f_D(x) - E_D [f_D(x)])^2]]$$

所以最小化期望误差等价于最小化偏差与方差之和.

图7.2给出了机器学习模型的四种偏差和方差组合情况. 每个图的中心点为最优模型 $f^*(x)$ ，蓝点为不同训练集 D 上得到的模型 $f_D(x)$. 图7.2a 给出了一种理想情况，方差和偏差都比较小. 图7.2b 为高偏差低方差的情况，表示模型的泛化能力很好，但拟合能力不足. 图7.2c 为低偏差高方差的情况，表示模型的拟合能力很好，但泛化能力比较差. 当训练数据比较少时会导致过拟合. 图7.2d 为高偏差高方差的情况，是一种最差的情况.

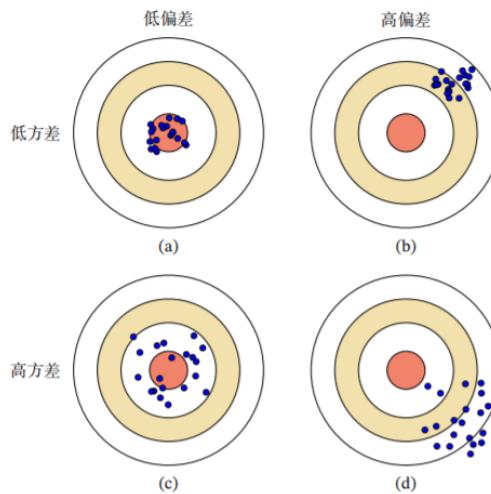


图 7.2: 偏差与方差的组合

方差一般会随着训练样本的增加而减少。当样本比较多时，方差比较少，这时可以选择能力强的模型来减少偏差。然而在很多机器学习任务上，训练集往往都比较有限，最优的偏差和最优的方差就无法兼顾。

随着模型复杂度的增加，模型的拟合能力变强，偏差减少而方差增大，从而导致过拟合。以结构错误最小化为例，我们可以调整正则化系数来控制模型的复杂度。当 λ 变大时，模型复杂度会降低，可以有效地减少方差，避免过拟合，但偏差会上升。当 λ 过大时，总的期望错误反而会上升。因此，一个好的正则化系数 λ 需要在偏差和方差之间取得比较好的平衡。图 7.3 给出了机器学习模型的期望错误、偏差和方差随复杂度的变化情况，其中红色虚线表示最优模型。最优模型并不一定是偏差曲线和方差曲线的交点。

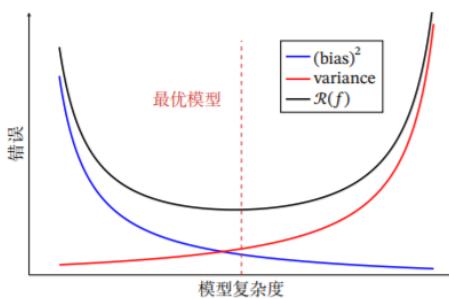


图 7.3: 机器学习模型的期望错误、偏差和方差随复杂度的变化情况

偏差和方差分解给机器学习模型提供了一种分析途径，但在实际操作中难以直接衡量。一般来说，当一个模型在训练集上的错误率比较高时，说明模型的拟合能力不够，偏差比较高。这种情况可以通过增加数据特征、提高模型复杂度、减少正则化系数等操作来改进模型。当模型在训练集上的错误率比较低，但验证集上的错误率比较高时，说明模型过拟合，方差比较高。这种情况可以通过降低模型复杂度、加大正则化系数、引入先验等方法来缓解。此外，还有一种有效降低方差的方法为集成模型，即通过多个高方差模型的平均来降低方差。

7.4 概率不等式

不等式对于一些很难计算的量非常有用，它也常用于收敛定理，有关收敛定理将在下一节具体讨论，这里首先介绍的不等式是马尔可夫不等式。

定理 7.4.1. (马尔可夫不等式) 令 X 为一非负随机变量，假设 $E(X)$ 存在，对任意 $t > 0$ 有

$$P(X > t) \leq \frac{E(X)}{t}$$

证明. 因为 $X > 0$ ，所以

$$\begin{aligned} E(X) &= \int_0^\infty xf(x)dx = \int_0^t xf(x)dx + \int_t^\infty xf(x)dx \\ &\geq \int_t^\infty xf(x)dx \geq t \int_t^\infty f(x)dx = tP(X > t) \end{aligned}$$

□

定理 7.4.2. (切比雪夫不等式) 令 $\mu = E(X)$, $\sigma^2 = V(X)$, 则

$$P(|X - \mu| \geq t) \leq \frac{\sigma^2}{t^2}, \quad P(|Z| \geq k) \leq \frac{1}{k^2}$$

其中, $Z = (x - \mu)/\sigma$, 特别地, $P(|Z| > 2) \leq 1/4$, $P(|Z| > 3) \leq 1/9$

证明. 利用马尔可夫不等式可得

$$P(|X - \mu| \geq t) = P(|X - \mu|^2 \geq t^2) \leq \frac{E(X - \mu)^2}{t^2} = \frac{\sigma^2}{t^2}$$

第二部分令 $t = k\sigma$ 即得。 □

例 7.4.1. 假设检验是一种预测方法，涉及 n 种检验情形，以神经网络为例。如果预测错误则令 $X_i = 1$ ，反之则令 $X_i = 0$ 。从而 $\bar{X}_n = n^{-1} \sum_{i=1}^n X_i$ 是观察到的误差率。每个 X_i 可认为服从未知均值 p 的伯努利分布。想求出未知的真实误差率 p 。从直觉上判断, \bar{X}_n 应与 p 非常接近, \bar{X}_n 不在 p 附近 ϵ 的范围内的概率为多少？已知 $V(\bar{X}_n) = V(X_1)/n = p(1-p)/n$ ，从而

$$P(|X_n - p| > \epsilon) \leq \frac{V(\bar{X}_n)}{\epsilon^2} = \frac{p(1-p)}{n\epsilon^2} \leq \frac{1}{4n\epsilon^2}$$

上式利用了不等式 $p(1-p) \leq 1/4$ ，对于 $\epsilon = 2$ 和 $n = 100$ ，所求的界为 0.0625。

定理 7.4.3. (霍夫丁不等式) 令 Y_1, \dots, Y_n 为独立观察值, 满足 $E(Y_i) = 0$, 且 $a_i \leq Y_i \leq b_i$ 。令 $\epsilon > 0$, 则对于任意 $t > 0$ 有

$$P\left(\sum_{i=1}^n Y_i \geq \varepsilon\right) \leq e^{-te} \prod_{i=1}^n e^{t^2(b_i-a_i)^2/8}$$

如下证明将用到泰勒定理: 如果 g 为光滑函数, 则存在数值 $\xi \in (0, \mu)$ 使得 $g(u) = g(0) + ug'(0) + (u^2/2)g''(\xi)$ 。

证明. 对任意 $t > 0$, 由马尔可夫不等式得

$$\begin{aligned} P\left(\sum_{i=1}^n Y_i \geq \varepsilon\right) &= P\left(t \sum_{i=1}^n Y_i \geq t\varepsilon\right) = P\left(e^{t \sum_{i=1}^n Y_i} \geq e^{t\varepsilon}\right) \\ &\leq e^{-te} E\left(e^{t \sum_{i=1}^n Y_i}\right) = e^{-te} \prod_i E\left(e^{tY_i}\right) \end{aligned}$$

因为 $a_i \leq Y_i \leq b_i$, 可将 Y_i 写成 a_j, b_j 的凸组合, 即 $Y_i = \alpha b_i + (1 - \alpha)a_i$, 其中, $\alpha = (Y_i - a_i) / (b_i - a_i)$, 所以根据 e^{ty} 的凸性得到

$$e^{tY_i} \leq \frac{Y_i - a_i}{b_i - a_i} e^{tb_i} + \frac{b_i - Y_i}{b_i - a_i} e^{ta_i}$$

两边取期望并利用 $E(Y_i) = 0$ 得

$$E(e^{tY_i}) \leq -\frac{a_i}{b_i - a_i} e^{tb_i} + \frac{b_i}{b_i - a_i} e^{ta_i} = e^{g(\omega)}$$

其中, $u = t(b_i - a_i)$, $g(u) = -\gamma u + \log(1 - \gamma + \gamma e^\omega)$, $\gamma = -a_i / (b_i - a_i)$, 注意到 $g(0) = g'(0) = 0$ 且对所有 $u > 0$, $g''(u) < 1/4$, 根据泰勒定理, 存在 $\xi \in (0, \mu)$ 满足

$$\begin{aligned} g(u) &= g(0) + ug'(0) + \frac{u^2}{2}g''(\xi) \\ &= \frac{u^2}{2}g''(\xi) \leq \frac{u^2}{8} = \frac{t^2(b_i - a_i)^2}{8} \end{aligned}$$

因此

$$Be^{tY_i} \leq e^{g(\omega)} \leq e^{t^2(b_i - a_i)^2/8}$$

□

定理 7.4.4. 令 X_1, \dots, X_n 服从参数为 p 的伯努利分布, 则对于任意 $\epsilon > 0$ 有

$$P(|\bar{X}_n - p| > \varepsilon) \leq 2e^{-2ne^2}$$

其中, $\bar{X}_n = n^{-1} \sum_{i=1}^n X_i$

证明. 令 $Y_i = (1/n)(X_i - p)$, 则 $E(Y_i) = 0$, 令 $a = -p/n, b = (1 - p)/n$, 则 $a \leq Y_i \leq b$ 且 $(b - a)^2 = 1/n^2$, 根据霍夫丁不等式得

$$P(\bar{X}_n - p > \varepsilon) = P\left(\sum_i Y_i > \varepsilon\right) \leq e^{-te} e^{t^2/(8n)}$$

□

例 7.4.2. 令 X_1, \dots, X_n 服从参数为 p 的伯努利分布, 令 $n = 100, \epsilon = 0.2$, 由切比雪夫不等式可得

$$P(|\bar{X}_n - p| > \epsilon) \leq 0.0625$$

由霍夫丁不等式得

$$P(|\bar{X}_n - p| \leq 0.2) \leq 2e^{-2(100)(0.2)^2} = 0.00067$$

这比 0.0625 要小很多。

霍夫丁不等式提供了一种建立在参数 p 的二项式分布置信区间的简单方法。固定 $\alpha > 0$ 并令

$$\varepsilon_n = \sqrt{\frac{1}{2n} \log \left(\frac{2}{\alpha} \right)}$$

由霍夫丁不等式可知

$$P(|\bar{X}_n - p| > \varepsilon_n) \leq 2e^{-2n\varepsilon_n^2} = \alpha$$

令 $C = (\bar{X}_n - \varepsilon_n, \bar{X}_n + \varepsilon_n)$, 则 $P(p \notin C) = P(|\bar{X}_n - p| > \varepsilon_n) \leq \alpha$ 。因此, $P(p \in C) \geq 1 - \alpha$, 也即随机区间 C 包括真实参数 p 的概率为 $1 - \alpha$; 称 C 为 $1 - \alpha$ 置信区间。

下面的不等式对于正态分布随机变量的概率范围确定非常有用。

定理 7.4.5. (Mill 不等式) 令 $Z \sim N(0, 1)$, 则:

$$P(|Z| > t) \leq \sqrt{\frac{2}{\pi}} \frac{e^{-t^2/2}}{t}$$

有关期望的不等式

本节介绍有关期望的两个不等式。

定理 7.4.6. (柯西-施瓦兹不等式) 如果 X 和 Y 具有有限方差, 则

$$E|XY| \leq \sqrt{E(X^2)E(Y^2)}$$

高等数学或类似课程中曾经学过, 如果对任意 x, y 以及 $\alpha \in [0, 1]$, 函数 g 满足

$$g(\alpha x + (1 - \alpha)y) \leq \alpha g(x) + (1 - \alpha)g(y)$$

则函数 g 是凸函数。如果对于所有 x , 函数 g 二阶可导, 且 $g''(x) \geq 0$, 则可证明 g 是凸的, g 位于与其相切于任一点的直线的上方, 该直线称为切线。如果 $-g$ 是凸函数, 则 g 是凹函数。凸函数的例子如 $g(x) = x^2, g(x) = e^x$; 凹函数的例子如 $g(x) = -x^2, g(x) = \log x$ 。

定理 7.4.7. (詹森不等式) 如果 g 为凸函数, 则

$$Eg(X) \geq g(EX)$$

如果 g 为凹函数, 则

$$Eg(X) \leq g(EX)$$

证明. 令直线 $L(x) = a + bx$ 与 $g(x)$ 相切于点 $E(X)$, 因为 g 是凸函数, 它位于直线 $L(x)$ 的上方, 所以

$$Eg(X) \geq EL(X) = E(a + bX) = a + bE(X) = L(E(X)) = g(EX)$$

由詹森不等式可知 $E(X^2) \geq (EX)^2$; 如果 X 为正, 则 $E(1/X) \geq 1/E(X)$; 因为对数函数是凹函数, 所以 $E(\log X) \leq \log E(X)$. \square

概率不等式在统计机器学习中的应用: 泛化能力分析

定义 7.4.1. (泛化误差) 如果学到的模型是 \hat{f} , 那么用这个模型对未知数据预测的误差即为泛化误差 (*generalization error*):

$$R_{exp}(\hat{f}) = E_p[L(Y, \hat{f}(X))] = \int_{\mathcal{X} \times \mathcal{Y}} L(y, \hat{f}(x))P(x, y)dxdy$$

泛化误差的概率上界称为泛化误差上界。具体来说就是通过比较两种学习方法的泛化误差上界的大小来比较它们的优劣。泛化误差上界通常具有以下性质:

- 它是样本容量的函数, 单样本容量增加时, 泛化上界趋于 0
- 它是假设空间容量的函数, 假设空间容量越大, 模型就越难学, 泛化误差上界就越大

考虑二分类问题, 已知训练数据集 $\mathbb{T} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, N 是样本容量, \mathbb{T} 是从联合概率分布 $P(X, Y)$ 独立同分布产生的, $X \in \mathbb{R}^n, Y \in \{-1, +1\}$ 。假设空间是函数的有限集合 $\mathcal{F} = \{f_1, f_2, \dots, f_d\}$, d 是函数个数。设 f 是从 \mathcal{F} 中选取的函数。损失函数是 0-1 损失。关于 f 的期望风险和经验风险分别是

$$\begin{aligned} R(f) &= E[L(Y, f(X))] \\ \hat{R}(f) &= \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) \end{aligned}$$

经验风险最小化函数是

$$f_N = \arg \min_{f \in \mathcal{F}} \hat{R}(f)$$

f_N 依赖训练数据集的样本容量 N 。我们更关心 f_N 的泛化能力

$$R(f_N) = E[L(Y, f_N(X))]$$

接下来我们讨论从有限集合 $\mathcal{F} = \{f_1, f_2, \dots, f_d\}$ 中任意选出的函数 f 的泛化误差上界。

定理 7.4.8. [泛化误差上界] 对于二分类问题, 当假设空间是有限个函数的集合 $\mathcal{F} = \{f_1, f_2, \dots, f_d\}$ 时, 对于任意函数 $f \in \mathcal{F}$, 至少以概率 $1 - \delta$, $0 \leq \delta \leq 1$, 以下不等式成立:

$$R(f) \leq \hat{R}(f) + \epsilon(d, N, \delta) \tag{7.5}$$

其中

$$\epsilon(d, N, \delta) = \sqrt{\frac{1}{2N} (\log d + \log \frac{1}{\delta})} \tag{7.6}$$

不等式7.5左端 $R(f)$ 是泛化误差，右端即泛化误差的上界。在泛化误差上界中，第1项是训练误差，训练误差越小，泛化误差也越小。第2项 $\epsilon(d, N, \delta)$ 是 N 的单调递减函数，当 N 趋于无穷时趋于0；同时它也是 $\sqrt{\log d}$ 阶的函数，假设空间包含的函数越多，其值越大。

证明. 对任意函数 $f \in \mathcal{F}$, $\hat{R}(f)$ 是 N 个独立的随机变量 $L(Y, f(X))$ 的样本均值, $R(f)$ 是随机变量 $L(Y, f(X))$ 的期望值。如果损失函数取值于区间 $[0, 1]$, 即对所有 i , $[a_i, b_i] = [0, 1]$, 那么由 Hoeffding 不等式不难得知, 对 $\epsilon > 0$, 以下不等式成立:

$$P(R(f) - \hat{R}(f) \geq \epsilon) \leq \exp(-2N\epsilon^2)$$

由于 $\mathcal{F} = \{f_1, f_2, \dots, f_d\}$ 是一个有限集合, 故

$$\begin{aligned} P(\exists f \in \mathcal{F} : R(f) - \hat{R}(f) \geq \epsilon) &= P\left(\bigcup_{f \in \mathcal{F}} \{R(f) - \hat{R}(f) \geq \epsilon\}\right) \\ &\leq \sum_{f \in \mathcal{F}} P(R(f) - \hat{R}(f) \geq \epsilon) \\ &\leq d \exp(-2N\epsilon^2) \end{aligned}$$

令 $\delta = d \exp(-2N\epsilon^2)$, 则等价地, 对任意 $f \in \mathcal{F}$, 有 $P(R(f) - \hat{R}(f) \geq \epsilon) \leq \delta$

或者有 $P(R(f) < \hat{R}(f) + \epsilon) \geq 1 - \delta$

即至少以概率 $1 - \delta$ 有 $R(f) < \hat{R}(f) + \epsilon$, 其中 ϵ 可从 $\delta = d \exp(-2N\epsilon^2)$ 中反解, 即定理中的表达式7.5。

□

7.5 大数定律与中心极限定理

7.5.1 引言

概率论最重要的一方面就是关注随机变量序列的趋势, 这部分内容称为大样本理论或极限理论或渐进理论. 最基本的问题是: 关于随机变量序列 X_1, X_2, \dots 的极限性质可以作何论断? 因为统计与数据挖掘涉及大量数据, 自然而然地, 也会关心当收集到越来越多的数据时会发生什么.

在积分理论中, 如果对任意 $\epsilon > 0$, $|x_n - x| < \epsilon$ 对充分大的 n 都成立, 则称实数序列 x_n 收敛于极限 x . 在概率论中, 极限的概念更加深奥, 回忆积分理论中的介绍, 假设对所有 n 有 $x_n = x$, 则 $\lim_{n \rightarrow \infty} x_n = x$. 考虑该例子的概率模型, 假设 X_1, X_2, \dots 为独立同分布随机序列, 服从 $N(0, 1)$ 分布, 因为所有变量具有相同的分布, 所以可以尝试着称 X_n “收敛于” $X \sim N(0, 1)$, 但这种描述并不十分精确, 因为对所有 n , $P(X_n = X) = 0$ (两个连续随机变量相同的概率为0)

还有另外一个例子, 假设 $X_1, X_2, \dots \sim N(0, 1/n)$, 从直觉上判断, 当 n 很大时, X_n 集中在0附近, 所以很希望称 X_n 收敛于0, 但是对所有 n , $P(X_n = 0) = 0$. 很明显, 需要其他工具来讨论更严格意义上的随机变量的收敛。

本章将主要介绍两种思想

1. 大数定律说明样本均值 $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ 依概率收敛于期望 $\mu = E(X_i)$, 这意味着 \bar{X}_n 以很高的概率趋于 μ .
2. 中心极限定理说明 $\sqrt{n}(\bar{X}_n - \mu)$ 依分布收敛于正态分布, 这也意味着对很大的 n , 样本均值渐进服从正态分布.

7.5.2 大数定律

依概率收敛和依分布收敛

定义 7.5.1. 令 X_1, X_2, \dots 为随机变量序列, X 为另一随机变量, 用 F_n 表示 X_n 的 CDF, 用 F 表示 X 的 CDF

1. 如果对任意 $\epsilon > 0$, 当 $n \rightarrow \infty$ 时有

$$P(|X_n - X| > \epsilon) \rightarrow 0$$

则 X_n 依概率收敛于 X , 记为 $X_n \xrightarrow{P} X$.

2. 如果对所有的 F 的连续点 t , 有

$$\lim_{n \rightarrow \infty} F_n(t) = F(t)$$

则 X_n 依分布收敛于 X , 记为 $X_n \rightsquigarrow X$

当求 X 服从点分布时, 需要改变一下符号, 如果 $P(X = c) = 1$ 且 $X_n \xrightarrow{P} X$, 则记 $X_n \xrightarrow{P} c$, 类似地, 如果 $X_n \rightsquigarrow X$, 则记 $X_n \rightsquigarrow c$.

这里再介绍另外一种形式的收敛, 这种收敛对证明概率中的收敛很有用.

定义 7.5.2. 如果 $n \rightarrow \infty$ 时有

$$E(X_n - X)^2 \rightarrow 0,$$

则称 X_n 均方意义上收敛于 X (也称 L_2 收敛), 记为 $X_n \xrightarrow{qm} X$.

同上面类似, 如果 X 服从在 c 点的点分布, 则用 $X_n \xrightarrow{qm} c$ 代替 $X_n \xrightarrow{qm} X$

例 7.5.1. 令 $X_n \sim N(0, 1/n)$, 从直觉上判断, 当 n 很大时, X_n 集中在 0 附近, 所以就希望称 X_n 以概率收敛于 0, 那么现在来看一下这是否正确. 令 F 为在零点的点分布的分布函数, 注意到 $\sqrt{n}X_n \sim N(0, 1)$, 令 Z 表示标准正态分布随机变量, 对于 $t < 0$, 因为 $\sqrt{n}t \rightarrow -\infty$, 所以 $F_n(t) = P(X_n < t) = P(\sqrt{n}X_n < \sqrt{n}t) = P(Z < \sqrt{n}t) \rightarrow 0$; 对于 $t > 0$, 因为 $\sqrt{n}t \rightarrow \infty$, 所以 $F_n(t) = P(X_n < t) = P(\sqrt{n}X_n < \sqrt{n}t) = P(Z < \sqrt{n}t) \rightarrow 1$. 因此, 对所有 $t \neq 0$ 有 $F_n(t) \rightarrow F(t)$, 所以 $X_n \rightsquigarrow 0$. 注意 $F_n = 1/2 \neq F(1/2) = 1$, 所以在 $t = 1$ 处收敛不成立. 这并不影响结果, 因为 $t = 0$ 不是 F 的连续点, 而分布收敛的定义仅需连续的点收敛即可.

现在再考察概率收敛，对于任意 $\epsilon > 0$ ，使用马尔科夫不等式，当 $n \rightarrow \infty$ 时有

$$P(|X_n| > \epsilon) = P(|X_n|^2 > \epsilon^2) \leq \frac{E(X_n^2)}{\epsilon^2} = \frac{1/n}{\epsilon^2} \rightarrow 0$$

因此 $X_n \xrightarrow{P} 0$

下面给出定理来说明各种收敛类型之间的关系

定理 7.5.1. (a) $X_n \xrightarrow{qm} X$ 意味着 $X_n \xrightarrow{P} X$

(b) $X_n \xrightarrow{P} X$ 意味着 $X_n \rightsquigarrow X$

(c) 如果 $X_n \rightsquigarrow X$ 且对于实数 c 有 $P(X = c) = 1$ ，则 $X_n \xrightarrow{P} X$.

证明. (a) 假设 $X_n \xrightarrow{qm} X$ 成立，对固定 $\epsilon > 0$ ，利用马尔科夫不等式

$$P(|X_n - X| > \epsilon) = P(|X_n - X|^2 > \epsilon^2) \leq \frac{E|X_n - X|^2}{\epsilon^2} \rightarrow 0$$

(b) 的证明有些复杂，暂省略。

(c) 对固定 $\epsilon > 0$

$$\begin{aligned} P(|X_n - c|) &= P(X_n < c - \epsilon) + P(X_n > c + \epsilon) \\ &\leq P(X_n \leq c - \epsilon) + P(X_n > c + \epsilon) \\ &\rightarrow F(c - \epsilon) + 1 - F(c + \epsilon) \\ &= 0 + 1 - 1 = 0 \end{aligned}$$

□

下面来说明反向并不成立。

依概率收敛不能推出均方意义下收敛 令 $U \sim Uniform(0, 1)$, $X_n = \sqrt{n}I_{0,1/n}(U)$, 则 $P(|X_n| > \epsilon) = P(\sqrt{n}I_{0,1/n}(U) > \epsilon) = P(0 \leq U < 1/n) = 1/n \rightarrow 0$. 因此 $X_n \xrightarrow{P} 0$, 但是对所有 n 有 $E(X_n^2) = n \int_0^{1/n} du = 1$, 所以均方意义下不收敛。

依分布收敛不能推出依概率收敛 令 $X \sim N(0, 1)$, $X_n = -X$, 其中 $n = 1, 2, 3, \dots$; 因此, $X_n \sim N(0, 1)$, 即对所有 n , X_n 与 X 同分布, 所以对所有 x , $\lim_n F_n(x) = F(x)$, 也就是说 $X_n \rightsquigarrow X$ 但是 $P(|X_n - X| > \epsilon) = P(|2X| > \epsilon) = P(|X| > \frac{\epsilon}{2}) \neq 0$, 也即 X_n 不依概率收敛于 X .

弱大数定律

接下来的讨论议题可以被称为是概率论中最伟大的成果，它就是大数定律。大数定律指出大量样本的均值近似于分布的均值，例如，无数次投掷硬币出现正面的概率趋近于 $1/2$ ，下面对该定律简要描述。

令 X_1, X_2, \dots 为 IID 样本，令 $\mu = E(X_1)$, $\sigma^2 = V(X_1)$. 则样本均值为 $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$, $E(\bar{X}_n) = \mu$, $V(\bar{X}_1) = \sigma^2/n$

定理 7.5.2. (弱大数定律)(the weak law of large numbers(WLLN)) 若 X_1, X_2, \dots, X_n 为 IID 样本, 则 $\bar{X}_n \xrightarrow{P} \mu$.

WLLN 的含义: 当 n 逐渐变大时, X_n 的分布靠近 μ , 称 \bar{X}_n 为 μ 的一致估计(一致性)

在定理条件下, 当样本数目 N 无限增加时, 随机样本均值将几乎变成一个常量

样本方差也依概率收敛于方差 σ^2

证明. 假设 $\sigma < \infty$, 该假设并不是必需的, 但有利于简化证明, 利用切比雪夫不等式得:

$$P(|X_n - \mu| > \epsilon) \leq \frac{V(\bar{X}_n)}{\epsilon^2} = \frac{\sigma^2}{n\epsilon}$$

当 $n \rightarrow 0$ 时, 上式趋于 0. □

例 7.5.2. 假定抛一枚硬币, 出现正面的概率是 p , 令 X_i 表示每次结果, 因此 $p = P(X_i = 1) = E(X_i)$, 当抛 n 次后正面次数所占比例为 \bar{X}_n , 根据大数定律 X_n 以概率收敛于 p , 它意味着当 n 很大时, X_n 的分布会紧密围绕在 p 的附近. 假设 $p = 1/2$, 需要多大的 n 才能使得 $P(0.4 \leq \bar{X}_n \leq 0.6) = 0.7$ 呢? 首先, $E(\bar{X}_n) = p = 1/2$ 且 $V(\bar{X}_n) = \sigma^2/n = p(1-p)/n = 1/(4n)$, 然后便有切比雪夫不等式

$$\begin{aligned} P(0.4 \leq \bar{X}_n \leq 0.6) &= P(|\bar{X}_n - \mu| \leq 0.1) \\ &= 1 - P(|\bar{X}_n - \mu| \geq 0.1) \\ &\geq 1 - \frac{1}{4n(0.1)^2} = 1 - \frac{25}{n} \end{aligned}$$

当 $n = 84$ 时就能保证上式大于 0.7.

大数定律的推广和应用

我们前面介绍过在函数集 $Q(z, \alpha), \alpha \in \Lambda$ 上最小化风险泛函的问题

$$R(\alpha) = \int Q(z, \alpha) dF(z), \alpha \in \Lambda$$

其中, 分布函数 $F(z)$ 是未知的, 但给定了依据分布函数抽取的独立同分布数据 z_1, \dots, z_t . 为了求解上述问题, 我们提出了经验风险最小化原则. 根据这一原则, 我们用最小化经验风险泛函

$$R_{emp}(\alpha) = \frac{1}{t} \sum_{i=1}^t Q(z_i, \alpha), \alpha \in \Lambda$$

来代替最小化泛函. 设

$$Q(z, \alpha_t) = Q(z, \alpha(z_1, \dots, z_t))$$

为最小化泛函的一个函数. 经验风险最小化理论的基本问题是描述经验风险最小化原则一致性的条件. 下面我们给出一致性的经典定义.

大数定律指出 X_n 的分布会聚集在 u 附近, 这还不能描述 X_n 的概率性质, 为此还需要中心极限定理.

假设 X_1, X_2, \dots, X_n 为均值 μ , 方差 σ^2 的 IID 序列, 中心极限定理(CLT)指出 $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ 近似服从期望为 μ , 方差为 σ^2/n 的正态分布, 这一结论非常卓越, 因为只需要对 X_i 的分布的均值和方差进行要求, 没有其他别的条件.

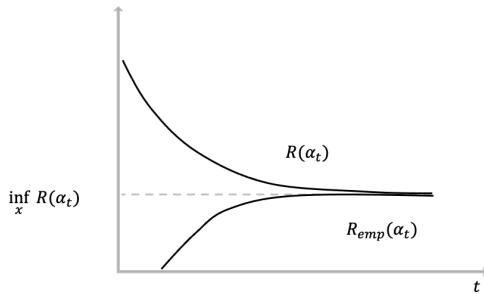


图 7.4: 如果期望风险 $R(\alpha_t)$ 和经验风险 $R_{emp}(\alpha_t)$ 都收敛于风险最小可能值 $\inf_{\alpha \in \Lambda} R(\alpha)$, 则学习过程是一致的

定义 7.5.3. 对于函数集 $Q(z, \alpha), \alpha \in \Lambda$ 和概率分布函数 $F(z)$, 如果下面两个序列依概率收敛于同一极限

$$R(\alpha_t) \xrightarrow[t \rightarrow \infty]{P} \inf R(\alpha) \quad (7.7)$$

$$R_{emp}(\alpha_t) \xrightarrow[t \rightarrow \infty]{P} \inf R(\alpha) \quad (7.8)$$

则, 我们称经验风险最小化原则 (方法) 是一致的。

换句话说, 如果经验风险最小化方法能够提供一个函数序列 $Q(z, \alpha_t), \alpha_t \in \Lambda$ 使得期望风险和经验风险依概率收敛于 (对于给定的函数集) 最小的可能风险值, 则经验风险最小化方法是一致的。方程(7.7)表明, 对于给定的函数集, 所得风险值序列收敛于最小的可能风险; 方程(7.8)表明, 经验风险序列的极限估计出风险的最小可能值。

统计学习理论的核心问题之一是找到经验风险最小化方法的一致性条件。而经验风险最小化的一致性分析在本质上是与两种经验过程的收敛性分析相联系的。

- 设概率分布函数 $F(z)$ 定义在空间 $z \in \mathbb{R}^n$ 上, $Q(z, \alpha), \alpha \in \Lambda$ 为一个 (关于分布 $F(z)$ 的) 可测函数集, 又设 z_1, \dots, z_t, \dots 为一个独立同分布的向量序列。考虑随机变量序列

$$\xi^l = \sup_{\alpha \in \Lambda} \left| \int Q(z, \alpha) dF(z) - \frac{1}{l} \sum_{i=1}^l Q(z_i, \alpha) \right|, l = 1, 2, \dots \quad (7.9)$$

我们将这一依赖于测度 $F(z)$ 和函数集 $Q(z, \alpha), \alpha \in \Lambda$ 的随机变量序列称为双边经验过程。

- 我们的问题是要描述一组条件, 在这组条件下上述经验过程依概率收敛于零。

换句话说, 我们的问题是描述一组条件, 使得对于任意正的 ϵ 下列收敛性成立:

定义 7.5.4. 假设测度 $F(z)$ 和函数集 $Q(z, \alpha), \alpha \in \Lambda$, 对于任意正的 ϵ 有

$$P \left\{ \sup_{\alpha \in \Lambda} \left| \int Q(z, \alpha) dF(z) - \frac{1}{l} \sum_{i=1}^l Q(z, \alpha) \right| > \epsilon \right\} \xrightarrow[t \rightarrow \infty]{} 0$$

我们称上述关系式为给定函数集上均值到数学期望的一致收敛性, 或者简称为一致收敛性。

定义 7.5.5. 与经验过程 ξ^l 一起, 我们考虑由随机值序列给出的单边经验过程

$$\xi_+^l = \sup_{\alpha \in \Lambda} \left(\int Q(z, \alpha) dF(z) - \frac{1}{l} \sum_{i=1}^l Q(z, \alpha) \right), l = 1, 2, \dots$$

我们称上述关系式为在给定的函数集上均值到数学期望的一致单边收敛性, 或者简称为一致单边收敛性。

值得注意的是如果函数集 $Q(z, \alpha), \alpha \in \Lambda$ 仅包含一个元素, 则由式7.9定义的随机变量序列 ξ^l 永远依概率收敛于零。这一事实构成了统计学的主要定律, 即大数定律。随着 l 的增加, 均值序列收敛于随机变量的期望 (如果期望存在)。

将大数定律推广到函数集 $Q(z, \alpha), \alpha \in \Lambda$ 包含有限个元素的情况是容易的。与包含有限个元素情况相比, 对于包含无穷多个元素的函数集 $Q(z, \alpha), \alpha \in \Lambda$ 随机变量序列 ξ^l 不一定收敛于零, 那么就出现一个问题: 描述函数集 $Q(z, \alpha), \alpha \in \Lambda$ 和概率测度 $F(z)$ 的性质, 使得在这些性质下随机变量序列 ξ^l 依概率收敛于零。在这种情况下, 我们称大数定律在函数空间 (函数 $Q(z, \alpha), \alpha \in \Lambda$ 的空间) 中成立, 或者给定函数集上存在均值到期望的一致 (双边) 收敛性。因此, 函数空间中的大数定律 (均值到期望的一致双边收敛性) 的存在性问题可以看成经典大数定律的推广。应该注意的是, 在经典统计学中, 并没有考虑一致单边收敛性的存在性问题。由于关键定理指出了分析 ERM 归纳原则一致性问题的方法, 一致单边收敛性的存在性问题变得重要起来了。一致收敛性意味着, 对于充分大的 l , 在给定函数集的所有函数上, 经验风险泛函一致地逼近于风险泛函。在前面, 我们已经证明, 当存在一致收敛性时, 最小化经验风险的函数给出了接近于最小可能风险的风险值。所以, 一致收敛性给出了经验风险最小化方法一致性的充分条件。在这种情况下, 就会出现一个新问题:

是否有可能认为一致收敛性的要求太强? 是否存在这样一种情况, 经验风险最小化方法是一致的, 但一致收敛性不成立?

事实上, 这样的情况是不可能出现的。可以证明一致单边收敛性不但构成了经验风险最小化方法一致性的充分条件, 而且构成了它的必要条件。

定理 7.5.3. 设存在常数 α 和 A , 使得对于函数集 $Q(z, \alpha), \alpha \in \Lambda$ 中的所有函数和给定的分布函数 $F(z)$, 有下列不等式成立:

$$\alpha \leq \int Q(z, \alpha) dF(z) \leq A, \alpha \in \Lambda$$

则, 下面两种表述方式是等价的:

表 7.1: 经典统计学体系和统计学习理论体系的结构

	经典统计学体系	统计学习理论体系
问题的表达	函数的参数估计	利用经验数据最小化期望风险
问题的解决方法	ML 法	ERM 或 SRM 方法
证明	参数估计的有效性	一致大数定律的存在性

1. 对于给定分布函数 $F(z)$, 经验风险最小化方法在函数集 $Q(z, \alpha), \alpha \in \Lambda$ 上是严格一致的。
2. 对于给定分布函数 $F(z)$, 在函数集 $Q(z, \alpha), \alpha \in \Lambda$ 上出现均值到数学期望的一致单边收敛性的。

推论 7.5.1. 假设存在常数 α 和 A , 使得对于函数集 $Q(z, \alpha), \alpha \in \Lambda$ 中的所有函数和集合 \mathcal{P} 中的所有分布函数 $F(z)$, 有下列不等式成立:

$$\alpha \leq \int Q(z, \alpha) dF(z) \leq A, \alpha \in \Lambda$$

则, 下面两种表述方式是等价的:

1. 对于集合 \mathcal{P} 中的任意分布函数 $F(z)$, 经验风险最小化方法在函数集 $Q(z, \alpha), \alpha \in \Lambda$ 上是严格一致的。
2. 对于集合 \mathcal{P} 中的任意分布函数 $F(z)$, 在函数集 $Q(z, \alpha), \alpha \in \Lambda$ 上出现均值到数学期望的一致单边收敛性的。

7.5.3 中心极限定理

定义 7.5.6. (中心极限定理 (CLT)) 令 X_1, \dots, X_n 的均值为 μ , 方差为 σ^2 的 IID 序列, 令 $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$, 则

$$Z_n = \frac{\bar{X}_n - \mu}{\sqrt{V(\bar{X}_n)}} = \frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \rightsquigarrow Z$$

其中 $Z \sim N(0, 1)$, 换句话说, 下式成立:

$$\lim_{n \rightarrow \infty} P(Z_n \leq z) = \Phi(z) = \int_{-\infty}^z \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$$

含义: 有关 \bar{X}_n 概率陈述可以利用正态分布来近似, 注意这仅仅是概率陈述上的近似, 而并不是随机变量本身.

除了 $Z_n \rightsquigarrow N(0, 1)$ 外, 还有其他几个符号可以表示 Z_n 的分布收敛于正态分布, 他们表达的含义本质是一样的, 具体形式如下:

$$Z_n \approx N(0, 1),$$

$$\bar{X}_n \approx N(\mu, \frac{\sigma^2}{n}),$$

$$\bar{X}_n - \mu \approx (0, \frac{\sigma^2}{n}),$$

$$\sqrt{n}(\bar{X}_n - \mu) \approx N(0, \sigma^2),$$

$$\frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \approx N(0, 1),$$

例 7.5.3. 假设每个计算机程序产生误差的数量服从均值为 5 的泊松分布，有 125 个程序，令 X_1, \dots, X_{125} 分别表示程序中的误差数量，求 $P(\bar{X}_n < 5.5)$. 令 $\mu = E(X_1) = \lambda = 5, \sigma^2 = V(X_1) = \lambda = 5$ 则

$$P(\bar{X}_n < 5.5) = P\left(\frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} < \frac{\sqrt{n}(5.5 - \mu)}{\sigma}\right) \approx P(Z < 2.5) = 0.9938$$

中心极限定理说明 $Z_n = \sqrt{n}(\bar{X}_n - \mu)/\sigma$ 近似服从 $(0, 1)$ ，然而 σ 值在大部分情况下是未知的，后面将介绍用 X_1, \dots, X_n 的函数

$$S_n^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2$$

去估计 σ^2 的方法。这又产生了另外一个问题：如果用 S_n 去代替 σ ，中心极限定理还成立吗？答案是肯定的。

定理 7.5.4. 假设跟 CLT 相同条件，则

$$\frac{\sqrt{n}(\bar{X}_n - \mu)}{S_n} \rightsquigarrow N(0, 1)$$

读者或许要问，正态近似的精度有多大那？答案将在 Berry – Esseen 定理中给出。

定理 7.5.5. (Berry-Esseen 定理) 假设 $E|X_1|^3 < \infty$, 则

$$\tilde{|P(Z_n \leq z) - \Phi(z)|} \leq \frac{33}{4} \frac{E|X_1 - \mu|^3}{\sqrt{n}\sigma^3}$$

中心极限定理也存在多元的情形

定理 7.5.6. (多元中心极限定理) 令 X_1, \dots, X_n 为 IID 随机向量，其中

$$\begin{pmatrix} X_{1i} \\ X_{2i} \\ \vdots \\ X_{ki} \end{pmatrix}$$

草稿勿外传

其均值为

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix} = \begin{pmatrix} E(X_{1i}) \\ E(X_{2i}) \\ \vdots \\ E(X_{ki}) \end{pmatrix}$$

方差矩阵为 σ , 令

$$\bar{X} = \begin{pmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_n \end{pmatrix}$$

其中 $\bar{X}_j = \frac{1}{n} \sum_{i=1}^n X_{ji}$, 则

$$\sqrt{n}(\bar{X} - \mu) \rightsquigarrow N(0, \Sigma)$$

7.5.4 统计学习策略

有了模型的假设空间, 统计学习接着需要考虑的是按照什么样的准则学习或者选择最优的模型. 统计学习的目标在于从假设空间中选取最优模型.

首先引入损失函数与风险函数的概念. 损失函数度量模型一次预测的好坏, 风险函数度量平均意义上模型预测的好坏.

损失函数与风险函数

监督学习问题是在假设空间 \mathcal{F} 中选取模型 f 作为决策函数, 对于给定的输入 X , 由 $f(X)$ 给出相应的输出 Y , 这个输出的预测值 $f(X)$ 与真实值 Y 可能一致也可能不一致, 用一个损失函数 (loss function) 来度量预测错误的程度. 损失函数是 $f(X)$ 和 Y 的非负实值函数, 记作 $L(Y, f(X))$.

统计学习常用的损失函数有以下几种:

(1) 0-1 损失函数 (0-1 loss function)

$$L(Y, f(X)) = \begin{cases} 1 & Y = f(X) \\ 0 & Y \neq f(X) \end{cases}$$

(2) 平方损失函数 (quadratic loss function)

$$L(Y, f(X)) = (Y - f(X))^2$$

(3) 绝对损失函数 (absolute loss function)

$$L(Y, f(X)) = |Y - f(X)|$$

(4) 对数损失函数 (logarithmic loss function) 或者对数似然损失函数 (log likelihood loss function)

$$L(Y, f(X)) = -\log P(Y|X)$$

期望损失与经验损失

损失函数值越小，模型就越好。由于模型的输入、输出 (X, Y) 是随机变量，遵循联合分布 $P(X, Y)$ ，所以损失函数期望是

$$R_{exp}(f) = E_p[L(Y, f(X))] = \int L(y, f(x))P(x, y)dxdy$$

这是理论上模型 $f(X)$ 关于联合分布 $P(X, Y)$ 的平均意义下的损失，称为风险函数 (risk function) 或者期望损失 (expected loss)。

学习目标就是选择期望风险最小的模型。由于联合分布 $P(X, Y)$ 是未知的， $R_{exp}(f)$ 不能直接计算。实际上，如果知道联合分布 $P(X, Y)$ ，也就不需要学习了。正因为不知道联合概率分布，所以才需要进行学习。这样一来，一方面根据期望风险最小学习模型要用到联合分布，另一方面联合分布又是未知的，所以监督学习就成为一个病态问题。

给定训练数据集

$$T = (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

模型 $f(X)$ 关于训练数据集的平均损失称为经验风险 (empirical risk) 或者经验损失 (empirical loss)，记作 R_{emp} ：

$$R_{emp}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i))$$

期望风险 $R_{exp}(f)$ 是模型关于联合分布的期望损失，经验风险 $R_{emp}(f)$ 是模型关于训练样本的平均损失。根据大数定律，当样本容量 N 趋于无穷时，经验风险 $R_{emp}(f)$ 趋近于期望风险 $R_{exp}(f)$ 。所以一个很自然的想法是使用经验风险估计期望风险。

7.5.5 泛化误差上界

考虑二分类问题，已知训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ ， N 是样本容量， T 是从概率分布 $P(X, Y)$ 独立同分布产生的， $X \in \mathbb{R}^n, Y \in \{-1, +1\}$ 。假设空间是函数的有限集合 $\mathcal{F} = \{f_1, f_2, \dots, f_d\}$ ， d 是函数个数。设 f 是从 \mathcal{F} 中选取的函数。损失函数是 0-1 损失。关于 f 的期望风险和经验风险分别是

$$R(f) = E[L(Y, f(X))] \tag{7.10}$$

$$\hat{R}(f) = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) \tag{7.11}$$

经验风险最小化函数是

$$f_N = \arg \min_{f \in \mathcal{F}} \hat{R}(f) \quad (7.12)$$

f_N 依赖训练数据集的样本容量 N 。我们更关心 f_N 的泛化能力

$$R(f_N) = E[L(Y, f_N(X))] \quad (7.13)$$

接下来我们讨论从有限集合 $\mathcal{F} = \{f_1, f_2, \dots, f_d\}$ 中任意选出的函数 f 的泛化误差上界。

定理 7.5.7. [泛化误差上界] 对于二分类问题, 当假设空间是有限个函数的集合 $\mathcal{F} = \{f_1, f_2, \dots, f_d\}$ 时, 对于任意函数 $f \in \mathcal{F}$, 至少以概率 $1 - \delta$, $0 \leq \delta \leq 1$, 以下不等式成立:

$$R(f) \leq \hat{R}(f) + \epsilon(d, N, \delta) \quad (7.14)$$

其中

$$\epsilon(d, N, \delta) = \sqrt{\frac{1}{2N} (\log d + \log \frac{1}{\delta})} \quad (7.15)$$

不等式(7.14)左端 $R(f)$ 是泛化误差, 右端即泛化误差的上界。在泛化误差上界中, 第 1 项是训练误差, 训练误差越小, 泛化误差也越小。第 2 项 $\epsilon(d, N, \delta)$ 是 N 的单调递减函数, 当 N 趋于无穷时趋于 0; 同时它也是 $\sqrt{\log d}$ 阶的函数, 假设空间包含的函数越多, 其值越大。

证明. 在证明中要用到 Hoeffding 不等式, 叙述如下:

设 X_1, X_2, \dots, X_N 是独立随机变量, 且 $X_i \in [a_i, b_i]$, $i = 1, 2, \dots, N$; \bar{X} 是 X_1, X_2, \dots, X_N 的经验均值, 即 $\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$, 则对任意 $t > 0$, 以下不等式成立:

$$P[\bar{X} - E(\bar{X}) \geq t] \leq \exp\left(-\frac{2N^2t^2}{\sum_{i=1}^N (b_i - a_i)^2}\right) \quad (7.16)$$

$$P[\bar{X} - E(\bar{X}) \geq t] \leq \exp\left(-\frac{2N^2t^2}{\sum_{i=1}^N (b_i - a_i)^2}\right) \quad (7.17)$$

Hoeffding 不等式的证明省略。

对任意函数 $f \in \mathcal{F}$, $\hat{R}(f)$ 是 N 个独立的随机变量 $L(Y, f(X))$ 的样本均值, $R(f)$ 是随机变量 $L(Y, f(X))$ 的期望值。如果损失函数取值于区间 $[0, 1]$, 即对所有 i , $[a_i, b_i] = [0, 1]$, 那么由 Hoeffding 不等式(7.17)不难得知, 对 $\epsilon > 0$, 以下不等式成立:

$$P(R(f) - \hat{R}(f) \geq \epsilon) \leq \exp(-2N\epsilon^2) \quad (7.18)$$

由于 $\mathcal{F} = \{f_1, f_2, \dots, f_d\}$ 是一个有限集合, 故

$$\begin{aligned} P(\exists f \in \mathcal{F} : R(f) - \hat{R}(f) \geq \epsilon) &= P\left(\bigcup_{f \in \mathcal{F}} \{R(f) - \hat{R}(f) \geq \epsilon\}\right) \\ &\leq \sum_{f \in \mathcal{F}} P(R(f) - \hat{R}(f) \geq \epsilon) \\ &\leq d \exp(-2N\epsilon^2) \end{aligned} \quad (7.19)$$

或者等价的，对任意 $f \in \mathcal{F}$ ，有

$$P(R(f) - \hat{R}(f) < \epsilon) \geq 1 - d \exp(-2N\epsilon^2) \quad (7.20)$$

令

$$\delta = d \exp(-2N\epsilon^2) \quad (7.21)$$

则

$$P(R(f) < \hat{R}(f) + \epsilon) \geq 1 - \delta \quad (7.22)$$

即至少以概率 $1 - \delta$ 有 $R(f) < \hat{R}(f) + \epsilon$ ，其中 ϵ 由(7.21)得到，也就是(7.15)。 \square

7.6 随机过程简介

定义 7.6.1. 一个随机过程 $\{X_t : t \in T\}$ 是一个随机变量集合，通常写成 $X(t)$ 而不是 X_t ，其中变量 X_t 在一个被称作状态空间的集合 \mathcal{X} 里取值，集合 T 被称作指标集，通常可以视为时间。指标集可以是离散的 $T = \{0, 1, 2, \dots\}$ 或者连续的 $T = [0, \infty)$ 。

日常生活中的很多例子包括股票的波动、语音信号、身高的变化都可以看作是随机过程。常见的和时间相关的随机过程模型包括伯努利过程、随机游走、马尔可夫过程等，和空间相关的随机过程通常称为随机场。比如一张二维的图片，每个像素点（变量）通过空间的位置进行索引，这些像素就组成了一个随机过程。

例 7.6.1. (IID 观测) 一个 IID 随机变量序列可以写作 $\{X_t : t \in T\}$ ，其中 $T = \{1, 2, 3, \dots\}$ 。因此，一个 IID 随机变量序列就是一个随机过程。

例 7.6.2. (天气) 令 $\mathcal{X} = \{\text{晴}, \text{多云}\}$ 。一个典型的序列（依赖于你住在哪里）为

晴，晴，多云，晴，多云，多云…

该过程具有一个离散的状态空间和一个离散的指标集。

例 7.6.3. (经验分布函数) 令 $X_1, \dots, X_n \sim F$ 其中 F 为 $[0, 1]$ 上的某个 CDF。令

$$\hat{F}_n(t) = \frac{1}{n} \sum_{i=1}^n I(X_i \leq t)$$

为经验 CDF。对于任意固定值 t ， $\hat{F}_n(t)$ 是一个随机变量。但是整个经验 CDF

$$\hat{F}_n(t) : t \in [0, 1]$$

为一个具有连续状态空间和连续指标集的随机过程。

7.6.1 马尔科夫链

离散时间的马尔可夫过程也称为马尔可夫链。如果一个马尔可夫链的条件概率

$$P(X_{t+1} = s | X_t = s') = m_{ss'}$$

只和状态 s 和 s' 相关，和时间 t 无关，则称为时间同质的马尔科夫链，其中 $m_{ss'}$ 称为状态转移概率，如果状态空间大小 K 是有限的，状态转移概率可以用一个矩阵 $\mathbf{M} \in \mathbb{R}^{K \times K}$ 表示，称为状态转移矩阵，其中元素 m_{ij} 表示状态 s_i 转移到状态 s_j 的概率。

定义 7.6.2. 若

$$P(X_n = x | X_0, \dots, X_{n-1}) = P(X_n = x | X_{n-1})$$

对于所有的 n 和对所有的 $x \in \mathcal{X}$ 成立，则称过程 $\{X_n : n \in T\}$ 是一个马尔可夫链。

马尔可夫链可以用下面的 DAG 来表示：

$$X_0 \rightarrow X_1 \rightarrow X_2 \rightarrow \dots \rightarrow X_n \rightarrow \dots$$

每个变量具有单个母节点，即前一个观测。马尔可夫链理论非常丰富且复杂，主要涉及如下问题：

- 一个马尔可夫链何时“安定”为某种平稳态？
- 如何估计一个马尔可夫链的参数？
- 如何构造一个收敛到既定平稳分布的马尔可夫链和为什么想要那样做？

转移概率

一个马尔可夫链的重要的量为一个状态到另一个状态的概率。若 $P(X_{n+1} = j | X_n = i)$ 不随着时间而变化，则一个马尔可夫链是时齐的。因此，对于一个时齐马尔可夫链， $P(X_{n+1} | X_n = i) = P(X_1 = j | X_0 = i)$ 。下面只讨论时齐马尔可夫链

定义 7.6.3. 称

$$p_{ij} = P(X_{n+1} = j | X_n = i)$$

为转移概率，第 (ij) 个元素为 p_{ij} 的矩阵 P 称作转移矩阵。

注意到 P 具有两个性质 (i) $p_{ij} \geq 0$ 且 (ii) $\sum_i p_{ij} = 1$ 。每行可以看作一个概率密度函数。

例 7.6.4. (带吸收壁的随机游动) 令 $\mathcal{X} = \{1, \dots, N\}$ 。假设你正站在这些点中的一个点上，以 $P(\text{正面朝上}) = p$ 且 $P(\text{反面朝上}) = q = 1 - p$ 的概率投掷一枚硬币。若是正面朝上，向右走一

步，若是反面朝上，向左走一步。若你碰上某个终点，停止。转移矩阵为

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ q & 0 & p & 0 & \dots & 0 & 0 \\ 0 & q & 0 & p & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & p \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

例 7.6.5. 假设状态空间为 $\mathcal{X} = \{\text{晴}, \text{多云}\}$ ，则 X_1, X_2, \dots 表示一系列日子的天气。今天的天气还明显依赖于昨天的天气。它还可能依赖于前两天的天气，但是作为第一个近似，可以假设依赖性只倒退一天。在这种情况下，天气为一个马尔可夫链且一个典型的转移矩阵为

$$P = \begin{pmatrix} 0.4 & 0.6 \\ 0.8 & 0.2 \end{pmatrix}$$

例如，若今天是晴天，则明天有 60% 的可能性是多云。

定义 7.6.4. 令

$$p_{ij}(n) = P(X_{m+n} = j | X_m = i)$$

为在 n 步中从状态 i 转移到状态 j 的概率。令 P_n 表示第 (i, j) 个元素为 $p_{ij}(n)$ 的元素。这些被称为 n 步转移概率。

定理 7.6.1. (Chapman-Kolmogorov 方程) n 步概率满足

$$p_{ij}(m+n) = \sum_k p_{ik}(m)p_{kj}(n)$$

仔细观察上述方程，这只不过是矩阵乘法公式。因此证明了

$$\mathbf{P}_{m+n} = \mathbf{P}_m \mathbf{P}_n$$

由定义， $\mathbf{P}_1 = P$ 由上述定理， $\mathbf{P}_2 = \mathbf{P}_{1+1} = \mathbf{P}_1 \mathbf{P}_1 = \mathbf{P} \mathbf{P} = \mathbf{P}^2$ 按该方法继续下去，可以看到

$$\mathbf{P}_n = \mathbf{P}^n = \mathbf{P} \times \mathbf{P} \times \cdots \times \mathbf{P}$$

令 $\mu_n = (\mu_n(1), \dots, \mu_n(N))$ 为行向量，其中，

$$\mu_n(i) = P(X_n = i)$$

为该链在时刻 n 时处于状态 i 的边际概率。特别地， μ_0 被称作初始分布。为了模拟一个马尔可夫链，所要知道的就是 μ_0 和 \mathbf{P} 。模拟步骤应如下：

- 第一步产生 $X_0 \sim \mu_0$ ，因此 $P(X_0 = i) = \mu_0(i)$
- 第二步用 i 表示第一步的输出。产生 $X_1 \sim P$ 。换句话说， $P(X_1 = j | X_0 = i) = p_{ij}$
- 第三步假设第二步的输出为 j 。产生 $X_2 \sim P$ 。换句话说 $P(X_2 = k | X_1 = j) = p_{jk}$

- 继续下去。

理解 μ_n 的含义可能比较困难。想象模拟该链许多次，将所有的链在时刻 n 的输出收集起来。该直方图会近似于 μ_n

定理 7.6.2. 边际概率可由下式给出

$$\mu_n = \mu_0 P^n$$

状态分类

定义 7.6.5. i 到达 j （或 j 从 i 是可达的）若对于某个 n 有 $p_{ij}(n) > 0$ ，且记作 $i \rightarrow j$ 。若 $i \rightarrow j$ 且 $j \rightarrow i$ 则记作 $i \leftrightarrow j$ ，并且称 i 和 j 互通。

定理 7.6.3. 互通关系满足下面的性质

- $i \leftrightarrow i$
- 若 $i \leftrightarrow j$ 则 $j \leftrightarrow i$
- 若 $i \leftrightarrow j$ 且 $j \leftrightarrow k$ 则 $i \leftrightarrow k$
- 状态集 \mathcal{X} 可以写作不相交的类的并 $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \dots$ ，其中，两个状态之间互通当且仅当它们在同一个类中。

注：按互通关系是等价关系，可以把状态空间 \mathcal{X} 划分为若干个不相交的集合（或者说等价类），并称之为状态类。若两个状态互通，则这两个状态属于同一类。任意两个类或不相交或者相同。

定义 7.6.6. 设 C 为状态空间 \mathcal{X} 的一个子集，若对任意的 $i \in C$ 和 $j \notin C$ 有 $p_{ij} = 0$ 则称 C 为闭集。

注：若 C 为闭集，则表示自 C 内任意状态 i 出发，始终不能到达 C 以外的任何状态 j 。显然，整个状态空间构成一个闭集。

定义 7.6.7. 只含有单个状态的闭集称作为吸收态。

注：若状态空间含有吸收态，那么这个吸收态构成一个最小的闭集。

定义 7.6.8. 若除整个状态空间 \mathcal{X} 以外没有其它的闭集，则称此马氏链是不可约的。

如果闭集 C 的状态都是互通的，则称闭集 C 是不可约的。

例 7.6.6. 令 $\mathcal{X} = \{1, 2, 3, 4\}$ 且

$$P = \begin{pmatrix} 1/2 & 2/3 & 0 & 0 \\ 2/3 & 1/3 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

类为 $\{1, 2\}, \{3\}, \{4\}$ 。状态 4 为一个吸收态。

假设从状态 i 开始一个链。该链会返回状态 i 吗？若如此，称状态 i 为持久的或常返的。

定义 7.6.9. 状态 i 为持久的或常返的，若

$$P(X_n = i \text{ 对于某个 } n \geq 1 | X_0 = i) = 1$$

否则，状态 i 为瞬过的。

定理 7.6.4. 一个状态 i 为常返的当且仅当

$$\sum_n p_{ii}(n) = \infty$$

一个状态为瞬过的当且仅当

$$\sum_n p_{ii}(n) < \infty$$

定理 7.6.5. 关于常返性的事实

- 若状态 i 为常返的且 $i \leftrightarrow j$ ，则 j 是常返的。
- 若状态 i 为瞬过的且 $i \leftrightarrow j$ ，则 j 是瞬过的。
- 一个有限马尔可夫链必然至少有一个常返态。
- 一个有限的不可约马尔可夫链的状态都是常返的。

定理 7.6.6. (分解定理) 状态空间 \mathcal{X} 可以写成不相交集的并

$$\mathcal{X} = \mathcal{X}_T \cup \mathcal{X}_1 \cup \mathcal{X}_2 \dots$$

其中 \mathcal{X}_T 为瞬过态，且每个 X_i 为一个闭的，不可约的常返态集。

马尔可夫链的收敛性

为了讨论马尔可夫链的收敛性，需要一些定义。

定义 7.6.10. 假设 $X_0 = i$ 定义常返时间

$$T_{ij} = \min\{n > 0 : X_n = j\}$$

假设 X_n 可返回状态 i ，否则定义 $T_{ij} = \infty$ 一个常返态 i 的平均常返时间为

$$m_i = E(T_{ii}) = \sum_n n f_{ii}(n)$$

其中

$$f_{ij}(n) = P(X_1 \neq j, X_2 \neq j, \dots, X_{n-1} \neq j, X_n \neq j | X_0 = i)$$

若 $m_i = \infty$ 称一个常返态是零的，否则称之为非零的或正的。

定理 7.6.7. 若一个状态是零的且是常返的，则 $p_{ii}^n \rightarrow 0$

定理 7.6.8. 在一个有限的状态的马尔可夫链里，所有的常返态都是正的。

例 7.6.7. 考虑具有三个状态的马尔可夫链，其转移矩阵为

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

假设该链的初始状态为 1，那么将在时刻 3, 6, 9, … 到达状态 3，这是一个周期链的例子。

定义 7.6.11. 若 $p_{ii}(n) = 0$ ，其中 n 不能被 d 整除且 d 是满足该性质的最大整数，则称状态 i 的周期为 d 。因此， $d = \gcd\{n : p_{ii}(n) > 0\}$ ，其中 \gcd 的意思为“最大公约数”。若 $d(i) > 1$ ，则称该链的状态 i 是周期的。若 $d(i) = 1$ 是非周期的。周期为 1 的一个状态被称作非周期的。

定理 7.6.9. 若状态 i 具有周期 d 且 $i \leftrightarrow j$ ，则 j 也具有周期 d 。

定义 7.6.12. 若一个状态是常返的，非零的且周期的，则称这个状态 i 是遍历的。若其所有状态是遍历的，则称这一个链是遍历的。

令 $\pi = (\pi_i : i \in \mathcal{X})$ 为一个非负数向量，且分量和为 1。因此 π 可以视为一个概率密度函数。

定义 7.6.13. 若 $\pi = \pi\mathbf{P}$ ，则称 π 是一个平稳（或不变）分布。

这里给出直观的思路。 X_0 服从 π 分布并且假设 π 是一个平稳分布。现在根据马尔可夫链的转移概率来抽取 X_1 ，得到 X_1 的分布为 $\mu_1 = \mu_0\mathbf{P} = \pi\mathbf{P} = \pi$ 。 X_2 的分布为 $\pi\mathbf{P}^2 = (\pi\mathbf{P})\mathbf{P} = \pi\mathbf{P} = \pi$ ，如此继续下去，会看到 X_n 的分布为 $\pi\mathbf{P}^n = \pi$ 。换句话说，若该链在任何时候都具有分布 π ，则它将持续具有分布 π 。

定义 7.6.14. 称一个链具体极限分布 π ，若 $\mathbf{P}^n \rightarrow (\pi, \pi, \dots, \pi)^T$ 对于某个 π ，即 $\pi_j = \lim_{n \rightarrow \infty} P_{ij}^n$ 存在与 i 是独立的。

下面给出收敛性的主要定理，该定理表明一个遍历链收敛到它的平稳分布。而且，样本均值收敛到它的平稳分布下的理论期望。

定理 7.6.10. 一个不可约，遍历的马尔可夫链具有唯一的平稳分布 π 。极限分布存在且等于 π 。若 g 是任意一个有界函数，则以概率 1

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N g(X_n) \rightarrow E_\pi(g) = \sum_j g(j)\pi_j$$

定义 7.6.15. 若

$$\pi_i p_{ij} = p_{ij}\pi_j$$

则 π 满足细致平衡

细致平衡保证了 π 是一个平稳分布。

定理 7.6.11. 若 π 满足细致平衡，则 π 是一个平稳分布。

注意仅仅因为一个链有一个平稳分布并不意味着它收敛。

例 7.6.8. 令

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

令 $\pi = 1/3, 1/3, 1/3$, 则 $\pi\mathbf{P} = \pi$ 所以 π 是一个平稳分布。若该链是从分布 π 开始的，它将停留在该分布里。想象模拟许多链且在每个时刻 n 去验证其边际分布。它将永远为均匀分布 π 但是该链没有极限。它将继续循环下去。

例 7.6.9. 令 $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$, 令

$$\mathbf{P} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 1/4 & 3/4 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 \\ 1/4 & 0 & 1/4 & 1/4 & 0 & 1/4 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \end{pmatrix}$$

则 $C_1 = \{1, 2\}$ 且 $C_2 = \{5, 6\}$ 是不可约的闭集。状态 3 和状态 4 是暂留的因为路径为 $3 \rightarrow 4 \rightarrow 6$ 且一旦到达状态 6 就不能返回 3 或 4. 因为 $p_{ii}(1) > 0$, 所有的状态都是非周期的, 总之 3 和 4 是暂留的, 而 1, 2, 5 和 6 是遍历的。

这里简述一种叫马尔可夫链蒙特卡罗 (MCMC) 的模拟方法的基本思想。

例 7.6.10. (马尔可夫链蒙特卡罗)

令 $f(x)$ 为实轴上的一个概率密度函数且假设 $f(x) = cg(x)$ 其中 $g(x)$ 是一个已知函数且 $c > 0$ 是未知的。原则上可以计算出 c , 因为 $\int f(x)dx = 1$ 意味着 $c = 1 / \int g(x)dx$ 。然而, 计算该积分可能行不通, 而且 c 对下面的计算也没有必要。令 X_0 为一个任意开始值。给定 X_0, \dots, X_i 按下面的方法产生 X_{i+1} 。首先, 选取 $W \sim N(X_i, b^2)$ 其中 $b > 0$ 是一个固定的常数。令

$$r = \min \left\{ \frac{g(W)}{g(X_i)}, 1 \right\}$$

选取 $U \sim U(0, 1)$ 且设定

$$X_{i+1} = \begin{cases} W, & U < r \\ X_i & U \geq r \end{cases}$$

在弱条件下, X_0, X_1, \dots 是以一个遍历的马尔可夫链且平稳分布为 f 。因此, 可以将选取出来的变量看作来自 f 的一个样本。

7.6.2 高斯过程

高斯过程也是一种应用广泛的随机过程模型。假设有一组连续随机变量 X_0, X_1, \dots, X_T , 如果由这组随机变量构成的任一有限集合

$$X_{t_1, \dots, t_N} = [X_{t_1}, \dots, X_{t_N}]^\top, \quad 1 \leq N \leq T$$

都服从一个多元正态分布, 那么这组随机变量为一个随机过程, 高斯过程也可以定义为: 如果 X_{t_1, \dots, t_N} 的任一线性组合都服从一元正态分布, 那么这组随机变量为一个随机过程。

高斯过程回归

高斯过程回归是利用高斯过程来对一个函数分布进行建模。和机器学习中参数化建模(比如贝叶斯线性回归)相比, 高斯过程是一种非参数模型, 可以拟合一个黑盒函数, 并给出拟合结果的置信度。

假设一个未知函数 $f(\mathbf{x})$ 服从高斯函数, 且为平滑函数, 如果两个样本 $\mathbf{x}_1, \mathbf{x}_2$ 比较接近, 那么对应的 $f(\mathbf{x}_1), f(\mathbf{x}_2)$ 也比较接近, 假设从函数 $f(\mathbf{x})$ 中采样有限个样本 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$, 这 N 个点服从一个多元正态分布,

$$[f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_N)]^\top \sim N(\boldsymbol{\mu}(X), \mathbf{K}(X, X))$$

其中 $\boldsymbol{\mu}(X) = [\boldsymbol{\mu}(\mathbf{x}_1), \boldsymbol{\mu}(\mathbf{x}_2), \dots, \boldsymbol{\mu}(\mathbf{x}_N)]^\top$ 是均值向量, $\mathbf{K}(X, X) = [k(\mathbf{x}_i, \mathbf{x}_j)]_{N \times N}$ 是协方差矩阵, $k(\mathbf{x}_i, \mathbf{x}_j)$ 为核函数, 可以衡量两个样本的相似度。

在高斯过程回归中, 一个常用的核函数是平方指数函数

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2l^2}\right)$$

其中 l 为超参数, 当 \mathbf{x}_i 和 \mathbf{x}_j 越接近, 其核函数的值越大, 表明 $f(\mathbf{x}_i)$ 和 $f(\mathbf{x}_j)$ 越相关。

假设 $f(\mathbf{x})$ 的一组带噪声的观测值为 $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$, 其中 $y_n \sim N(f(x_n), \sigma^2)$ 为 $f(x_n)$ 的观测值, 服从正态分布, σ 为噪声方差。

对于一个新的样本点 \mathbf{x}^* , 我们希望预测 $f(\mathbf{x}^*)$ 观测值 y^* 。令向量 $\mathbf{y} = [y_1, y_2, \dots, y_N]^\top$ 为已有的观测值, 根据高斯过程的假设, $[\mathbf{y}; y^*]$ 满足

$$\begin{bmatrix} \mathbf{y} \\ y^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}(X) \\ \boldsymbol{\mu}(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(X, X) + \sigma^2 \mathbf{I} & \mathbf{K}(\mathbf{x}^*, X)^\top \\ \mathbf{K}(\mathbf{x}^*, X) & k(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right)$$

其中 $\mathbf{K}(\mathbf{x}^*, X) = [k(\mathbf{x}^*, \mathbf{x}_1), \dots, k(\mathbf{x}^*, \mathbf{x}_n)]$

根据上面的联合分布, y^* 的后验分布为

$$p(y^* | \mathbf{X}, \mathbf{y}) = \mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\sigma}^2)$$

其中均值 $\hat{\boldsymbol{\mu}}$ 和方差 $\hat{\sigma}$ 为

$$\hat{\boldsymbol{\mu}} = \mathbf{K}(\mathbf{x}^*, X) (\mathbf{K}(X, X) + \sigma^2 \mathbf{I})^{-1} (\mathbf{y} - \boldsymbol{\mu}(X)) + \boldsymbol{\mu}(\mathbf{x}^*)$$

$$\hat{\sigma}^2 = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{K}(\mathbf{x}^*, X) (\mathbf{K}(X, X) + \sigma^2 \mathbf{I})^{-1} \mathbf{K}(\mathbf{x}^*, X)^\top$$

从公式可以看出, 均值函数 $\boldsymbol{\mu}(\mathbf{x})$ 可以近似地互相抵消, 在实际应用中, 一般假设 $\boldsymbol{\mu}(\mathbf{x}) = 0$, 均值 $\hat{\boldsymbol{\mu}}$ 可以将简化为

$$\hat{\boldsymbol{\mu}} = \mathbf{K}(\mathbf{x}^*, X) (\mathbf{K}(X, X) + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$$

高斯过程回归可以认为是一种有效地贝叶斯优化方法, 广泛地应用于机器学习中。

7.7 阅读材料

机器学习中的概率模型 Bishop (2006) ; Murphy (2012) 为用户提供了一种以原则方式捕捉数据和预测模型的不确定性的方法。Ghahramani (2015) 简要回顾了机器学习中的概率模型。本章有时候会出现这种情况，Grinstead 和 Snell (1997) 提供了一种适合自学的更轻松的表达方式。对概率的更多哲学方面感兴趣的读者应该考虑 Hacking (2001)，而 Downey (2014) 提出了更多的软件工程方法。

给定概率模型，我们可能足够幸运能够分析地计算感兴趣的参数。然而，一般而言，分析方法很少用，而使用的是诸如采样 (Brooks 等人, 2011) 和变分推理 (Blei 等人, 2017) 之类的计算方法。具有讽刺意味的是，最近对神经网络兴趣的激增导致了对概率模型的更广泛的认识。例如，流量标准化的想法 (Rezende 和 Mohamed, 2015) 依赖于变量来改变随机变量。应用于神经网络的变分推断方法的概述在 Goodfellow 等人 (2016) 的第 16 至 20 章中描述。

对概率论细节感兴趣的更多读者有很多选择 (Jacod 和 Protter, 2004; Jaynes, 2003; Mackay, 2003)，包括一些非常技术性的讨论 (Dudley, 2002; Shirayev, 1984; Lehmann 和 Casella, 1998; Bickel 和 Doksum, 2006)。我们通过对测量理论问题进行掩饰来回避大部分困难 (Billingsley, 1995; Pollard, 2002)，并不进行构造地假设我们有实数，以及定义实数集的方法以及它们的适当频率发生。由于机器学习允许我们在移动复杂类型的数据上模拟移动复杂的分布，因此概率机器学习模型的开发者必须理解这些更多的技术方面。具有概率建模焦点的机器学习书包括 Mackay (2003) ;Bishop (2006) ;Murphy (2012) ;Barber (2012 年) ; Rasmussen 和 Williams (2006 年)。

习题

习题 7.1. (1) 设 A, B, C 是三个事件，且 $P(A) = P(B) = P(C) = 1/4$, $P(AB) = P(BC) = 0$, $P(AC) = 1/8$ ，求 A, B, C 至少有一个发生的概率。

(2) 已知 $P(A) = 1/2 \cdot P(B) = 1/3$, $P(C) = 1/5$, $P(AB) = 1/10$, $P(AC) = 1/15$, $P(BC) = 1/20$, $P(ABC) = 1/30$ ，求 $A \cup B$, \overline{AB} , $A \cup B \cup C$, \overline{ABC} , \overline{ABC} , $\overline{AB} \cup C$ 的概率

习题 7.2. (1) 已知 $P(\overline{A}) = 0.3$, $P(B) = 0.4$, $P(AB) = 0.5$ ，求条件概率 $P(B|A \cup \overline{B})$

(2) 已知 $P(A) = 1/4$, $P(B|A) = 1/3$, $P(A|B) = 1/2$ ，求 $P(A \cup B)$

习题 7.3. 设事件 A, B 的概率均大于零，说明以下的叙述 (1) 必然对.(2) 必然错. (3) 可能对. 并说明理由。

(1) 若 A 与 B 互不相容，则它们相互独立。

(2) 若 A 与 B 相互独立，则它们互不相容。

(3) $P(A) = P(B) = 0.6$, 且 A, B 互不相容。

(4) $P(A) = P(B) = 0.6$, 且 A, B 相互独立。

习题 7.4. (1) 袋中装有 5 只球, 编号为 1,2,3,4,5。在袋中同时取 3 只, 以 X 表示去除的 3 只中的最大号码, 写出随机变量 X 的分布律。

(2) 将一颗骰子抛掷两次, 以 X 表示两次中得到的小的点数, 试求 X 的分布律。

习题 7.5. 设随机变量 X 的分布函数为

$$F_x(x) = \begin{cases} 0, & x < 1 \\ \ln x, & 1 \leq x < e \\ 1, & x \geq e \end{cases}$$

(1) 求 $P\{X < 2\}, P\{0 < X \leq 3\}, P\{2 < X < 5/2\}$

(2) 求概率密度 $f_x(x)$

习题 7.6. 设随机变量 X 的概率密度为

$$f(x) = \begin{cases} \frac{2x}{\pi^2}, & x > 0 \\ 0, & \text{其他} \end{cases}$$

求 $Y = \sin X$ 的概率密度。

习题 7.7. 设 $X \sim N(0, 1)$.

(1) 求 $Y = e^x$ 的概率密度。

(2) 求 $Y = 2X^2 + 1$ 的概率密度。

(3) 求 $Y = |X|$ 的概率密度。

习题 7.8. 设二维随机变量 (X, Y) 的概率密度为

$$f(x, y) = \begin{cases} e^{-y}, & 0 < x < y \\ 0, & \text{其他} \end{cases}$$

(1) 确定常数 c 。

(2) 求边缘概率密度。

习题 7.9. 设二维随机变量 (X, Y) 的概率密度为

$$f(x, y) = \begin{cases} 1, & |y| < x, 0 < x < 1 \\ 0, & \square \square \end{cases}$$

求条件概率密度 $f_{Y|X}(y|x), f_{X|Y}(x|y)$

习题 7.10. 设随机变量 X, Y 的联合密度为

$$f(x, y) = \begin{cases} \frac{1}{y} e^{-(y+x/y)}, & x > 0, y > 0 \\ 0, & \text{其他} \end{cases}$$

求 $E(X), E(Y), E(XY)$

习题 7.11. 计算样在进行加法时, 将每个加数舍人最靠近它的整数, 设所有舍人误差相互独立且在 $(-0.5, 0.5)$ 上服从均匀分布.

(1) 将 1500 个数相加, 问误差总和的绝对值超过 15 的概率是多少?

(2) 最多可有几个数相加使得误差总和的绝对值小于 10 的概率不小于 0.90?

习题 7.12. (1) 设 A, B, C 是三个事件, 且 $P(A) = P(B) = P(C) = 1/4, P(AB) = P(BC) = 0, P(AC) = 1/8$, 求 A, B, C 至少有一个发生的概率。

(2) 已知 $P(A) = 1/2 \cdot P(B) = 1/3, P(C) = 1/5, P(AB) = 1/10, P(AC) = 1/15, P(BC) = 1/20, P(ABC) = 1/30$, 求 $A \cup B, \overline{AB}, A \cup B \cup C, \overline{ABC}, \overline{AC}, \overline{AB} \cup C$ 的概率

习题 7.13. (1) 已知 $P(\overline{A}) = 0.3, P(B) = 0.4, P(AB) = 0.5$, 求条件概率 $P(B|A \cup \overline{B})$ (2) 已知 $P(A) = 1/4, P(B|A) = 1/3, P(A|B) = 1/2$, 求 $P(A \cup B)$

传外勿请稿草

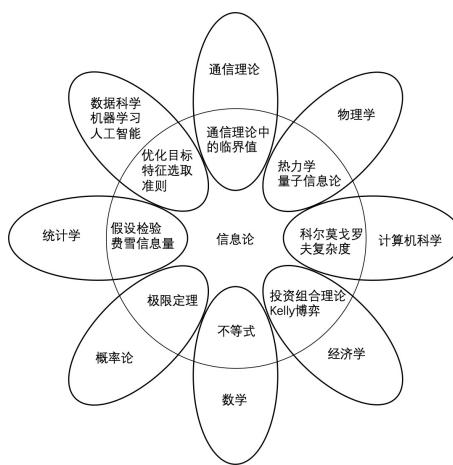
插 情勿外传

第八章 信息论基础

信息论是通信的基础理论之一。信息论创始人香农对信息的定义为信息是对事物运动状态或存在方式的不确定性的表示。信息论解答了通信理论中的两个基本问题：

1. 数据压缩的临界值；
2. 通信传输速率的临界值。

然而，信息论的影响力远不止于此，它在很多学科，如统计物理（热力学），计算机科学（Kolmogorov 复杂度或算法复杂度），统计推断（奥卡姆剃刀：最简洁的解释最佳），概率和统计等学科（关于最优化假设检验和估计的误差指数）都具有奠基性的贡献。



数据承载着信息，有些信息可以直接从数据中获得，有些信息需要对数据进行一定的运算处理后才能获得。比如一张表示猫的图片，人类可以从图片上看出这是一只猫，计算机却需要将图片经过模型运算后才能知道图片上表示的内容是猫，从而获得了图片表示内容的信息。

在机器学习中一般有两类学习准则：一类是如经验风险、经验误差、经验损失的经验函数。另一类是如信息熵、交叉熵、相对熵、互信息的基于信息论中熵的函数。David MacKay 认为信息论和机器学习就是一枚硬币的两面。信息论指导机器学习中的很多算法的设计和改进，比如

用交叉熵损失作为损失函数，利用互信息进行特征选择等。以信息理论为基础的机器学习在理论上更具有优势：我们可以将机器学习或深度模型中的编码解码模型看做是一个通信系统，输入为信源，这样信息论中的一些度量也可以作为学习算法的度量。信息瓶颈理论认为深度学习模型具有特征拟合和特征压缩两个阶段，用互信息评价特征的保真度和压缩率。

我们把发出数据的一端称作信源，从信源发出一张图片，这个图片是什么样的我们是不知道的。因此，图片数据本身就包含着信息。对于特定一张图片，在将图片数据经过分类模型处理前，计算机是不知道图片表示的物体类别的。当图片经过模型处理，确定了图片所属的类别，也就消除了不确定性，从而获得了图片所属类别这一信息。Watanabe 认为学习就是一个熵减的过程。

概率是研究不确定性的工具，我们可以基于概率对信息量进行度量。

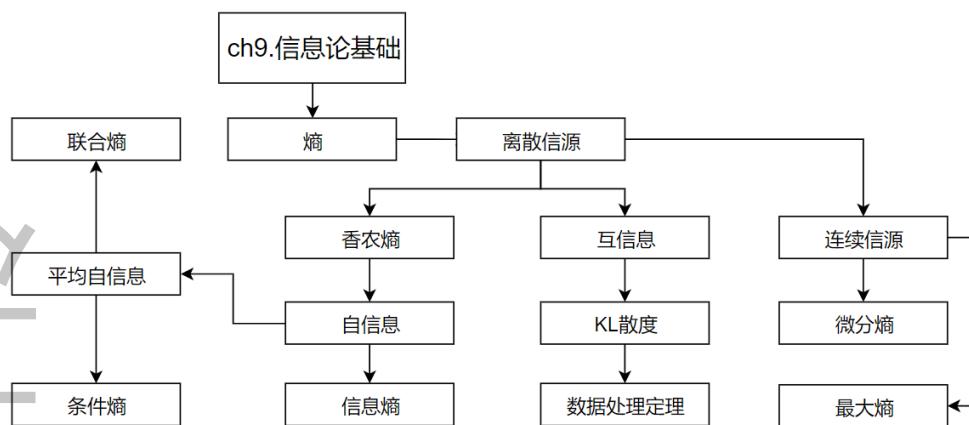


图 8.1: 本章导图

8.1 熵、相对熵和互信息

信息论主要研究的是对一个信号包含信息的多少进行量化。这种量化应该符合如下的直觉：

- 事件的信息量应当是事件发生概率的函数。
- 小概率事件，不确定性大，一旦出现使人感到意外，因此产生的信息量就大，特别是几乎不可能出现的事件一旦出现，必然产生极大的信息量；
- 大概率事件，是预料之中的事件，不确定性小，即使发生，也没什么信息量，特别是概率为 1 的确定事件发生以后，不会给人以任何信息量。

信息是个相当宽泛的概念，很难用一个简单的定义将其完全准确地把握，然而对于任何一个概率分布可以定一个称为熵的量。它具有许多特性符合度量信息的直观要求，这个概念可以

推广到互信息，互信息是一种测度，用来度量一个随机变量包含另一个随机变量的信息量。熵恰好变成了一个随机变量的自信息。相对熵是个更广泛的量，他是刻画两个概率分布之间的距离的一种度量，而互信息又是它的特殊情形，以上所有的这些量密切相关，存在许多简单的共性。

8.1.1 自信息

信源发出的消息(事件)具有不确定性。不确定小的，即发生概率大的消息含的信息量小。不确定性大的，即发生概率小的消息含的信息量大。如果两个消息是毫无关联的，即发生的概率是相互独立的，那么通过这两个消息获得的信息量应当是两个消息各自信息量的和。

因此随机事件的自信息量 $I(x_i)$ 是该事件发生概率 $I(x_i)$ 的函数，并且 $I(x_i)$ 应该满足以下公理化条件：

(1) $I(x_i)$ 是 $P(x_i)$ 的严格递减函数. 当 $P(x_1) < P(x_2)$ 时, $I(x_1) > I(x_2)$, 概率越小, 事件发生的不确定性越大, 事件发生以后所包含的自信息量越大.

(2) 极限情况下, 当 $P(x_i) = 0$ 时, $I(x_i) \rightarrow \infty$; 当 $P(x_i) = 1$ 时, $I(x_i) = 0$.

(3) 从直观概念上讲, 由两个相对独立的不同的消息所提供的信息量应等于它们分别提供的信息量之和, 即自信息量满足可加性.

可以证明, 满足以上公理化条件的函数形式是对数形式.

定义 8.1.1. 随机事件的自信息量定义为该事件发生概率的对数的负值. 设事件 x_i 的概率为 $P(x_i)$, 则它的自信息量定义为

$$I(x_i) = -\log p(x_i) = \log \frac{1}{p(x_i)}$$

$I(x_i)$ 代表两种含义: 在事件 x_i 发生以前, 等于事件 x_i 发生的不确定性的大小; 在事件 x_i 发生以后, 表示事件 x_i 所含有或所能提供的信息量。

自信息量的单位与所用对数的底有关. (1) 通常取对数的底为 2, 信息量的单位为比特 (bit, binary unit). 当 $p(x_i) = 1/2$ 时, $I(x_i) = 1$ bit, 即概率等于 $1/2$ 的事件具有 1 bit 的自信息量. 例如, 一枚均匀硬币的任何一种抛掷结果均含有 1 bit 的信息量. 比特是信息论中最常用的信息量单位, 当取对数的底为 2 时, 2 常省略. 注意: 计算机术语中 bit 是位的单位 (bit, binary digit), 与信息量单位不同, 但有联系, 1 位的二进制数字最大能提供 1 bit 的信息量.

(2) 若取自然对数 (以 e 为底), 自信息量的单位为奈特 (nat, natural unit). 理论推导中或用于连续信源时用以 e 为底的对数比较方便.

$$1 \text{ nat} = \log_2 e \text{ bit} = 1.443 \text{ bit}$$

(3) 工程上用以 10 为底较方便. 若以 10 为对数底, 则自信息量的单位为哈特莱 (Hartley), 用来纪念哈特莱首先提出用对数来度量信息.

$$1 \text{ Hartley} = \log_2 10 \text{ bit} = 3.322 \text{ bit}$$

(4) 如果取以 r 为底的对数 ($r > 1$) , 则 $I(x_i) = -\log r p(x_i)$, r 进制单位

$$1 r \text{ 进制单位} = \log_2 r \text{ bit}$$

例 8.1.1. (1) 英文字母中 “ a ” 出现的概率为 0.064, “ c ” 出现的概率为 0.022, 分别计算它们的自信息量.

(2) 假定前后字母出现是互相独立的, 计算 “ ac ” 的自信息量.

(3) 假定前后字母出现不是互相独立的, 当 “ a ” 出现以后, “ c ” 出现的概率为 0.04, 计算 “ a ” 出现以后, “ c ” 出现的自信息量.

解. (1) $I(a) = -\log_2 0.064 = 3.96 \text{ bit}$

$$I(c) = -\log_2 0.022 = 5.51 \text{ bit}$$

(2) 由于前后字母出现是互相独立的, “ ac ” 出现的概率为 0.064×0.022 , 所以 $I(ac) = -\log_2(0.064 \times 0.022) = -(\log_2 0.064 + \log_2 0.022) = I(a) + I(c) = 9.47 \text{ bit}$ 即两个相对独立的事件的自信息量满足可加性, 也就是由两个相对独立的事件的积事件所提供的信息量应等于它们分别提供的信息量之和.

(3) “ a ” 出现的条件下 “ c ” 出现的概率变大, 它的不确定性变小.

$$I(c|a) = -\log_2 0.04 = 4.64 \text{ bit}$$

8.1.2 熵及其性质

自信息量是信源发出某一具体消息所含有的信息量, 发出的消息不同它的自信息量就不同, 所以有信息量本身为随机变量, 不能用来表征整个信源的不确定度. 我们用平均自信息量来表征整个信源的不确定度. 平均自信息量又称为信息熵、信源熵, 简称熵.

因为信源具有不确定性, 所以把信源用随机变量来表示, 用随机变量的概率分布来描述信源的不确定性. 通常把一个随机变量的所有可能的取值和这些取值对应的概率 $[X, P(X)]$ 称为它的概率空间.

假设随机变量 X 有 q 个可能的取值 $x_i, i = 1, 2, \dots, q$, 各种取值出现的概率为 $p(x_i), i = 1, 2, \dots, q$, 它的概率空间表示为

$$\begin{pmatrix} X \\ P(X) \end{pmatrix} = \begin{pmatrix} X = x_1 & \cdots & X = x_i & \cdots & X = x_q \\ p(x_1) & \cdots & p(x_i) & \cdots & p(x_q) \end{pmatrix}$$

这里要注意, $p(x_i)$ 满足概率空间的基本特性: 非负性 $0 \leq p(x_i) \leq 1$ 和完备性 $\sum_{i=1}^q p(x_i) = 1$.

定义 8.1.2. 随机变量 X 的每一个可能取值的自信息 $I(x_i)$ 的统计平均值定义为随机变量 X 的信息熵.

$$H(X) = \mathbb{E}[I(x_i)] = -\sum_{i=1}^q p(x_i) \log p(x_i)$$

这里 q 为 X 的所有可能取值的个数。

熵的单位也是与所取的对数底有关, 根据所取的对数底不同, 可以是比特/ 符号、奈特/ 符号、哈特莱/ 符号或者是 r 进制单位/ 符号, 通常用比特/ 符号为单位.

例 8.1.2. 假设随机变量 X 的概率分布为 $p(x_i) = 2^{-i}, i = 1, 2, 3, \dots$, 求 $H(X)$.

解.

$$H(X) = \sum_{i=1}^{\infty} 2^{-i} \log_2 \frac{1}{2^{-i}} = \sum_{i=1}^{\infty} i 2^{-i} = 2 \text{ 比特/符号}$$

熵编码

信息论的研究目标之一是如何用最少的编码表示传递信息. 假设我们要传递一段文本信息, 这段文本中包含的符号都来自于一个字母表 A , 我们就需要对字母表 A 中的每个符号进行编码. 以二进制编码为例, 我们常用的 ASCII 码就是用固定的 8bits 来编码每个字母. 但这种固定长度的编码方案不是最优的. 一种高效的编码原则是字母的出现概率越高, 其编码长度越短. 比如对字母 a, b, c 分别编码为 0, 10, 110.

给定一串要传输的文本信息, 其中字母 x 的出现概率为 $p(x)$, 其最佳编码长度为 $-\log_2 p(x)$, 整段文本的平均编码长度为 $-\sum_x p(x) \log_2 p(x)$, 即底为 2 的熵. 在对分布 $p(x)$ 的符号进行编码时, 熵 $H(p)$ 也是理论上最优的平均编码长度, 这种编码方式称为熵编码 (Entropy Encoding). 由于每个符号的自信息通常都不是整数, 因此在实际编码中很难达到理论上的最优值. 霍夫曼编码 (Huffman Coding) 和算术编码 (Arithmetic Coding) 是两种最常见的熵编码技术.

熵函数的性质

信息熵 $H(X)$ 是随机变量 X 的概率分布的函数, 所以又称为熵函数. 如果把概率分布 $p(x_i), i = 1, 2, \dots, q$, 记为 p_1, p_2, \dots, p_q , 则熵函数又可以写成概率向量 $\mathbf{p} = (p_1, p_2, \dots, p_q)$ 的函数形式, 记为 $H(\mathbf{p})$.

$$H(X) = -\sum_{i=1}^q p(x_i) \log p(x_i) = H(p_1, p_2, \dots, p_q) = H(\mathbf{p})$$

因为概率空间的完备性, 即 $\sum_{i=1}^q p(x_i) = 1$, 所以 $H(\mathbf{p})$ 是 $(q-1)$ 元函数. 当 $q = 2$ 时, 因为 $p_1 + p_2 = 1$, 若令其中一个概率为 p , 则另一个概率为 $(1-p)$, 熵函数可以写成 $H(\mathbf{p})$.

熵函数 $H(\mathbf{p})$ 具有以下性质:

1. 对称性

$$H(p_1, p_2, \dots, p_q) = H(p_2, p_1, \dots, p_q) = \dots = H(p_q, p_1, \dots, p_q - 1)$$

也就是说概率向量 $\mathbf{p} = (p_1, p_2, \dots, p_q)$ 各分量的次序可以任意变更, 熵值不变. 对称性说明熵函数仅与信源的总体统计特性有关.

2. 确定性

$$H(1, 0) = H(1, 0, 0) = H(1, 0, 0, 0) = \cdots = H(1, 0, \dots, 0) = 0$$

在概率向量 $p = (p_1, p_2, \dots, p_q)$ 中, 只要有一个分量为 1, 其他分量必为 0, 它们对熵的贡献均为 0, 因此熵等于 0, 也就是说确定信源的平均不确定度为 0。

3. 非负性

$$H(p) = H(p_1, p_2, \dots, p_q) \geq 0$$

对确定信源, 等号成立.

信源熵是自信息的数学期望, 自信息是非负值, 所以信源熵必定是非负的. 离散信源熵才有这种非负性, 以后会讲到连续信源的相对熵则可能出现负值.

4. 扩展性

$$\lim_{\epsilon \rightarrow 0} H_{q+1}(p_1, p_2, \dots, p_q - \epsilon, \epsilon) = H_q(p_1, p_2, \dots, p_q)$$

这是因为 $\lim_{\epsilon \rightarrow 0} \epsilon \log \epsilon = 0$

这个性质的含义是: 增加一个基本不会出现的小概率事件, 信源的熵保持不变. 虽然小概率事件出现给予收信者的信息量很大, 但在熵的计算中, 它占的比重很小, 可以忽略不计, 这也是熵的总体平均性的体现.

5. 连续性

$$\lim_{\epsilon \rightarrow 0} H(p_1, p_2, \dots, p_{q-1} - \epsilon, p_q + \epsilon) = H(p_1, p_2, \dots, p_q)$$

即信源概率空间中概率分量的微小波动, 不会引起熵的变化.

6. 递增性

$$H(p_1, p_2, \dots, p_{n-1}, q_1, q_2, \dots, q_m) = H(p_1, p_2, \dots, p_n) + p_n H\left(\frac{q_1}{p_n}, \frac{q_2}{p_n}, \dots, \frac{q_m}{p_n}\right)$$

这个性质表明, 假如有一个信源的 n 个元素的概率分布为 p_1, p_2, \dots, p_n , 其中某个元素 x_n 又被划分成 m 个元素, 这 m 个元素的概率之和等于元素 x_n 的概率, 这样得到的新信源的熵增加了一项, 增加的一项是由于划分产生的不确定性.

例 8.1.3. 利用递推性计算 $H(1/2, 1/8, 1/8, 1/8, 1/8)$.

解.

$$\begin{aligned} & H(1/2, 1/8, 1/8, 1/8, 1/8) \\ &= H(1/2, 1/2) + \frac{1}{2} \times H(1/4, 1/4, 1/4, 1/4) \\ &= 1 + \frac{1}{2} \times 2 \\ &= 2 \text{ 比特/符号} \end{aligned}$$

7. 极值性

$$H(p_1, p_2, \dots, p_n) \leq H\left(\frac{1}{n}, \dots, \frac{1}{n}\right) = \log n \quad (8.1)$$

式中 n 是随机变量 X 的可能取值的个数.

极值性表明离散信源中各消息等概率出现时熵最大, 这就是最大离散熵定理. 连续信源的最大熵则还与约束条件有关.

极值性可看成

$$H(p_1, p_2, \dots, p_n) \leq -\sum_{i=1}^n p_i \log_2 q_i \quad (8.2)$$

的特例情况. 下面先证明式(8.2)

证明. 利用 Jensen 不等式, 有

$$\begin{aligned} & H(p_1, p_2, \dots, p_n) + \sum_{i=1}^n p_i \log_2 q_i \\ &= -\sum_{i=1}^n p_i \log_2 p_i + \sum_{i=1}^n p_i \log_2 q_i = \sum_{i=1}^n p_i \log_2 \frac{q_i}{p_i} \leq \log_2 \sum_{i=1}^n \left(p_i \cdot \frac{q_i}{p_i} \right) = 0 \end{aligned}$$

当 $\frac{q_i}{p_i} = 1$, $i = 1, 2, \dots, n$ 时, 等号成立. 证毕. \square

式(8.2)表明任一随机变量的概率分布 p_i , 对其他概率分布 q_i 定义的自信息 $-\log_2 q_i$ 的数学期望, 必不小于概率分布 p_i 本身定义的熵 $H(p_1, p_2, \dots, p_n)$.

如果取 $q_i = \frac{1}{n}$, $i = 1, 2, \dots, n$ 时, 由式(8.2)就得到

$$H(p_1, p_2, \dots, p_n) \leq H\left(\frac{1}{n}, \dots, \frac{1}{n}\right) = \log n$$

当 $p_i = \frac{1}{n}$, $i = 1, 2, \dots, n$ 时, 等号成立.

当信源输出的消息等概分布时, 信源熵达到最大值——1 比特/符号. 因此当二元数字是由等概的二元信源输出时, 每个二元数字提供 1 bit 的信息量, 否则, 每个二元数字提供的信息量小于 1 bit. 这就是信息量的单位比特和计算机术语中位的单位比特的关系.

8. 上凸性

$H(\mathbf{p})$ 是严格的上凸函数, 设 $\mathbf{p} = (p_1, p_2, \dots, p_q)$, $\mathbf{p}' = (p'_1, p'_2, \dots, p'_q)$, $\sum_{i=1}^q p_i = 1$, $\sum_{i=1}^q p'_i = 1$, 则对于任意小于 1 的正数 α , $0 < \alpha < 1$, 以下不等式成立:

$$H[\alpha\mathbf{p} + (1 - \alpha)\mathbf{p}'] > \alpha H(\mathbf{p}) + (1 - \alpha)H(\mathbf{p}')$$

证明. 因为 $0 \leq p_i \leq 1$, $0 \leq p'_i \leq 1$, 且 $0 < \alpha < 1$, 所以 $0 \leq \alpha p_i + (1 - \alpha)p'_i \leq 1$, 并且 $\sum_{i=1}^q (\alpha p_i + (1 - \alpha)p'_i) = 1$, 所以 $\alpha\mathbf{p} + (1 - \alpha)\mathbf{p}'$ 可以看作是一种新的概率分布.

$$\begin{aligned}
 H(\alpha p + (1 - \alpha)p') &= -\sum_{i=1}^q (\alpha p_i + (1 - \alpha)p'_i) \log_2 (\alpha p_i + (1 - \alpha)p'_i) \\
 &= -\alpha \sum_{i=1}^q p_i \log (\alpha p_i + (1 - \alpha)p'_i) - (1 - \alpha) \sum_{i=1}^q p'_i \log_2 (\alpha p_i + (1 - \alpha)p'_i) \\
 &\geq -\alpha \sum_{i=1}^q p_i \log_2 p_i - (1 - \alpha) \sum_{i=1}^q p'_i \log_2 p'_i \\
 &\geq \alpha H(p) + (1 - \alpha) H(p')
 \end{aligned}$$

当 $p \neq p'$ 时, 有 $\frac{\alpha p_i + (1 - \alpha)p'_i}{p_i} \neq 1$, 式(8.2)中等号不成立, 所以

$$H(\alpha p + (1 - \alpha)p') > \alpha H(p) + (1 - \alpha) H(p') \quad (8.3)$$

成立。证毕。 \square

上凸函数在定义域内的极值必为极大值, 可以利用熵函数的这个性质证明熵函数的极值性。

直观来看, 随机变量的不确定程度并不都是一样的. 例如, 抛掷一枚均匀硬币结果所得到的信息量会比抛掷一枚偏畸硬币所得到的信息量大; 投掷一颗均匀骰子的试验比抛掷一枚均匀硬币的试验所得到的信息量大. 怎么度量这种不确定性呢? 香农指出, 存在这样的不确定性的度量, 它是随机变量的概率分布的函数, 而且必须满足 3 个公理性条件:

(1) 连续性条件: $f(p_1, p_2, \dots, p_n)$ 应是 $p_i, i = 1, 2, \dots, n$ 的连续函数;

(2) 等概时为单调函数: $f(1/n, 1/n, \dots, 1/n)$ 应是 n 的增函数;

(3) 递增性条件: 当随机变量的取值不是通过一次试验而是若干次试验才最后得到时, 随机变量在各次试验中的不确定性应该可加, 且其和始终与通过一次试验取得的不确定程度相同, 即

$$f(p_1, p_2, \dots, p_n)$$

$$= f((p_1 + p_2 + \dots + p_k), p_{k+1}, \dots, p_n) + (p_1 + p_2 + \dots + p_k) f(p'_1, p'_2, \dots, p'_k)$$

其中, $p'_k = p_k / (p_1 + p_2 + \dots + p_k)$ 。

香农根据这 3 个公理性条件于 1948 年先提出了熵的概念, 他当时并没有像我们现在这样把熵看成自信息的均值. 后来, Feinstein(范恩斯坦) 等人从数学上严格地证明了当满足上述条件时, 信息熵的表达形式是唯一的.

8.1.3 联合熵和条件熵

一个随机变量的不确定性可以用熵来表示, 这一概念可以方便地推广到多个随机变量。

定义 8.1.3. [联合熵] 二维随机变量 XY 的概率空间表示为

$$\begin{bmatrix} XY \\ P(XY) \end{bmatrix} = \begin{bmatrix} x_1y_1 & \cdots & x_iy_j & \cdots & x_ny_n \\ p(x_1y_1) & \cdots & p(x_iy_j) & \cdots & p(x_ny_n) \end{bmatrix}$$

其中, $p(x_iy_j)$ 满足概率空间的非负性和完备性: $0 \leq p(x_iy_j) \leq 1$, $\sum_{i=1}^n \sum_{j=1}^m p(x_iy_j) = 1$ 。

二维随机变量 XY 的联合熵定义为联合自信息的数学期望, 它是二维随机变量 XY 的不确定性的度量。

$$H(XY) = \sum_{i=1}^n \sum_{j=1}^m p(x_iy_j) I(x_iy_j) = -\sum_{i=1}^n \sum_{j=1}^m p(x_iy_j) \log p(x_iy_j)$$

定义 8.1.4. [条件熵] 考虑在给定 $X = x_i$ 的条件下, 随机变量 Y 的不确定性为

$$H(Y|x_i) = -\sum_j p(y_j|x_i) \log p(y_j|x_i)$$

对 $H(Y|x_i)$ 的所有可能值进行统计平均, 就得出给定 X 时, Y 的条件熵 $H(Y|X)$

$$\begin{aligned} H(Y|X) &= \sum_i p(x_i) H(Y|x_i) \\ &= -\sum_i \sum_j p(x_i) p(y_j|x_i) \log p(y_j|x_i) \\ &= -\sum_i \sum_j p(x_iy_j) \log p(y_j|x_i) \end{aligned}$$

性质 8.1.1. 联合熵和条件熵有如下关系:

$$H(XY) = H(X) + H(Y|X)$$

证明.

$$\begin{aligned} H(XY) &= \mathbb{E}(\log \frac{1}{p(xy)}) \\ &= \mathbb{E}(\log \frac{1}{p(x)p(y|x)}) \\ &= \mathbb{E}(\log \frac{1}{p(x)} + \log \frac{1}{p(y|x)}) \\ &= \mathbb{E}(\log \frac{1}{p(x)}) + \mathbb{E}(\log \frac{1}{p(y|x)}) \\ &= H(X) + H(Y|X) \end{aligned}$$

□

推论 8.1.1. 当二维随机变量 X , Y 相互独立时, 联合熵等于 X 和 Y 各自熵之和。

$$H(XY) = H(X) + H(Y)$$

证明. 因为随机变量 X, Y 相互独立, 所以有

$$\begin{aligned}
 p(x_i y_j) &= p(x_i)p(y_j) \\
 H(XY) &= E[-\log_2 p(xy)] \\
 &= E[-\log_2 p(x)p(y))] \\
 &= E[-\log_2 p(x) + \log_2 p(y))] \\
 &= E[-\log_2 p(x)) + E[-\log p(y)] \\
 &= H(X) + H(Y)
 \end{aligned}$$

证毕。 □

8.1.4 互信息和相对熵

互信息

定义 8.1.5. 一个事件 y_j 所给出关于另一个事件 x_i 的信息定义为互信息, 用 $I(x_i; y_j)$ 表示.

$$I(x_i; y_j) = I(x_i) - I(x_i|y_j) = \log_2 \frac{p(x_i|y_j)}{p(x_i)} \quad (8.4)$$

互信息 $I(x_i; y_j)$ 是已知事件 y_j 后所消除的关于事件 x_i 的不确定性, 它等于事件 x_i 本身的不确定性 $I(x_i)$ 减去已知事件 y_j 后对 x_i 仍然存在的不确定性 $I(x_i|y_j)$. 互信息的引出, 使信息的传递得到了定量的表示.

例 8.1.4. 某地二月份天气出现的概率分别为晴 $1/2$, 阴 $1/4$, 雨 $1/8$, 雪 $1/8$. 某一天有人告诉你: “今天不是晴天”, 把这句话作为收到的消息 y_1 , 求收到 y_1 后, y_1 与各种天气的互信息量.

解. 把各种天气记作 x_1 (晴), x_2 (阴), x_3 (雨), x_4 (雪). 收到消息 y_1 后, 各种天气发生的概率变成了后验概率:

$$p(x_1|y_1) = \frac{p(x_1 y_1)}{p(y_1)} = 0$$

$$p(x_2|y_1) = \frac{p(x_2 y_1)}{p(y_1)} = \frac{1/4}{1/4 + 1/8 + 1/8} = \frac{1}{2}$$

$$p(x_3|y_1) = \frac{p(x_3 y_1)}{p(y_1)} = \frac{1/8}{1/4 + 1/8 + 1/8} = \frac{1}{4}$$

同理

$$p(x_4|y_1) = \frac{1}{4}$$

根据互信息量的定义, 可计算出 y_1 与各种天气之间的互信息:

$$I(x_1; y_1) = \log_2 \frac{p(x_1|y_1)}{p(x_1)} = \infty$$

$$I(x_2; y_1) = \log_2 \frac{p(x_2|y_1)}{p(x_2)} = \log_2 \frac{1/2}{1/4} = 1\text{bit}$$

$$I(x_3; y_1) = \log_2 \frac{p(x_3|y_1)}{p(x_3)} = \log_2 \frac{1/4}{1/8} = 1\text{bit}$$

$$I(x_4; y_1) = \log_2 \frac{p(x_4|y_1)}{p(x_4)} = \log_2 \frac{1/4}{1/8} = 1\text{bit}$$

定义 8.1.6. 定义互信息 $I(x_i; y_j)$ 在 XY 的联合概率空间中的统计平均值为随机变量 X 和 Y 间的平均互信息。

$$I(X; Y) = \sum_x \sum_y p(x, y) I(x_i; y_j)$$

也称为互信息。

互信息有以下性质：

性质 8.1.2. [对称性]

$$I(X; Y) = I(Y; X)$$

证明.

$$\begin{aligned} I(X; Y) &= \sum_{i=1}^n \sum_{j=1}^m p(x_i y_j) \log_2 \frac{p(x_i | y_j)}{p(x_i)} \\ &= \sum_{i=1}^n \sum_{j=1}^m p(x_i y_j) \log_2 \frac{p(x_i y_i)}{p(x_i) p(y_j)} \\ &= \sum_{i=1}^n \sum_{j=1}^m p(x_i y_j) \log_2 \frac{p(y_j | x_i)}{p(y_j)} \\ &= I(Y; X) \end{aligned}$$

□

证毕。

对称性表示从 Y 中获得关于 X 的信息量等于从 X 中获得关于 Y 的信息量。

性质 8.1.3. [非负性]

$$I(X; Y) \geq 0$$

当且仅当 $p(x, y) = p(x)p(y)$ 即 X 与 Y 独立时，互信息为 0

证明.

$$\begin{aligned}
 -I(X;Y) &= \sum_{i=1}^n \sum_{j=1}^m p(x_i y_j) \log_2 \frac{p(x_i) p(y_j)}{p(x_i y_j)} \\
 &\leq \log_2 \sum_{i=1}^n \sum_{j=1}^m p(x_i y_j) \frac{p(x_i) p(y_j)}{p(x_i y_j)} \\
 &= \log_2 \sum_{i=1}^n \sum_{j=1}^m p(x_i) p(y_j) \\
 &= 0
 \end{aligned}$$

所以

$$I(X;Y) \geq 0$$

证毕。 □

平均互信息是非负的, 说明给定随机变量 Y 后, 一般来说总能消除一部分关于 X 的不确定性.

相对熵

相对熵是两个随机分布之间距离的度量。统计学上对应于对数似然比的期望。

定义 8.1.7. 定义同一个随机变量 x 的两个概率密度函数 $p(x)$ 和 $q(x)$ 间的相对熵为:

$$D(p||q) = \mathbb{E}_{x \sim p} \left[\log \frac{p(x)}{q(x)} \right] = \mathbb{E}_{x \sim p} [\log p(x) - \log q(x)]$$

在机器学习中, 相对熵更常用的名称是 KL 散度, 记做 $D_{KL}(p||q)$ 。

KL 散度有很多有用的性质,

- 可以证明它是非负的;
- KL 散度为 0 当且仅当 p 和 q 在离散型变量的情况下是相同的分布, 或者在连续型变量的情况下是“几乎处处”相同的;
- KL 散度是非负的并且可以度量两个分布之间的差异。然而, 它并不是距离, 因为它不是对称的;
- 联合分布 $p(X, Y)$ 和 $p(X)p(Y)$ 之间的 KL 散度可以作为 X 和 Y 的互信息的另一种定义:

$$I(X;Y) := D_{KL}(p(X, Y) \| p(X)p(Y)) = \sum_x \sum_y p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

性质 8.1.4. [互信息和熵的关系]

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(XY)$$

当 X, Y 统计独立时, $I(X; Y) = 0$.

性质 8.1.5. $H(X) = I(X; X)$

也就是随机变量 X 的熵是自己对自己的互信息。

性质 8.1.6. [极值性]

$$I(X; Y) \leq H(X), I(X; Y) \leq H(Y)$$

由于 $I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$, 而条件熵 $H(X|Y)$ 、 $H(Y|X)$ 是非负的, 所以可得到 $I(X; Y) \leq H(X), I(X; Y) \leq H(Y)$. 极值性说明从一个事件获得的关于另一个事件的信息量至多只能是另一个事件的平均自信息量, 不会超过另一事件本身所含的信息量. 最好的情况是通信后 $I(X; Y) = H(X) = H(Y)$, 最坏的情况是当 X, Y 相互独立时, 从一个事件不能得到另一个事件的任何信息, 即 $I(X; Y) = 0$.

推论 8.1.2. 条件熵和信息熵的关系

$$H(X|Y) \leq H(X), \quad H(Y|X) \leq H(Y) \quad (8.5)$$

证明. 利用式(8.2)先证明式 $H(X|Y) \leq H(X)$

$$\begin{aligned} & - \sum_i \sum_j p(x_i y_j) \log_2 p(x_i | y_j) \\ &= - \sum_i \sum_j p(y_j) p(x_i | y_j) \log_2 p(x_i | y_j) \\ &= - \sum_j p(y_j) \sum_i p(x_i | y_j) \log_p p(x_i | y_j) \\ &\leq - \sum_j p(y_j) \sum_i p(x_i | y_j) \log_2 p(x_i) \\ &= - \sum_i \sum_j p(x_i y_j) p \log_2 p(x_i) \\ &= - \sum_i p(x_i) \log_2 p(x_i) = H(X) \end{aligned}$$

当 $p(x_i | y_j) = p(x_i)$ 时等号成立。

类似地, 可以证明 $H(Y|X) \leq H(Y)$ 。证毕。 □

推论 8.1.3. 联合熵和信息熵的关系:

$$H(XY) \leq H(X) + H(Y) \quad (8.6)$$

证明.

$$H(XY) = H(X) + H(Y|X) \leq H(X) + H(Y)$$

当 X, Y 相互独立时等号成立。推广到 N 个随机变量的情况:

$$H(X_1 X_2 \cdots X_N) \leq H(X_1) + H(X_2) + \cdots + H(X_N)$$

当 X_1, X_2, \dots, X_N 相互独立时, 等号成立。 □

8.1.5 熵、相对熵和互信息的链式法则

即两个随机变量 X 和 Y 的联合熵等于 X 的熵加上在 X 已知条件下 Y 的条件熵, 这个关系可以方便地推广到 N 个随机变量的情况, 即

$$H(X_1 X_2 \cdots X_N) = H(X_1) + H(X_2 | X_1) + \cdots + H(X_N | X_1 X_2 \cdots X_{N-1})$$

称为熵函数的链规则。

如果 N 个随机变量 X_1, X_2, \dots, X_N 相互独立, 则有

$$H(X_1 X_2 \cdots X_N) = \sum_{i=1}^N H(X_i) \quad (8.7)$$

互信息的链式法则

我们先定义条件互信息:

定义 8.1.8. 随机变量 X 和 Y 在给定随机变量 Z 时的条件互信息为

$$I(X; Y|Z) = H(X|Z) - H(X|Y, Z) = \sum_{i=1}^l \sum_{j=1}^m \sum_{k=1}^n p(x_i, y_j, z_k) \log_2 \frac{p(x_i, y_j|z_k)}{p(x_i|z_k)p(y_j|z_k)}$$

性质 8.1.7. 互信息的链式法则

$$I(X_1, X_2, \dots, X_n; Y) = \sum_{i=1}^n I(X_i; Y | X_{i-1}, X_{i-2}, \dots, X_1)$$

相对熵的链式法则

我们先定义条件相对熵:

定义 8.1.9. 联合概率密度函数 $p(x, y)$ 和 $q(x, y)$ 的条件概率熵 $D(p(y|x)||q(y|x))$ 定义为条件概率密度函数 $p(y|x)$ 和 $q(y|x)$ 间关于 $p(x)$ 的平均相对熵, 即

$$D(p(y|x)||q(y|x)) = \sum_{i=1}^m p(x_i) \sum_{j=1}^n p(y_j|x_i) \log_2 \frac{p(y_j|x_i)}{q(y_j|x_i)} = \sum_{i=1}^m \sum_{j=1}^n p(x_i, y_j) \log_2 \frac{p(y_j|x_i)}{q(y_j|x_i)}$$

性质 8.1.8. 互信息的链式法则

$$D(p(x, y)||q(x, y)) = D(p(x)||q(x)) + D(p(y|x)||q(y|x))$$

8.1.6 信息不等式

性质 8.1.9. 信息不等式 设 $p(x), q(x)$ 是两个概率密度函数, 则

$$D(p||q) \geq 0$$

当且仅当对任意 x , $p(x) = q(x)$ 时, 等号成立。

证明.

$$\begin{aligned} -D(p||q) &= -\sum_x p(x) \log \frac{p(x)}{q(x)} = \sum_x p(x) \log \frac{q(x)}{p(x)} \\ &\leq \log \sum_x p(x) \frac{q(x)}{p(x)} = \log \sum_x q(x) \\ &= \log 1 = 0 \end{aligned}$$

□

利用信息不等式，可以导出如下推论

推论 8.1.4. 对任意两个随机变量 X 和 Y ,

$$I(X; Y) \geq 0$$

当且仅当 X 与 Y 相互独立时，等号成立.

推论 8.1.5. 条件相对熵非负，即 $D(p(y|x)||q(y|x)) \geq 0$. 当且仅当对任意 y 满足 $p(y|x) = q(y|x)$ 时，等号成立。

推论 8.1.6. 条件互信息非负，即 $I(X; Y|Z) \geq 0$ ，当且仅当对给定随机变量 Z 时， X 和 Y 是条件独立的，等号成立。

信息处理定理

下面我们介绍信息处理定理。为了表述数据处理定理，需要引入三元随机变量 X, Y, Z 的平均条件互信息和平均联合互信息的概念。

定义 8.1.10. [平均条件互信息]

$$I(X; Y|Z) = E[I(x; y|z)] = \sum_x \sum_y \sum_z p(xyz) \log_2 \frac{p(x|yz)}{p(x|z)} \quad (8.8)$$

它表示随机变量 Z 给定后，从随机变量 Y 所得到的关于随机变量 X 的信息量。

定义 8.1.11. [平均联合互信息]

$$I(X; YZ) = E[I(x; yz)] = \sum_x \sum_y \sum_z p(xyz) \log_2 \frac{p(x|yz)}{p(x)} \quad (8.9)$$

它表示从二维随机变量 YZ 所得到的关于随机变量 X 的信息量。

可以证明

$$\begin{aligned} I(X; YZ) &= \sum_x \sum_y \sum_z p(xyz) \log_2 \frac{p(x|z)p(x|yz)}{p(x)p(x|z)} \\ &= I(X; Z) + I(X; Y|Z) \end{aligned} \quad (8.10)$$

同理

$$I(X; YZ) = I(X; Y) + I(X; Z|Y) \quad (8.11)$$

定理 8.1.1. [数据处理定理] 如果随机变量 X, Y, Z 构成一个马尔可夫链, 则有以下关系成立:

$$I(X; Z) \leq I(X; Y), I(X; Z) \leq I(Y; Z) \quad (8.12)$$

等号成立的条件是对于任意的 x, y, z , 有 $p(x|yz) = p(x|z)$ 和 $p(z|xy) = p(z|x)$ 。

证明. 当 X, Y, Z 构成一个马尔可夫链时, Y 值给定后, X, Z 可以认为是互相独立的. 所以,

$$I(X; Z|Y) = 0$$

又因为 $I(X; YZ) = I(X; Y) + I(X; Z|Y) = I(X; Z) + I(X; Y|Z)$, 并且 $I(X; Y|Z) \geq 0$, 所以 $I(X; Z) \leq I(X; Y)$ 。

当 $p(x|yz) = p(x|z)$ 时, Z 值给定后, X 和 Y 相互独立, 所以

$$I(X; Y|Z) = 0$$

因此

$$I(X; Z) = I(X; Y)$$

这时 $p(x|yz) = p(x|z) = p(x|y)$. Y, Z 为确定关系时显然满足该条件。

同理可以证明 $I(X; Z) \leq I(Y; Z)$, 并且当 $p(z|xy) = p(z|x)$ 时, 等号成立。

证毕。 □

$I(X; Z) \leq I(X; Y)$ 表明从 Z 所得到的关于 X 的信息量小于等于从 Y 所得到的关于 X 的信息量. 如果把 $Y \rightarrow Z$ 看作数据处理系统, 那么通过数据处理后, 虽然可以满足我们的某种具体要求, 但是从信息量来看, 处理后会损失一部分信息, 最多保持原有的信息, 也就是说, 对接收到的数据 Y 进行处理后, 决不会减少关于 X 的不确定性. 这个定理称为数据处理定理. 数据处理定理与日常生活中的经验是一致的. 比如: 通过别人转述一段话或多或少会有一些失真, 通过书本得到的间接经验总不如直接经验来得详实.

8.2 连续分布的微分熵和最大熵

8.2.1 连续信源的微分熵

连续随机变量的取值是连续的, 一般用概率密度函数来描述其统计特征.

单变量连续信源的数学模型为 $X : \begin{bmatrix} \mathbb{R} \\ p(x) \end{bmatrix}$, 并且满足 $\int_R p(x)dx = 1$, \mathbb{R} 是实数域, 表示 X 的取值范围。

对于取值范围有限的连续信源还可以表示成 $X : \begin{bmatrix} (a, b) \\ p(x) \end{bmatrix}$, 并且满足 $\int_a^b p(x)dx = 1$, (a, b) 是 X 的取值范围。

通过对连续变量的取值进行量化分层, 可以将连续随机变量用离散随机变量来逼近. 量化间隔越小, 离散随机变量与连续随机变量越接近. 当量化间隔趋于 0 时, 离散随机变量就变成了连续随机变量. 通过对离散随机变量的熵取极限, 可以推导出连续随机变量熵的计算公式.

我们把连续随机变量 X 的取值分割成 n 个小区间, 各小区间等宽, 区间宽度 $\Delta = \frac{b-a}{n}$, 则变量落在第 i 个小区间的概率为

$$P_r\{a + (i-1)\Delta \leq x \leq a + i\Delta\} = \int_{a+(i-1)\Delta}^{a+i\Delta} p(x)dx = p(x_i)\Delta \quad (8.13)$$

其中, x_i 是 $a + (i-1)\Delta$ 到 $a + i\Delta$ 之间的某一值. 当 $p(x)$ 是连续函数时, 由中值定理可知, 比存在一个 x_i 使式(8.13)成立, 这样, 连续变量 X 就可用取值为 $x_i, i = 1, 2, \dots, n$ 的离散变量来近似, 连续信源就被量化成离散信源, 这 n 个取值对应的概率分布为 $p_i = p(x_i)\Delta$, 这时的离散信源熵是

$$H(X) = - \sum_{i=1}^n p(x_i)\Delta \log_2 [p(x_i)\Delta] = - \sum_{i=1}^n p(x_i)\Delta \log_2 p(x_i) - \sum_{i=1}^n p(x_i)\Delta \log_2 \Delta \quad (8.14)$$

当 $n \rightarrow \infty$ 时, $\Delta \rightarrow 0$, 如果(8.14)极限存在, 离散信源熵就变成了连续信源的熵:

$$\begin{aligned} \lim_{\Delta \rightarrow 0} H(X) &= \lim_{n \rightarrow \infty} - \sum_{i=1}^n p(x_i)\Delta \log_2 p(x_i) - \lim_{n \rightarrow \infty} \sum_{i=1}^n p(x_i)\Delta \log_2 \Delta \\ &= - \int_a^b p(x) \log_2 p(x) dx - \lim_{n \rightarrow \infty} \log_2 \int_a^b p(x) dx \\ &= - \int_a^b p(x) \log_2 p(x) dx - \lim_{\Delta \rightarrow 0} \log_2 \Delta \end{aligned} \quad (8.15)$$

式(8.15)第一项一般是定值, 第二项为无穷大量, 因此连续信源的熵实际是无穷大量. 这一点是可以理解的, 因为连续信源的可能取值是无限多的, 所以它的不确定性是无限大的, 当确知输出为某值后, 所获得的信息量也是无限大. 在丢掉第二项后, 定义第一项为连续信源的微分熵:

$$h(X) = - \int_R p(x) \log_2 p(x) dx \quad (8.16)$$

微分熵又称为差熵. 虽然 $h(X)$ 已不能代表连续信源的平均不确定性, 也不能代表连续信源输出的信息量, 但是它具有和离散熵相同的形式, 也具有离散熵的主要特性, 比如可加性, 但是不具有非负性. 另外, 我们在实际问题中常常考虑的是熵差, 比如平均互信息, 在讨论熵差时, 只要两者离散逼近时所取的间隔 Δ 一致, 这两个无限大量就将互相抵消, 所以熵差具有信息的特性, 如非负性. 由此可见, 连续信源的熵 $h(X)$ 具有相对性.

同样, 可以定义两个连续随机变量的联合熵:

$$h(XY) = - \iint_{\mathbb{R}^2} p(xy) \log_2 p(xy) dxdy \quad (8.17)$$

以及条件熵

$$h(X|Y) = - \iint_{\mathbb{R}^2} p(xy) \log_2 p(y|x) dxdy \quad (8.18)$$

$$h(Y|X) = - \iint_{\mathbb{R}^2} p(xy) \log_2 p(x|y) dxdy \quad (8.19)$$

并且它们之间也有与离散随机变量一样的相互关系:

$$h(XY) = h(X) + h(Y|X) = h(Y) + h(X|Y) \quad (8.20)$$

$$h(X|Y) \leq h(X) \quad (8.21)$$

$$h(Y|X) \leq h(Y) \quad (8.22)$$

例 8.2.1. X 是在区间 (a, b) 内服从均匀分布的连续随机变量, 求微分熵.

$$p(x) = \begin{cases} \frac{1}{b-a} & x \in (a, b) \\ 0 & x \notin (a, b) \end{cases}$$

解.

$$h(X) = - \int_a^b p(x) \log_2 p(x) dx = - \int_a^b \frac{1}{b-a} \log_2 \frac{1}{b-a} dx = \log_2(b-a)$$

当 $(b-a) > 1$ 时, $h(X) > 0$

当 $(b-a) = 1$ 时, $h(X) = 0$

当 $(b-a) < 1$ 时, $h(X) < 0$ 这说明连续熵不具有非负性, 失去了信息的部分含义和性质 (但是熵差具有信息的特性)。

例 8.2.2. 求均值为 m , 方差为 σ^2 的高斯分布的随机变量的微分熵.

解. 高斯随机变量的概率密度为

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

微分熵为

$$\begin{aligned} h(X) &= - \int_{-\infty}^{+\infty} p(x) \log_2 p(x) dx \\ &= - \int_{-\infty}^{+\infty} p(x) \log_2 \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}} \right] dx \\ &= - \int_{-\infty}^{+\infty} p(x) \log_2 e \frac{1}{\sqrt{2\pi}\sigma} - \log_2 \int_{-\infty}^{+\infty} p(x) \left[-\frac{(x-m)^2}{2\sigma^2} \right] dx \\ &= \log_2 \sqrt{2\pi}\sigma + \log_2 e \int_{-\infty}^{+\infty} p(x) \frac{(x-m)^2}{2\sigma^2} dx \\ &= \log_2 \sqrt{2\pi}\sigma + \frac{1}{2} \log_2 e \\ &= \log_2 \sqrt{2\pi e \sigma} \end{aligned}$$

这里对数以 2 为底, 所得微分熵的单位为比特/样值, 如果对数取以 e 为底, 则得到

$$h(X) = \log_2 \sqrt{2\pi e \sigma} \text{奈特/样值}$$

我们看到, 正态分布的连续信源的微分熵与数学期望 m 无关, 只与方差 σ^2 有关.

8.2.2 连续信源的最大熵

离散信源当信源符号为等概分布时有最大熵。连续信源微分熵也有极大值，但是与约束条件有关，当约束条件不同时，信源的最大熵不同。我们一般关心的是下面两种约束下的最大熵。

定理 8.2.1. 在均值一定的情况下，服从均匀分布的随机变量 X 具有最大熵。

即

$$p(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{其他} \end{cases}$$

$$h(X) = - \int_a^b p(x) \log_2 p(x) dx = - \int_a^b \frac{1}{b-a} \log_2 \frac{1}{b-a} dx = \log_2(b-a)$$

因此对于输出均值受限的连续信源，当满足均匀分布时达到最大熵。这个结论与离散信源在等概分布时达到最大熵的结论类似。

定理 8.2.2. 对于固定均值为 μ 和方差为 σ^2 的连续随机变量，当服从高斯分布 $N(\mu, \sigma^2)$ 时具有最大熵。

证明。对给定的 $p(x)$ ，利用相对熵非负，有

$$H(p) \leq - \int p(x) \log q(x) dx$$

取 $q(x) = N(\mu, \sigma^2)$ ，有

$$\begin{aligned} H(p) &\leq - \int p(x) \log \left(\frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\} \right) dx = \int p(x) \left\{ \frac{(x-\mu)^2}{2\sigma^2} + \log \sqrt{2\pi}\sigma \right\} dx \\ &= \frac{1}{2\sigma^2} \int p(x)(x-\mu)^2 dx + \log \sqrt{2\pi}\sigma = \frac{1}{2} + \log \sqrt{2\pi}\sigma \end{aligned}$$

当 $p(x) = N(\mu, \sigma^2)$ 时，可以取等，证毕。 \square

这说明，当均值和方差一定时，高斯分布的连续信源的熵最大。

8.3 信息论在数据科学中的应用

8.3.1 基于信息量的度量

信息熵

信息熵：

$$H(X) = \mathbb{E}[I(x_i)] = - \sum_{i=1}^q p(x_i) \log p(x_i)$$

信息熵是对信息不确定性的度量，也可以这样理解，数据信息熵越小，数据就越纯。

互信息

假设带标签 X 的数据集有若干属性 Y_1, Y_2, \dots, Y_n , 我们想通过选择数据集的某个属性判断数据集的标签, 那么我们就要根据哪一个属性对标签的信息量最大以选择属性。这时我们就要利用互信息

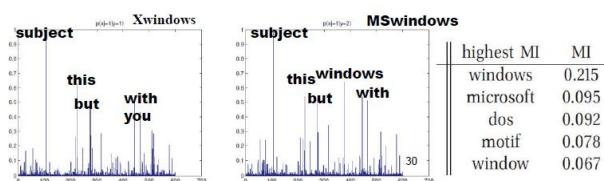
$$\arg \max_i I(X; Y_i) = \arg \max_i (H(X) - H(X|Y_i))$$

属性对标签的互信息可以理解成选择特定属性后, 不确定性的下降量, 也就是数据纯度的提升量。在机器学习中, 这个量等价于信息增益。

关于信息熵和互信息(信息增益)在机器学习的应用, 可以参考介绍决策树算法的相关书籍。

例 8.3.1. 在特征选择时, 可以通过计算特征与目标之间的互信息, 选择与目标互信息最大的那些特征, 抛弃与目标关系不大的特征。

- 给定文档分类任务, 将文档分成 *class 1 (X windows)* 和 *class 2 (MS windows)*, 特征为 600 个二维特征(600 个词语分别是否在文档中出现), 令 $p(x_i)$ 为词语在文档中出现的概率, $p(x_i|y_j)$ 为在 y_j 分类下词语在文档中出现的概率。则可计算 $I(X; Y) = H(X) - H(X|Y)$, 互信息高的词语(*windows, microsoft*)更有判别性。



KL 散度

相对熵或称 KL 散度可以衡量同一个随机变量 x 的概率分布 $p(x)$ 和 $q(x)$ 的差异:

$$D_{KL}(p||q) = \mathbb{E}_{x \sim p} [\log p(x) - \log q(x)]$$

并且 KL 散度具有非负性, 当且仅当 p 和 q 在离散型变量的情况下是相同的分布, 或者在连续型变量的情况下是“几乎处处”相同, KL 散度为 0。但 KL 散度并不是距离, 因为它不满足对称性。

交叉熵

一个和 KL 散度密切联系的量是交叉熵(cross-entropy):

定义 8.3.1. 设关于随机变量 x 的两个分布 $p(x), q(x)$, 关于这两个分布的交叉熵定义为:

$$H(p, q) = -\mathbb{E}_{x \sim p} \log q(x) = H(p) + D_{KL}(p||q) \quad (8.23)$$

- 针对 q 最小化交叉熵等价于最小化 KL 散度，因为 q 并不参与被省略的那一项。
- 在给定 p 的情况下，如果 q 和 p 越接近，交叉熵越小；如果 q 和 p 越远，交叉熵就越大。

JS 散度

JS 散度 (Jensen-Shannon Divergence) 是一种对称的衡量两个分布相似度的度量方式。

定义 8.3.2. 设关于随机变量 x 的两个分布 $p(x), q(x)$ ，关于这两个分布的 **JS 散度** 定义为：

$$D_{JS}(p, q) = \frac{1}{2}D_{KL}(p, M) + \frac{1}{2}D_{KL}(q, M)$$

其中 $M = \frac{1}{2}(p + q)$

JS 散度是 KL 散度一种改进。但两种散度都存在一个问题，即如果两个分布 p, q 没有重叠或者重叠非常少时，KL 散度和 JS 散度都很难衡量两个分布的距离。

8.3.2 其他概率相关的度量

本节还将介绍一些其他和概率相关的度量，作为基于信息论的度量的补充。

马氏距离

在前面，我们介绍了一些关于度量两个向量相似度的一些方法。

并且我们提到了闵氏距离 (包括曼哈顿距离、欧氏距离和切比雪夫距离) 存在明显的缺点，并通过下例进行了说明。

例 8.3.2. 给定二维样本 (身高, 体重)，其中身高范围是 $150 \sim 190$ ，体重范围是 $50 \sim 60$ ，有三个样本： $a(180, 50)$, $b(190, 50)$, $c(180, 60)$ 。

- 通过计算可以得出 ab 之间的闵氏距离等于 ac 之间的闵氏距离，但是身高的 $10cm$ 不等价于体重的 $10kg$ 。

现在我们就来介绍解决这个问题的一种度量相似度的方式。

定义 8.3.3. 马氏距离：表示点与一个分布之间的距离。有 m 个样本向量 $\mathbf{x}_1, \dots, \mathbf{x}_m$ ，协方差矩阵记为 \mathbf{S} ，均值记为向量 μ ，则其中样本向量 \mathbf{x} 到 μ 的马氏距离表示为：

$$dist(\mathbf{x}) = \sqrt{(\mathbf{x} - \mu)^T \mathbf{S}^{-1} (\mathbf{x} - \mu)}$$

而其中向量 \mathbf{x}_i 与 \mathbf{x}_j 之间的马氏距离定义为：

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{S}^{-1} (\mathbf{x}_i - \mathbf{x}_j)}$$

若协方差矩阵是单位矩阵（各个样本向量之间独立同分布），则公式就成了：

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}$$

也就是欧氏距离了。

马氏距离的优点：量纲无关，排除变量之间的相关性的干扰。

皮尔逊相关系数

相关系数是衡量随机变量 x 与 y 线性相关程度的一种方法，一般用 r 表示。 r 的取值范围是 $[-1, 1]$ 。 r 的绝对值越大，则表明 x 与 y 线性相关度越高。当 x 与 y 线性相关时，相关系数取值为 1（正线性相关）或 -1（负线性相关）。

定义 8.3.4. 设随机变量 x, y ，皮尔逊相关系数定义为：

$$r(x, y) = \frac{\text{Cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} = \frac{E((x - Ex)(y - Ey))}{\sqrt{D(x)}\sqrt{D(y)}}$$

其中， $\text{Cov}(x, y)$ 为 x 与 y 的协方差， $\sqrt{D(x)}$ 为 x 的方差， $\sqrt{D(y)}$ 为 y 的方差。

Wasserstein 距离

Wasserstein 距离（Wasserstein Distance）也用于衡量两个分布之间的距离。

定义 8.3.5. 对于两个分布 q_1, q_2 ， p th-Wasserstein 距离定义为

$$W_p(q_1, q_2) = \left(\inf_{\gamma(x, y) \in \Gamma(q_1, q_2)} E_{(x, y) \sim \gamma(x, y)} [d(x, y)^p] \right)^{\frac{1}{p}}$$

其中 $\Gamma(q_1, q_2)$ 是边际分布为 q_1 和 q_2 的所有可能的联合分布集合， $d(x, y)$ 为 x 和 y 的距离，比如 l_p 距离等。

Wasserstein 距离相比 KL 散度和 JS 散度的优势在于：即使两个分布没有重叠或者重叠非常少，Wasserstein 距离仍然能反映两个分布的远近。

例 8.3.3. 对于 \mathbb{R}^n 空间中的两个高斯分布 $p = N(\mu_1, \Sigma_1)$ 和 $q = N(\mu_2, \Sigma_2)$ ，它们的 2nd-Wasserstein 距离为

$$W_2(p, q) = \|\mu_1 - \mu_2\|_2^2 + \text{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_2^{\frac{1}{2}}\Sigma_1\Sigma_2^{\frac{1}{2}})^{\frac{1}{2}})$$

当两个分布的方差为 0 时，2nd-Wasserstein 距离等价于欧氏距离。

Jaccard 系数

Jaccard 系数又称为 Jaccard 相似系数，用于比较有限样本集之间的相似性与差异性。Jaccard 系数值越大，样本相似度越高。

定义 8.3.6. 两个集合 A 和 B 的交集元素在 A , B 的并集中所占的比例，称为两个集合的杰卡德相似系数，用符号 $J(A, B)$ 表示。

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

当集合 A , B 都为空时， $J(A, B)$ 定义为 1。

杰卡德相似系数是衡量两个集合的相似度一种指标。

对于等概率的随机排列，两个集合的 minHash 值相同的概率等于两个集合的 Jaccard 相似度。关于 minHash 算法，可以参考有关数据科学算法的教材如 *Mining of Massive Datasets*。

Jaccard 距离

Jaccard 距离：与 Jaccard 系数相反的概念是杰卡德距离。

定义 8.3.7. 杰卡德距离可用如下公式表示：

$$J_D(\mathbb{A}, \mathbb{B}) = 1 - J(\mathbb{A}, \mathbb{B}) = \frac{|\mathbb{A} \cup \mathbb{B}| - |\mathbb{A} \cap \mathbb{B}|}{|\mathbb{A} \cup \mathbb{B}|}$$

杰卡德距离用两个集合中不同元素占所有元素的比例来衡量两个集合的区分度。

例 8.3.4. 杰卡德相似系数用在衡量样本的相似度上。

样本 A 与样本 B 是两个 n 维向量，而且所有维度的取值都是 0 或 1。例如： $A = (0111)$ 和 $B = (1011)$ 。我们将样本看成是一个集合，1 表示集合包含该元素，0 表示集合不包含该元素。

p : 样本 A 与 B 都是 1 的维度的个数；

q : 样本 A 是 1, 样本 B 是 0 的维度的个数；

r : 样本 A 是 0, 样本 B 是 1 的维度的个数；

s : 样本 A 与 B 都是 0 的维度的个数。

那么样本 A 与 B 的杰卡德相似系数可以表示为：

$$J = \frac{p}{p + q + r}$$

这里 $p + q + r$ 可理解为 A 与 B 的并集的元素个数，而 p 是 A 与 B 的交集的元素个数。

而样本 A 与 B 的杰卡德距离表示为：

$$J_D = \frac{q + r}{p + q + r}$$

8.4 阅读材料

熵的概念首先在热力学中引入，用于表述热力学第二定律。此后，统计力学告诉我们，在系统的某个宏观状态中，热力学熵与微观状态数目的对数之间存在着联系。此项研究工作归功于玻尔兹曼的伟大成就，他给出了方程式 $S = k \ln W$ ，该方程式作为墓志铭刻在了他的墓碑上。

20 世纪 30 年代，Hanley 在通信系统中引入了信息的对数度量。这个度量本质上是字母表大小的对数。本章中熵与互信息的定义由香农首先给出。相对熵概念由库尔贝克 (Kullback) 和 Leibler 首先定义，它有各种各样的命名，包括 Kullback-Leibler 距离、叉熵、信息散度、信息判别，在 Csiszdr 和 Amari 中其详细的论述。

这些最的许多简单性质都是由香农发展起来的。费诺不等式的证明见 Fano。充分统计量概念由费希尔 (FiSher) 定义，而最小充分统计量是由 Lehmann 和 Scheffc 引入的。互信息与充分性

关系的解释归功于 Kullback。Brillouin 和 Jaynes 对信息论和热力学之间的关系给予了广泛的讨论。

信息物理学是一门相当新型的学科，产生于统计力学、量子力学和信息论。讨论的关键问题是如何将信息表示物理化。量子信道容量（物理系统中可分辨的制备数量的对数）和量子数据压缩都是定义明确的问题，利用冯·诺伊曼熵获得了完美的解答。由于量子纠缠的存在，以及观察到的物理事件的边际分布与任何联合分布均不一致（没有局部的真实）这一结论（体现于贝尔（Bell）不等式），量子信息的研究有了新的课题。Nielsen 和 Chuang 所著的基础文献较为详尽地论述了量子信息论，同时包含本书中的许多结论的量子形式。人们也试图确定在计算上是否存在本质的物理限制，这些工作包括 Bennett 以及 Bennett 与 Landauer。

习题

习题 8.1. 同时抛 2 颗骰子，事件 A, B, C 分别表示：(A) 仅有一个骰子是 3；(B) 至少有一个骰子是 4；(C) 骰子上点数的总和为偶数。是计算事件 A, B, C 发生后所提供的信息量。

习题 8.2. 用递推计算熵函数 $H(1/3, 1/3, 1/6, 1/6)$ 的值。

习题 8.3. X 和 Y 是 $\{0, 1, 2, 3\}$ 上的独立、均匀分布的随机变量，求：

- (1) $H(X + Y), H(X - Y), H(X \cdot Y)$
- (2) $H(X + Y, X - Y), H(X + Y, X \cdot Y)$

习题 8.4. X, Y, Z 为 3 个随机变量，证明一下不等式成立并指出等号成立的条件：

- (1) $H(XY|Z) \geq H(X|Z)$
- (2) $I(XY; Z) \geq I(X; Z)$
- (3) $H(XYZ) - H(XY) \leq H(XZ) - H(X)$
- (4) $I(X; Z|Y) \geq I(Z; Y|X) - I(Z; Y) + I(X; Z)$

习题 8.5. 找出一个概率分布 $\{p_1, p_2, p_3, p_4, p_5\}$ ，并且 $p_i > 0$ ，使得 $H(p_1, p_2, p_3, p_4, p_5) = 2$

习题 8.6. 假定 $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow \cdots \rightarrow X_n$ 形成一个马尔科夫链，那么 $p(x_1 x_2 \cdots x_n) = p(x_1)p(x_2|x_1) \cdots p(x_n|x_{n-1})$ ，化简 $I(X_1; X_2 \cdots X_n)$

习题 8.7. 假定 X 是一个离散随机变量， $g(X)$ 是 X 的函数，证明： $H[g(X)] \leq H(X)$ 。

习题 8.8. 三扇门中有一扇门后面藏有一袋金子，并且三扇门后面藏有金子的可能性相同。如果有人随机打开一扇门并告诉你门后是否藏有金子，他给了你多少关于金子位置的信息量？

参考文献

- [1] S.Amari.Differential-Geometrical Methods in Statistics.Springer-Vcrlag,New York, 1985.
- [2] C.H.Bennett.Demons, engines and the second law.Sci.Am.259(5):108-116.Nov.1987.
- [3] C.H.Bennett and R.Landauer. The fundamental physical limits of computation. Sci. Am. 255(1):48-56,July 1985.
- [4] I.Csiszar. Information type measures of difference of probability distributions and indirect observations. Stud. Sci. Math. Hung. 2:299-318,1967.
- [5] R Jozsa and B. Schumacher. A new proof of the quantum noiseless coding theorem. J Mod. Opt, pages 2343-2350,1994
- [6] S.Kullback and R.A.Leibler. On information and sufficiency. Ann.Math.Stat.22:79-87,1951.
- [7] D.Lindley.Boltzmann's Atom:The Great Debate That Launched A Revolution in Physics. Free Press,New York,2001.
- [8] M.Nielsen and I.Chang. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge,2000.
- [9] C.E.Shannon.A mathematical theory of communication.Bell Syst.Tech.J.27:379-423,623-656,1948.

草稿请勿外传

第九章 概率模型

一大类机器模型是基于概率的。一些参数或数据的概率基于另外一些参数或数据的概率，利用图的概念表示概率分布称为结构化概率模型或图模型。我们首先学习图模型的一般结构，包括有向图模型和无向图模型。之后我们学习模型中概率密度的估计方法，包括参数的和非参数的。在参数估计中，我们主要学习极大似然估计方法。之后，我们利用统计决策理论，对模型进行评估与选择。

禁
止
下
载
请
勿
外
传

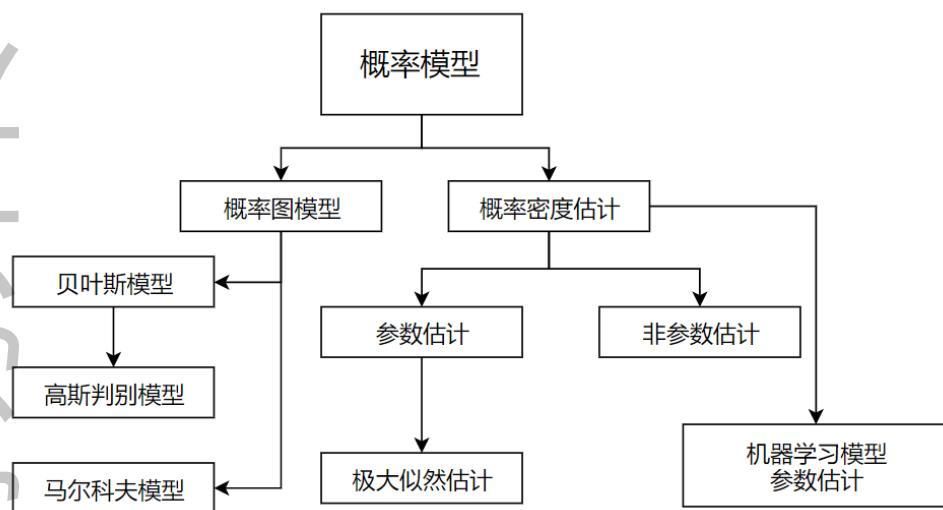


图 9.1: 本章导图

9.1 建模的概率思想

在本讲中我们将讨论一些概率方法。在这些模型里我们会对概率函数进行估计，并使用这些概率函数来指导我们的决策（例如分类、回归）。这是一个很大的话题，并且在该领域中存在

着大量的方法。本讲只会讨论一些最基本的概念和方法，并简要介绍其他方法。本讲的主要目的是介绍一些重要的概念和方法以及进行推断和决策的概率路线。

9.1.1 分布和推断

令 $p(X, Y)$ 表示 X 与 Y 的联合分布。假设 Y 是可以基于 X 按照某种方式进行预测， X 和 Y 之间必定存在着某种联系。换言之， X 与 Y 不可能是独立的——即

$$p_{X,Y}(X, Y) \neq p_X(X)p_Y(Y)$$

相反，如果我们有

$$p_{X,Y}(X, Y) = p_X(X)p_Y(Y)$$

则已知 X 对于预测 Y 没有提供有效的信息，我们也无法学习到任何有意义的模型。

边缘分布 $p_X(x)$ 度量的是在不考虑 Y 的影响（或者说，将 Y 的影响从联合分布中通过积分移除出去）时，数据 X 的密度函数。它被称为边缘似然。

在不考虑 X 时，边缘分布 $p_Y(y)$ 是关于 Y 的先验分布 (prior distribution)。它反映的是在观测到任何输入之前了解到的那些关于 Y 的先验知识。

在我们观测到 X 之后，由于 X 和 Y 之间的联系，我们可以更准确地估计 Y 的值。即 $p_{Y|X}(Y | X)$ （简写为 $p(Y | X)$ ）是关于 Y 的、比 $p_Y(Y)$ 更好的估计。这种分布被称为后验分布。当给定更多的证据（ X 的样本）时，可根据其更新我们对 Y 的信念。更新后的信念，即后验或条件分布 $p(Y | X)$ 将作为我们在给定 X 时对 Y 的最佳估计。

使用证据来更新信念（即更新后验分布）的过程被称为概率推断。我们还需要决定在获得后验之后能做些什么，因此决策过程紧随其后。分类是一种经典的决策类型。

9.1.2 生成式模型 vs. 判别式模型

如果直接对条件/后验分布 $p(Y | X)$ 进行建模，那么这是一个判别式模型。但判别式模型不能采样得到一个服从潜在联合分布的样本对 (x, y) 。在一些应用中，生成一个样本是很重要的。因此，我们可以对联合分布 $p(X, Y)$ 进行建模，这就导致了生成式模型。

就分类而言，通常会转而对先验分布 $p(Y)$ 和类条件分布 $p(X | Y)$ 进行建模。由于

$$p(X, Y) = p(Y)p(X | Y) \tag{9.1}$$

这相当于对 $p(X, Y)$ 进行建模。

判别式模型是适用于不直接从联合分布中进行采样，并且在实践中它通常比生成式模型具有更高的分类精度。但是，如果我们的目标是对数据的生成过程进行建模而非分类的话，一个生成式模型就是必要的。

在不考虑概率的情况下，直接找到分类边界（也被称为判别函数）有时甚至能比判别式模型产生更好的结果。

9.1.3 参数化模型和参数估计

一般地，参数模型具有如下形式：

$$\mathfrak{F} = \{f(x; \theta) : \theta \in \Theta\},$$

其中， θ 表示在参数空间 Θ 中取值的未知参数（或参数向量）。如果 θ 是向量，但仅关心其中的一个元素的时候，则称其他参数为冗余参数。

例 9.1.1. （一维参数估计）令 X_1, \dots, X_n 为相互独立的 $Bernoulli(p)$ 观察值，问题是如何估计参数 p 。

例 9.1.2. （二维参数估计）假设 $X_1, \dots, X_n \sim F$ 并假设 $PDF f \in \mathfrak{F}$ ，其中 \mathfrak{F} 在 12.1 式中给出。这种情况下就有两个参数 μ 和 σ ，目标是根据数据去估计这两个参数，如果仅关心估计 μ 的值，则 μ 就是感兴趣的参数而 σ 就是冗余参数。

参数化模型估计的方法一般有：矩估计、极大似然估计、极大后验估计、贝叶斯推断。这些方法将在后面进行介绍。

9.1.4 非参数模型和非参数估计

从上节已知，一般地，参数模型具有如下形式：

$$\mathfrak{F} = \{f(x; \theta) : \theta \in \Theta\},$$

非参数模型指一些不能用有限个参数表示的 \mathfrak{F} ，例如 $\mathfrak{F}_{\text{所有}} = \{\text{所有 CDF}\}$ 就是非参数模型。

例 9.1.3. （CDF 的非参数估计）令 X_1, \dots, X_n 是来源于 CDF 为 F 的独立观察值，问题是在假设 $F \in \mathfrak{F}_{\text{所有}}\{\text{所有 CDF}\}$ 的前提下如何去估计 F 。

例 9.1.4. （非参数密度估计）令 X_1, \dots, X_n 是来源于 CDF 为 F 的独立观察值，令 $f = F'$ 为 PDF。假设要估计 $PDF f$ 。如果仅假设 $F \in \mathfrak{F}_{\text{所有}}$ 是不可能估计 f 的，需要假设 f 的光滑性，例如，假设 $f \in \mathfrak{F} = \mathfrak{F}_{\text{DENS}} \cap \mathfrak{F}_{\text{SOB}}$ ，其中， $\mathfrak{F}_{\text{DENS}}$ 表示所有密度函数的集合

$$\mathfrak{F}_{\text{SOB}} = \{f : \int (f''(x))^2 dx < \infty\} \quad (9.2)$$

集合 $\mathfrak{F}_{\text{SOB}}$ 称为索伯列夫空间（Sobolev space），它表示一系列“波动不大”的函数的集合。

例 9.1.5. （函数的非参数估计）令 $X_1, \dots, X_n \sim F$ 。假定要在仅假设 μ 存在的条件下估计 $\mu = T(F) = \int x dF(x)$ ，通常情况下，任何 F 的函数称为统计泛函，其他一些统计泛函的例子有方差 $T(F) = \int x^2 dF(X) - (\int x dF(X))^2$ ，中位数 $T(F) = F^{-1}(1/2)$ 。

非参数化模型估计的方法一般有：直方图估计、核密度估计、非参数回归估计、CDF 和统计泛函的估计。

9.1.5 概率图模型

机器学习算法经常会涉及多元随机向量的概率分布，如果采用单个函数来描述整个随机变量的联合分布是非常低效的（无论是计算上还是统计上），因为这些随机变量中涉及到的直接相互作用通常只介于非常少的变量之间的。利用随机变量之间的条件独立性关系，可以将随机变量的联合分布分解为一些因式的乘积，得到简洁的概率表示。

我们可以采用图论中的“图”的概率来表示这种分解，得到概率图模型：图中的节点表示随机变量，边表示随机变量之间的直接作用。有向图和无向图均可以用于概率表示。

有向图模型

有向概率图模型使用有向边连接不同的结点，这些有向边通常表示了结点间的因果关系，在有向概率图模型中，隐马尔可夫模型、贝叶斯网络和动态贝叶斯网络被广泛的使用。

有向图模型 (directed graphical models,DGMs) 使用带有有向边的图，用条件概率分布来表示分解：每个随机变量 x_i 都包含着一个影响因子，这些影响因子被称为 x_i 的父节点，记为 $Pa(x_i)$ ：

$$p(\mathbf{x}) = \prod_i p(x_i | Pa(x_i))$$

例 9.1.6. 图9.2中对应的概率分布可以分解为

$$p(a, b, c, d, e) = p(a)p(b|a)p(c|a, b)p(d|b)p(e|c)$$

从图模型可以快速看出此分布的一些性质：如 a 和 c 直接相互影响，但 a 和 e 只有通过 c 间接相互影响。

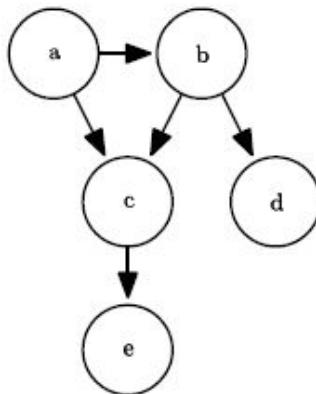


图 9.2: 有向图模型例子

禁
止
复
制
传
播

无向图模型

无向概率图模型使用一个无向图建立随机变量之间的关系模型。无向链接通常捕捉一对结点之间的相互依赖关系。马尔可夫随机场和条件随机场是两种无向概率图模型，其被广泛应用于图像处理、目标识别、图像分割和纹理合成等计算机视觉领域。

无向图模型 (undirected graphical models, UGM) 使用带有无向边的图，它将联合概率表示分解成一组函数的乘积；图中任何满足两两之间有边连接的顶点的集合称成为团 (clique)，每个团 C^i 都伴随着一个因子 $\phi^i(C^i)$ ，每个因子的输出都必须是非负的，但不像概率分布中那样要求因子的和/积为 1。

随机向量的联合概率与所有这些因子的乘积成比例：

$$p(\mathbf{x}) = \frac{1}{Z} \prod_i \phi^i(C^i)$$

其中归一化常数 Z 被定义为 ϕ 函数乘积的所有求和或积分，使得这些乘积的求和为 1($p(\mathbf{x})$ 为一个合法的概率分布)。

例 9.1.7. 图9.3中对应的概率分布可以分解为

$$p(a, b, c, d, e) = \frac{1}{Z} \phi^1(a, b, c) \phi^2(b, d) \phi^3(c, e)$$

从图中可以快速看出此分布的一些性质：如 a 和 c 直接相互影响，但 a 和 e 只有通过 c 简洁相互影响。

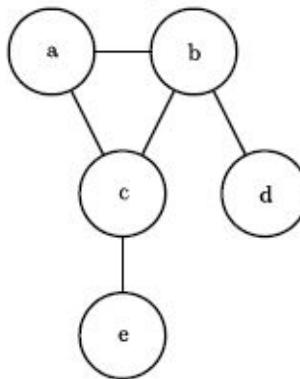


图 9.3: 无向图模型例子

注意：这些模型表示的分解仅仅是描述概率分布的一种语言，它们不是相互排斥的概率分布族，任何概率分布都可以用这两种方式进行描述。

9.1.6 统计推断的基本概念

许多统计推断问题都可以归入以下三类：估计，置信集和假设检验。在本书的余下章节将对这三类问题详细讨论，这里给出这些思想的简单介绍。

点估计

点估计指对感兴趣的某一单点提供“最优估计”。感兴趣的点可以是参数模型、分布函数 F 、概率密度函数 f 和回归函数 r 等中的某一参数，或者可以是对某些随机变量的未来值 Y 的预测。

为简化起见，记 θ 的点估计为 $\hat{\theta}$ 或 $\hat{\theta}_n$ ，记住 θ 是固定且未知的，而估计 $\hat{\theta}$ 依赖于数据，所以它是随机的。

一般地，令 X_1, \dots, X_n 为从某分布得来的 n 个IID数据点，参数 θ 的点估计 $\hat{\theta}_n$ 是 X_1, \dots, X_n 的函数：

$$\hat{\theta}_n = g(X_1, \dots, X_n), \quad (9.3)$$

估计量的偏差定义为

$$bias(\hat{\theta}_n) = \mathbb{E}_{\theta}(\hat{\theta}_n) - \theta. \quad (9.4)$$

如果 $\mathbb{E}_{\theta}(\hat{\theta}_n) = \theta$ ，则称 $\hat{\theta}_n$ 是无偏的，无偏性在以前备受关注，但如今无偏性已经不被看重了；许多估计量都是有偏的。对估计量的一个合理要求是当收集的数据越来越多的时候，它将收敛于真实的参数值。这一要求见如下定义：

定义 9.1.1. 如果 $\hat{\theta}_n \xrightarrow{P} \theta$ ，则参数 θ 的点估计 $\hat{\theta}_n$ 是相合的。

$\hat{\theta}_n$ 的分布称为抽样分布， $\hat{\theta}_n$ 的标准差称为标准误差，记为 se ，

$$se = se(\hat{\theta}_n) = \sqrt{\mathbb{V}(\hat{\theta}_n)}. \quad (9.5)$$

通常，标准误差依赖于未知分布 F ，在另外一些情况下， se 是未知量，估计的标准误记为 \hat{se} 。

例 9.1.8. 令 $X_1, \dots, X_n \sim Bernoulli(p)$ ， $\hat{p}_n = n^{-1} \sum_i X_i$ ，则 $\mathbb{E}(\hat{p}_n) = n^{-1} \sum_i \mathbb{E}(X_i) = p$ ，所以 \hat{p}_n 是无偏的，标准误差为 $se = \sqrt{\mathbb{V}(\hat{p}_n)} = \sqrt{p(1-p)/n}$ ，估计的标准误差为 $\hat{se} = \sqrt{\hat{p}(1-\hat{p})/n}$

点估计的质量好坏有时用均方误差或MSE来评价，均方误差定义为

$$MSE = \mathbb{E}_{\theta}(\hat{\theta}_n - \theta)^2. \quad (9.6)$$

需注意 $\mathbb{E}_{\theta}(\cdot)$ 是关于如下分布的期望而不是关于 θ 分布的平均，该分布由数据得来，具体见下

$$f(x_1, \dots, x_n; \theta) = \prod_{i=1}^n f(x_i; \theta). \quad (9.7)$$

定理 9.1.1. MSE 可写成如下形式：

$$MSE = bias^2(\hat{\theta}_n) + \mathbb{V}_{\theta}(\hat{\theta}_n).R \quad (9.8)$$

证明. 令 $\bar{\theta}_n = E_\theta(\hat{\theta}_n)$, 则

$$\begin{aligned}\mathbb{E}_\theta(\hat{\theta}_n - \theta)^2 &= \mathbb{E}_\theta(\hat{\theta}_n - \bar{\theta}_n + \bar{\theta}_n - \theta)^2 \\ &= \mathbb{E}_\theta(\hat{\theta}_n - \bar{\theta}_n)^2 + 2(\bar{\theta}_n - \theta)\mathbb{E}_\theta(\hat{\theta}_n - \bar{\theta}_n) + \mathbb{E}_\theta(\bar{\theta}_n - \theta)^2 \\ &= (\bar{\theta}_n - \theta)^2 + \mathbb{E}_\theta(\hat{\theta}_n - \bar{\theta}_n)^2 \\ &= \text{bias}^2(\hat{\theta}_n) + V_\theta(\hat{\theta}_n)\end{aligned}$$

推导过程中用到了规则: $\mathbb{E}_\theta(\hat{\theta}_n - \bar{\theta}_n) = \bar{\theta}_n - \bar{\theta}_n = 0$

□

定理 9.1.2. 如果 $\text{bias} \rightarrow 0$ 且当 $n \rightarrow \infty$ 时 $se \rightarrow 0$, 则 $\hat{\theta}_n$ 是相合的, 即 $\hat{\theta}_n \xrightarrow{P} \theta$.

证明. 如果 $\text{bias} \rightarrow 0$ 且 $se \rightarrow 0$, 则根据定理 6.9 有 $MSE \rightarrow 0$, 推出 $\hat{\theta}_n \xrightarrow{qm} \theta$ (定义 5.2), 再根据定理 5.4 的(a)即得证. □

例 9.1.9. 回到抛硬币的例子中, 因为 $E_p(\hat{p}_n) = p$, 所以 $\text{bias} = p - p = 0$, $se = \sqrt{p(1-p)/n} \rightarrow 0$, 因此 $\hat{p}_n \xrightarrow{P} p$, 即 \hat{p}_n 是一致估计量.

今后将遇到的许多估计量都近似服从正态分布。

定义 9.1.2.

$$\frac{\hat{\theta}_n - \theta}{se} \sim +N(0, 1)$$

则称估计量 $\hat{\theta}_n$ 是渐进正态的。

置信集

参数 θ 的 $1 - \alpha$ 置信区间为区间 $C_n = (a, b)$, 其中, $a = a(X_1, \dots, X_n)$, $b = b(X_1, \dots, X_n)$ 是数据的函数, 满足

$$P_\theta(\theta \in C_n) \geq 1 - \alpha, \quad \theta \in \Theta$$

其含义为 (a, b) 覆盖参数的概率为 $1 - \alpha$, 称 $1 - \alpha$ 为置信区间的覆盖。

注意! C_n 是随机的而 θ 是固定的。

通常, 人们喜欢用 95% 的置信区间, 相应的 $\alpha = 0.05$, 如果 θ 是向量则用置信集(例如, 球面或者椭圆面)代替置信区间。

注意! 关于如何解释置信区间很容易让人迷惑, 置信区间不是对 θ 的概率陈述, 因为 θ 是固定的而不是随机变量。一些教科书将置信区间解释如下: 如果反复的重复试验, 置信区间将有 95% 的机会可以包括参数。该解释并没错误, 但用处不大, 因为人们很少反复地多次重复相同地试验, 更好的解释如下:

第1次，对于参数 θ_1 ，收集到数据并建立了95%的置信区间；第2次，对于参数 θ_2 ，收集到数据并建立了95%的置信区间；第3次，对于参数 θ_3 ，收集到数据并建立了95%的置信区间。继续这过程，对一系列不相关参数 $\theta_1, \theta_2, \dots$ 建立置信区间，则这些置信区间有95%的概率覆盖真实的参数值，这一解释不需要反复地重复同一试验。

例9.1.10. 报纸每天都会报道民意调查的结果。例如，报道称“有83%的公众对飞行员随身配备真枪飞行的做法表示赞同”，通常你还会看到诸如这样的陈述“该调查有95%的概率在4个百分点的范围内变动”。意思就是赞同飞行员随身配备真枪飞行的做法的人数所占的比例 p 的95%的置信区间是83%±4%，如果以后都按这种方式建立置信区间，则有95%的区间将包括真实的参数值，即使每天估计的量不同（不同的民意测验），这一结论也是正确的。

例9.1.11. 置信区间不是参数 θ 的概率陈述容易让人迷惑，考察(Berger and Wolpert, 1984)中的一个例子，令 θ 为一固定且已知的实数， X_1, X_2 为独立随机变量，满足 $\mathbb{P}(X_i = 1) = \mathbb{P}(X_i = -1) = 1/2$ ，定义 $Y_i = \theta + X_i$ 并假设只观察到了 Y_1 和 Y_2 ，定义如下“置信区间”（该区间其实只包括了一个点）：

$$C = \begin{cases} Y_1 - 1, & Y_1 = Y_2 \\ (Y_1 + Y_2)/2, & Y_1 \neq Y_2 \end{cases}$$

可以验证不管 θ 为多少都有 $\mathbb{P}_\theta(\theta \in C) = 3/4$ ，所以这是一个75%的置信区间，假设重做试验得到 $Y_1 = 15, Y_2 = 17$ ，则以上的75%的置信区间为{16}，然而可以确信 $\theta = 16$ ，如果希望对 θ 进行概率陈述，可能有 $\mathbb{P}(\theta \in C|Y_1, Y_2) = 1$ ，这与称{16}是75%的置信区间并没有什么矛盾，但它并不是关于 θ 的置信区间。第11章将介绍当 θ 为随机变量时的贝叶斯方法以及关于 θ 的概率陈述，特别地，将做这样的陈述“在给定数据的情况下， θ 在 C_n 中的概率为95%”，然而，贝叶斯区间指的是可信度的可能性，一般来讲，贝叶斯区间不满足有95%的概率会覆盖参数。

例9.1.12. 在抛硬币的试验中，令 $C_n = (\hat{p}_n - \epsilon, \hat{p}_n + \epsilon)$ ，其中 $\epsilon^2 = \log(2/\alpha)/(2n)$ ，由霍夫不等式得，对任意 p

$$\mathbb{P}(p \in C_n) \geq 1 - \alpha.$$

因此， C_n 是 $1 - \alpha$ 置信区间。

就像前面提到的那样，点估计通常具有极限正态分布的，这意味着式成立，即 $\hat{\theta}_n \approx N(\theta, \hat{s}e^2)$ ，在这种情况下，可以通过如下方式建立（近似）置信区间。

例9.1.13. （基于正态的置信区间）假设 $\hat{\theta}_n \approx N(\theta, \hat{s}e^2)$ ，令 Φ 为标准正态分布的CDF， $z_{\alpha/2} = \Phi^{-1}(1 - (\alpha/2))$ ，即 $\mathbb{P}(Z > z_{\alpha/2}) = \alpha/2, \mathbb{P}(-z_{\alpha/2} < Z < z_{\alpha/2}) = 1 - \alpha$ ，其中 $Z \sim N(0, 1)$ ，令

$$C_n = (\hat{\theta}_n - z_{\alpha/2} \hat{s}e, \hat{\theta}_n + z_{\alpha/2} \hat{s}e),$$

则

$$\mathbb{P}_\theta(\theta \in C_n) \sim 1 - \alpha.$$

证明. 令 $Z_n = (\hat{\theta}_n - \theta) / \hat{se}$, 根据假设由 $Z_n \rightarrow Z$, 其中, $Z \sim N(0, 1)$, 因此

$$\mathbb{P}_{\theta}(\theta \in C_n) = \mathbb{P}_{\theta}(\hat{\theta}_n - z_{\alpha/2} \hat{se} < \theta < \hat{\theta}_n + z_{\alpha/2} \hat{se}) \quad (9.9)$$

$$= \mathbb{P}_{\theta}\left(-z_{\alpha/2} < \frac{\hat{\theta}_n - \theta}{\hat{se}} < z_{\alpha/2}\right) \quad (9.10)$$

$$\rightarrow \mathbb{P}(-z_{\alpha/2} < Z < z_{\alpha/2}) \quad (9.11)$$

$$= 1 - \alpha \quad (9.12)$$

□

对于 95% 的置信区间, $\alpha = 0.05, z_{\alpha/2} = 1.96 \approx 2$, 可以得到 95% 的置信区间为 $\hat{\theta}_n \pm 2\hat{se}$ 。

例 9.1.14. 令 $X_1, \dots, X_n \sim Bernoulli(p)$, $\hat{p}_n = n^{-1} \sum_{i=1}^n X_i$, 则 $\mathbb{V}(\hat{p}_n) = n^{-2} \sum_{i=1}^n \mathbb{V}(X_i) = n^{-2} \sum_{i=1}^n p(1-p) = n^{-2} np(1-p)/n = p(1-p)/n$, 因此 $se = \sqrt{p(1-p)/n}$, $\hat{se} = \sqrt{\hat{p}_n(1-\hat{p}_n)/n}$, 根据中心极限定理有 $\hat{p}_n \approx N(p, \hat{se}^2)$, 从而, 近似的 $1 - \alpha$ 置信区间为

$$\hat{p}_n \pm z_{\alpha/2} \hat{se} = \hat{p}_n \pm z_{\alpha/2} \sqrt{\frac{\hat{p}_n(1-\hat{p}_n)}{n}},$$

与例比较可知, 基于正态的区间较短, 它仅有近似的 (大样本) 正确覆盖。

假设检验

在假设检验中, 从缺省理论, 即原假设开始, 通过数据是否提供显著性证据来支持拒绝该假设, 如果不能拒绝, 则保留原假设。

例 9.1.15. (检验硬币是否均匀) 令

$$X_1, \dots, X_n \sim Bernoulli(p)$$

为 n 此独立的硬币投掷结果, 假设要检验硬币是否均匀, 令 H_0 表示硬币是均匀的假设, H_1 表示硬币不是均匀的假设, H_0 称为原假设, H_1 称为备择假设, 可以将假设写成

$$H_0 : p = 1/2 \quad vs \quad H_1 : p \neq 1/2.$$

如果 $T = |\hat{p}_n - (1/2)|$ 的值很大, 则有理由拒绝 H_0 , 当详细讨论假设检验的时候, 将会确定出拒绝 H_0 的精确 T 值。

9.2 参数估计

我们最终想要得到的是一个概率密度的模型。如果对观测的对象的概率密度分布已经了解了, 只是需要确定其中的参数而已, 这种情况就是属于参数估计问题。如果不清楚观测的对象的数据符合什么模型, 参数估计的方法就失效了, 只有用非参数估计的办法去估计真实数据符合的概率密度模型。

因此, 接下来主要讨论参数估计和非参数估计问题。而在参数估计中, 本节主要介绍了极大似然估计方法和极大后验估计。

9.2.1 矩估计

讨论的第一种参数估计方法为矩估计法。可以看出这些估计并不是最优的，但是最容易计算。它们也可以作为其他需要循环几次的算法的初始值。

假设参数 $\theta = (\theta_1, \dots, \theta_k)$ 有 k 个元素。对于 $1 \leq j \leq k$, 定义 j 阶矩为

$$\alpha_j \equiv \alpha_j(\theta) = E_\theta(X^j) = \int x^j dF_\theta(x). \quad (9.13)$$

而 j 阶样本矩为

$$\hat{\alpha}_j = \frac{1}{n} \sum_{i=1}^n X_i^j. \quad (9.14)$$

定义 9.2.1. θ 的矩估计定义为 $\hat{\theta}_n$, 使得

$$\begin{aligned} \alpha_1(\hat{\theta}_n) &= \hat{\alpha}_1, \\ \alpha_2(\hat{\theta}_n) &= \hat{\alpha}_2, \\ &\vdots \\ \alpha_k(\hat{\theta}_n) &= \hat{\alpha}_k. \end{aligned} \quad (9.15)$$

公式(9.15)定义了带有 k 个未知参数的 k 个方程的方程组。

例 9.2.1. 令 $X_1, \dots, X_n \sim Bernoulli(p)$ 。则 $\alpha_1 = E_p(X) = p$ 且 $\hat{\alpha}_1 = n^{-1} \sum_{i=1}^n X_i$ 。让它们相等可以得到估计值

$$\hat{p}_n = \frac{1}{n} \sum_{i=1}^n X_i.$$

例 9.2.2. 令 $X_1, \dots, X_n \sim Normal(\mu, \sigma^2)$ 。则 $\alpha_1 = E_\theta(X) = \mu$ 且 $\alpha_2 = E_\theta(X_1^2) = V_\theta(X_1) + (E_\theta(X))^2 = \sigma^2 + \mu^2$ 。现在需要解下述方程：

$$\begin{aligned} \hat{\mu} &= \frac{1}{n} \sum_{i=1}^n X_i, \\ \hat{\sigma}^2 + \hat{\mu}^2 &= \frac{1}{n} \sum_{i=1}^n X_i^2. \end{aligned}$$

这是由两个方程组成含有两个未知参数的方程组。它的解为

$$\hat{\mu} = \bar{X},$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2.$$

9.2.2 极大似然估计

参数估计是通过数据对模型参数进行推测的一种方法。极大似然估计 (MLE) 是一种常用的参数估计方式。极大似然估计可以用于确定具有分布的模型的参数。

若总体 X 属离散型, 其分布律 $P(X = x) = p(x; \theta), \theta \in \Theta$ 的形式为已知, θ 为待估参数, Θ 是 θ 可能取值的范围, 设 X_1, X_2, \dots, X_n 是来自 X 的样本, 则 X_1, X_2, \dots, X_n 的联合分布律为

$$\prod_{i=1}^n p(x_i; \theta)$$

又设 x_1, x_2, \dots, x_n 是相应于样本 X_1, X_2, \dots, X_n 值。易知样本 X_1, X_2, \dots, X_n 取到观察值 x_1, x_2, \dots, x_n 的概率, 亦即事件 $\{X_1 = x_1, X_2 = x_2, \dots, X_n = x_n\}$ 发生的概率为

$$L(\theta) = L(x_1, x_2, \dots, x_n; \theta) = \prod_{i=1}^n p(x_i; \theta), \theta \in \Theta$$

这一概率随 θ 的取值而变化, 它是 θ 的函数, $L(\theta)$ 称为样本的似然函数(注意, 这里 x_1, x_2, \dots, x_n 是已知的样本值, 它们都是常数)。

关于最大似然估计法, 我们有以下的直观想法: 现在已经取到样本值 x_1, x_2, \dots, x_n 了, 这表明取到这一样本值的概率 $L(\theta)$ 比较大。我们当然不会考虑那些不能使样本 x_1, x_2, \dots, x_n 出现的 $\theta \in \Theta$ 作为 θ 的估计, 再者, 如果已知当 $\theta = \theta_0 \in \Theta$ 时使 $L(\theta)$ 取很大值, 而 Θ 中的其他 θ 的值使 $L(\theta)$ 取很小值, 我们自然认为取 θ 取 θ_0 作为未知参数 θ 的估计值, 较为合理。由费希尔引进的最大似然估计法, 就是固定样本观察值 x_1, x_2, \dots, x_n , 在 θ 取值的可能范围 Θ 内挑选使似然函数 $L(x_1, x_2, \dots, x_n; \theta)$ 达到最大的参数值 $\hat{\theta}$, 作为参数 θ 的估计值, 即取 $\hat{\theta}$ 使

$$L(x_1, x_2, \dots, x_n; \hat{\theta}) = \max_{\theta \in \Theta} L(x_1, x_2, \dots, x_n; \theta)$$

这样得到的 $\hat{\theta}$ 与样本值 x_1, x_2, \dots, x_n 有关, 常记为 $\hat{\theta}(x_1, x_2, \dots, x_n; \theta)$, 称为参数 θ 的最大似然估计值, 而相应的统计量 $\hat{\theta}(X_1, X_2, \dots, X_n)$ 称为参数 θ 的最大似然估计量。

若总体 X 属连续型, 其概率密度 $f(x; \theta), \theta \in \Theta$ 的形式为已知, θ 为待估参数, Θ 是 θ 可能取值的范围, 设 X_1, X_2, \dots, X_n 是来自 X 的样本, 则 X_1, X_2, \dots, X_n 的联合密度为

$$\prod_{i=1}^n f(x_i; \theta)$$

设 x_1, x_2, \dots, x_n 是相应于样本 X_1, X_2, \dots, X_n 值。则随机点 (X_1, X_2, \dots, X_n) 落在点 x_1, x_2, \dots, x_n 的邻域(边长分别为 dx_1, dx_2, \dots, dx_n 的 n 维立方体)内的概率近似地为

$$\prod_{i=1}^n f(x_i; \theta) dx_i$$

其值 θ 的变化而变化。与离散值的情况一样, 我们取 θ 的估计值 $\hat{\theta}$ 使概率取到最大值, 但因子 $\prod_{i=1}^n dx_i$ 不随 θ 而变, 故只需考虑函数

$$L(\theta) = L(x_1, x_2, \dots, x_n; \theta) = \prod_{i=1}^n f(x_i; \theta)$$

的最大值。这里 $L(\theta)$ 称为样本的似然函数，若

$$L(x_1, x_2, \dots, x_n; \hat{\theta}) = \max_{\theta \in \Theta} L(x_1, x_2, \dots, x_n; \theta)$$

则称 $\hat{\theta}(x_1, x_2, \dots, x_n; \theta)$ 为数 θ 的最大似然估计值，称 $\hat{\theta}(X_1, X_2, \dots, X_n)$ 为 θ 的最大似然估计量。

定义 9.2.2. 令 $\mathbf{X}_1, \dots, \mathbf{X}_n$ 独立同分布于概率密度函数 $p(x|\theta)$ 。似然函数定义为

$$\mathcal{L}(\theta) = p(\mathcal{D}|\theta) = \prod_{i=1}^n p(x_i|\theta) \quad (9.16)$$

有时也记为 $\mathcal{L}(\theta|\mathcal{D})$ ，表示似然函数为在给定数据 \mathcal{D} 的情况下，参数 θ 的函数。

定义 9.2.3. 极大似然估计 MLE，记为 $\hat{\theta}_n$ ，是使得 $\mathcal{L}(\theta)$ 最大的 θ 值。

似然函数在数值上是数据的联合密度，但它是参数 θ 的一个函数。 $\mathcal{L}_n : \theta \rightarrow [0, \infty)$ 。一般来说， $\mathcal{L}(\theta)$ 关于 θ 的积分并不等于 1。因此似然函数并不是一个密度函数。

我们把极大似然估计 (MLE)，记 $\hat{\theta}$ ，是使得 $\mathcal{L}(\theta)$ 最大的 θ 的值。即

$$\hat{\theta} = \arg \max_{\theta} \mathcal{L}(\theta)$$

对数 (log) 似然函数为 $l(\theta) = \log \mathcal{L}(\theta)$ 。 $l(\theta)$ 和 $\mathcal{L}(\theta)$ 在同一个点取得最大值，因此，最大化对数似然函数就可以最大化似然函数。通常，对数似然函数求解要容易一点。

注 将 $\mathcal{L}_n(\theta)$ 乘以一个正常数 c (它并不依赖于 θ)，并不会改变极大似然估计 MLE。因此，经常去掉似然函数的常数。

因为最大化

$$l(\theta) = \sum_{i=1}^N \log p(x_i|\theta)$$

等价于最小化

$$-l(\theta) = \sum_{i=1}^N -\log p(x_i|\theta)$$

因此可以将负 log 似然函数作为损失函数，在训练集上训练使其最小。

9.2.3 常见分布的极大似然参数估计

本节主要介绍了如何使用极大似然估计法对一些常见分布的参数进行估计。

高斯分布

首先给出以下高斯分布的概率密度函数。其中 μ 为均值， σ^2 为方差。

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

令 $x_1, \dots, x_N \sim \mathcal{N}(\mu, \sigma^2)$, 参数为 μ, σ^2 , 似然函数为

$$\begin{aligned}\ell(\mu, \sigma^2) &= \sum_{i=1}^N \log p(x_i | \mu, \sigma) \\ &= -\frac{1}{2\sigma^2} \sum_{i=1}^N (x_i - \mu)^2 - \frac{N}{2} \log \sigma^2 - \frac{N}{2} \log 2\pi \\ &= -\frac{NS^2}{2\sigma^2} - \frac{N(\bar{x} - \mu)^2}{2\sigma^2} - N \log \sigma - \frac{N}{2} \log 2\pi\end{aligned}$$

其中 $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$ 为样本均值, $S^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$ 为样本方差

$$\sum_{i=1}^n (x_i - \mu)^2 = \sum_{i=1}^N (x_i - \bar{x} + \bar{x} - \mu)^2 = NS^2 + N(\bar{x} - \mu)^2$$

对 \log 似然函数求极值点, 即分别对 μ 和 σ 求一阶导数为 0。解方程

$$\begin{cases} \frac{\partial \ell(\mu, \sigma)}{\partial \mu} = \frac{N(\bar{x} - \mu)}{\sigma^2} = 0 \\ \frac{\partial \ell(\mu, \sigma)}{\partial \sigma} = -\frac{N}{\sigma} + \frac{NS^3}{\sigma^3} = 0 \end{cases}$$

得到

$$\begin{cases} \hat{\mu} = \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \\ \hat{\sigma}^2 = S^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 = \left(\frac{1}{N} \sum_{i=1}^N x_i^2 \right) - \bar{x}^2 \end{cases}$$

可以证明, 这是似然函数的全局最大值。

Bernoulli 分布

首先给出 Bernoulli 分布的概率密度函数

$$\text{Ber}(x|\theta) = \theta^x (1-\theta)^{1-x}$$

假设我们投掷硬币 N 次, 并记录每次投掷结果的序列, 用 $\mathcal{D} = x_1, \dots, x_N$ 表示, 则概率函数为 $\text{Ber}(x_i|\theta)$ 。

似然函数为

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^N \log \text{Ber}(x_i|\theta) \\ &= \sum_{i=1}^N \log(\theta^{x_i} (1-\theta)^{1-x_i}) = N_1 \log \theta + N_2 \log(1-\theta)\end{aligned}$$

其中

$$\begin{cases} N_1 = \sum_{i=1}^N x_i & \text{实验中结果为 1 的次数} \\ N_2 = \sum_{i=1}^N (1 - x_i) & \text{实验中结果为 0 的次数} \end{cases}$$

所以

$$\frac{\partial \ell(\theta)}{\partial \theta} = \frac{N_1}{\theta} - \frac{N_2}{1-\theta} = 0 \implies \hat{\theta} = \frac{N_1}{N_1 + N_2} = \frac{N_1}{N}$$

Binomial 分布

$$\text{Bin}(x|n; \theta) = \binom{n}{x} \theta^x (1-\theta)^{n-x}$$

共进行 N 次实验，第 i 次实验中抛掷了 n_i 次硬币，其中 x_i 枚硬币正面朝上。

则似然函数为

$$\begin{aligned} \mathcal{L} &= \prod_{i=1}^N \text{Bin}(x_i|n_i; \theta) \\ &= \prod_{i=1}^N \binom{n_i}{x_i} \theta^{x_i} (1-\theta)^{n_i - x_i} \propto \theta^{N_1} (1-\theta)^{N_2} \end{aligned}$$

其中

$$\begin{cases} N_1 = \sum_{i=1}^N x_i \\ N_2 = \sum_{i=1}^N (n_i - x_i) \end{cases}$$

对似然函数取对数：

$$\log \mathcal{L} \propto N_1 \log \theta + N_2 \log(1-\theta)$$

求导数为 0 的点：

$$\frac{N_1}{\theta} - \frac{N_2}{1-\theta} = 0$$

解得：

$$\hat{\theta} = \frac{N_1}{N_1 + N_2}$$

参数估计值与 Bernoulli 分布的估计一样。

Multinoulli 分布

$$Mu(x|N, \theta) = \binom{N}{x_1 \cdots x_K} \prod_{k=1}^K \theta_k^{x_k}$$

$$\binom{N}{x_1 \cdots x_K} = \frac{N!}{x_1! \cdots x_K!}$$

假设我们投掷一个有 K 面的骰子，共进行了 N 次试验，并记每次投掷结果的序列，用 $\mathcal{D} = x_1, \dots, x_N$ 表示， $x_i \in 1, \dots, K$ ，则似然函数为：

$$l(\theta) = \log p(\mathcal{D}|\theta) = \sum_{k=1}^K N_k \log \theta_k$$

其中 $N_k = \sum_{i=1}^N \mathbf{1}(x_i = k)$ 表示 N 次此试验中出现 k 的次数，这是带有约束 $\sum_{k=1}^K \theta_k = 1$ 的优化问题，采用拉格朗日乘子法，得到

$$l(\theta, \lambda) = \sum_{k=1}^K N_k \log \theta_k + \lambda(1 - \sum_{k=1}^K \theta_k)$$

分别对 λ 和 θ_k 求偏导并令其等于 0，得到

$$\begin{cases} \frac{\partial l(\theta, \lambda)}{\partial \lambda} = 1 - \sum_{k=1}^K \theta_k = 0 \\ \frac{\partial l(\theta_k, \lambda)}{\partial \theta} = \frac{N_k}{\theta_k} - \lambda = 0 \end{cases}$$

因此：

$$\theta_k \propto N_k$$

解得：

$$\hat{\theta}_k = \frac{N_k}{N}$$

线性回归

回归是监督学习问题，正态分布可用于回归系统噪声建模，输入 $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ ，输出 y 为连续型变量，学习映射 $f: \mathcal{X} \rightarrow \mathcal{Y}$ 。

假设 $y = f(x) + \varepsilon$ ，残差服从正态分布： $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ 。那么有 $y|x \sim \mathcal{N}(y|f(x), \sigma^2)$ 。

对测试集上的一个新样本下，预测其输出 $\hat{y} = f(x)$ ，即正态分布 $y|x \sim \mathcal{N}(f(x), \sigma^2)$ 的期望。

最简单的回归模型是线性模型，我们假设

$$\begin{aligned} y &= f(x) + \varepsilon \\ &= w^T x + \varepsilon \end{aligned}$$

其中 w 称为权重向量， ε 为线性预测和真值之间的残差。

由于 $y|x \sim \mathcal{N}(f(x), \sigma^2)$ ，则 $p(y|x; \theta) \sim \mathcal{N}(y; w^T x, \sigma^2)$ 其中模型的参数为 $\theta = (w, \sigma^2)$

极大似然估计定义为

$$\hat{\theta} = \arg \max_{\theta} \log p(\mathcal{D}|\theta)$$

其中似然函数

$$\ell(\theta) = \log p(\mathcal{D}|\theta) = \sum_{i=1}^N \log p(y_i|x_i; \theta)$$

极大似然可等价地写成极小负 log 似然损失 (negative log likelihood, NLL)

$$\text{NLL}(\theta) = \sum_{i=1}^N -\log p(y_i|x_i; \theta)$$

将概率模型 $p(y_i|x_i, w, \sigma^2) = \mathcal{N}(y_i|w^T x_i, \sigma^2)$ 代入, 似然函数为

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^N \log \left[\left(\frac{1}{2\pi\sigma^2} \right)^{1/2} \exp \left(-\frac{1}{2\sigma^2} (y_i - w^T x_i)^2 \right) \right] \\ &= -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \underbrace{\sum_{i=1}^N (y_i - w^T x_i)^2}_{\text{RSS}(w)}\end{aligned}$$

其中 RSS 表示残差平方和 (residual sum of squares), RSS/N 为平均平方误差 (MSE), 也可以写成残差向量的 L2 模, 即

$$\text{RSS}(w) = |\varepsilon|_2^2 = \sum_{i=1}^N \varepsilon_i^2, \quad \varepsilon_i = y_i - w^T x_i$$

将 NLL 写成矩阵形式

$$\begin{aligned}\text{NLL}(w, \sigma) &= \frac{N}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - w^T x_i)^2 \\ &= \frac{N}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} (y - Xw)^T (y - Xw)\end{aligned}$$

只取与 w 有关的项, 得到

$$\text{NLL}(W) = w^T (X^T X) w - 2w^T (X^T y)$$

求梯度为 0 的点

$$\frac{\partial}{\partial w} \text{NLL}(w) = w X^T X w - 2X^T y = 0 \implies X^T X w = X^T y$$

$$\hat{w}_{OLS} = (X^T X)^{-1} X^T y$$

其中 OLS 指的是普通最小二乘 (Ordinary least squares)

对参数 σ

$$\text{NLL}(\hat{w}, \sigma) = \frac{N}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} (y - X\hat{w})^T (y - X\hat{w})$$

$$\frac{\partial}{\partial \sigma} \text{NLL}(\hat{w}, \sigma^2) = \frac{N}{\sigma} - \frac{1}{\sigma^3} (y - X\hat{w})^T (y - X\hat{w}) = 0$$

得到

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{w}^T x_i)^2 = \frac{1}{N} (y - X\hat{w})^T (y - X\hat{w})$$

当样本数目 N 较小时，可采用 OLS 结论，用矩阵 QR 分解分解得到优化解。

当样本数目 N 较大时，可采用随机梯度下降方法优化求解（略）

极大似然估计的特征有：

- (1) 极大似然估计是相合估计： $\hat{\theta}_n \xrightarrow{P} \theta_*$ ，其中， θ_* 表示参数 θ 的真实值。
- (2) 极大似然估计是同变估计：如果 $\hat{\theta}_n$ 是 θ 的极大似然估计，则 $g(\hat{\theta}_n)$ 是 $g(\theta)$ 的极大似然估计。
- (3) 极大似然估计是渐近正态的： $(\hat{\theta}_n - \theta_*) / \hat{s}\epsilon \sim N(0, 1)$ 。同时，估计的标准差 $\hat{s}\epsilon$ 可以解出来。
- (4) 极大似然估计是渐近最优或有效的：这表示，在所有表现优异的估计中，极大似然估计的方差最小，至少对大样本这肯定成立。
- (5) 极大似然估计接近于贝叶斯估计。

9.2.4 极大后验估计

极大后验估计的基本思想

假设有一所学校，学生中 60% 是男生和 40% 是女生。女生穿裤子与裙子的数量相同；所有男生穿裤子。现在有一个观察者，随机从远处看到一名学生，因为很远，观察者只能看到该学生穿的是裤子，但不能从长相发型等其他方面推断被观察者的性别。那么该学生是女生的概率是多少？用事件 G 表示观察到的学生是女生，用事件 T 表示观察到的学生穿裤子。于是，现在要计算的是条件概率 $P(G|T)$ ，我们需要知道：

$P(G)$ 表示一个学生是女生的概率。女生在全体学生中的占比是 40%，所以概率是 $P(G) = 0.4$ 。这是先验概率。 $P(T|G)$ 是在女生中穿裤子的概率，女生穿裙子和穿裤子各占一半，所以 $P(T|G) = 0.5$ 。这也就是在给定 G 的条件下， T 事件的概率。 $P(T|B)$ 是在男生中穿裤子的概率，这个值是 1。 $P(T)$ 是学生穿裤子的概率，即任意选一个学生，在没有其他信息的情况下，该名学生穿裤子的概率。根据全概率公式： $P(T) = \sum_{i=1}^n P(T|A_i)P(A_i) = P(T|G)P(G) + P(T|B)P(B)$ ，计算得到 $P(T) = 0.5 \times 0.4 + 1 \times 0.6 = 0.8$ 。

根据贝叶斯公式

$$P(A_i|T) = \frac{P(T|A_i)P(A_i)}{\sum_{i=1}^n P(T|A_i)P(A_i)} = \frac{P(T|A_i)P(A_i)}{P(T)}$$

基于以上所有信息，如果观察到一个穿裤子的学生，并且是女生的概率是

$$P(G|T) = \frac{P(T|G)P(G)}{P(T)} = 0.5 \times 0.4 \div 0.8 = 0.25.$$

下面给出极大后验估计的理论知识：

先验概率 (Prior probability) 在贝叶斯统计中，先验概率分布，即关于某个变量 X 的概率分布，是在获得某些信息或者依据前，对 X 之不确定性所进行的猜测。这是对不确定性(而不是随机性)赋予一个量化的数值的表征，这个量化数值可以是一个参数，或者是一个潜在的变量。先验概率仅仅依赖于主观上的经验估计，也就是事先根据已有的知识的推断。例如， X 可以是投一枚硬币，正面朝上的概率，显然在我们未获得任何其他信息的条件下，我们会认为 $P(X) = 0.5$ ；再比如上面例子中的， $P(G) = 0.4$ 。

似然函数 (Likelihood Function) 似然函数也称作似然，是一个关于统计模型参数的函数。也就是这个函数中自变量是统计模型的参数。对于观测结果 x ，在参数集合 θ 上的似然，就是在给定这些参数值的基础上，观察到的结果的概率 $L(\theta) = P(x|\theta)$ 。也就是说，似然是关于参数的函数，在参数给定的条件下，对于观察到的 x 的值的条件分布。似然函数在统计推断中发挥重要的作用，因为它是关于统计参数的函数，所以可以用来对一组统计参数进行评估，也就是说在一组统计方案的参数中，可以用似然函数做筛选。

后验概率 (Posterior probability) 后验概率是关于随机事件或者不确定性断言的条件概率，是在相关证据或者背景给定并纳入考虑之后的条件概率。后验概率分布就是未知量作为随机变量的概率分布，并且是在基于实验或者调查所获得的信息上的条件分布。后验概率是关于参数 θ 在给定的证据信息 X 下的概率，即 $P(\theta|X)$ 。若对比后验概率和似然函数，似然函数是在给定参数下的证据信息 X 的概率分布，即 $P(X|\theta)$ 。

后验概率与似然函数关系 二者有如下关系：

我们用 $P(\theta)$ 表示概率分布函数，用 $P(X|\theta)$ 表示观测值 X 的似然函数。后验概率定义为 $P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$ ，注意这也是贝叶斯定理所揭示的内容。鉴于分母是一个常数，上式可以表达成如下比例关系(而且这也是我们更多采用的形式)：

后验概率 \propto 似然 \times 先验概率。

常见分布的极大后验估计

高斯先验 考虑偏向小的系数值，从而得到比较平滑的曲线的 0 均值高斯先验 $w_j \sim N(0, \tau^2)$

$$p(\mathbf{w}) = \prod_{j=1}^D N(w_j|0, \tau^2) \propto \exp\left(-\frac{1}{2\tau^2} \sum_{j=1}^D w_j^2\right) = \exp\left(-\frac{1}{2\tau^2} [\mathbf{w}^T \mathbf{w}]\right)$$

其中 $1/\tau^2$ 控制先验的强度。

此时针对一个样本的似然函数为: $p(y_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) = N((y_i|\mathbf{w}^T \mathbf{x}_i, \sigma^2)$

针对整个数据集的似然函数为:

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \mathbf{w}, w_0, \sigma^2) &= N(\mathbf{y}|\mathbf{X}\mathbf{w} + w_0\mathbf{1}_N, \sigma^2\mathbf{1}_N) \\ &\propto \exp\left(-\frac{1}{2\sigma^2}[(y - (\mathbf{X}\mathbf{w} + w_0\mathbf{1}_N))^T(y - (\mathbf{X}\mathbf{w} + w_0\mathbf{1}_N))]\right) \end{aligned}$$

由贝叶斯公式知后验概率为:

$$p(\mathbf{w}, w_0|\mathbf{X}, \mathbf{y}, \sigma^2) \propto \exp\left(-\frac{1}{2\sigma^2}[(y - \mathbf{X}\mathbf{w} - w_0\mathbf{1}_N)^T(y - \mathbf{X}\mathbf{w} - w_0\mathbf{1}_N)] - \frac{1}{2\tau^2}[\mathbf{w}^T \mathbf{w}]\right)$$

则极大后验估计等价于最小化的目标函数如下:

$$\begin{aligned} J(\mathbf{w}) &= \sum_{i=1}^N (y_i - (\mathbf{w}^T \mathbf{x}_i + w_0))^2 + \lambda \|\mathbf{w}\|_2^2 \\ &= (\mathbf{y} - (\mathbf{X}\mathbf{w} + w_0 + \mathbf{1}_N))^T(\mathbf{y} - (\mathbf{X}\mathbf{w} + w_0 + \mathbf{1}_N)) + \lambda \mathbf{w}^T \mathbf{w} \end{aligned}$$

其中 $\lambda = \sigma^2/\tau^2$

这种形式称为岭回归, 或正则化的最小二乘。注意 w_0 没有被正则 (w_0 只影响函数的高度, 不影响复杂性)。

Laplace 先验 Laplace 分布:

$$\text{Lap}(x|\mu, b) = \frac{1}{2b} \exp\left\{-\frac{|x - \mu|}{b}\right\}$$

假设线性回归中参数的先验为 Laplace 先验:

$$p(\mathbf{W}|\lambda) = \prod_{j=1}^D \text{Lap}(w_j|0, \frac{1}{\lambda}) \propto \prod_{j=1}^D \exp(-\lambda |w_j|) = \exp(-\lambda \sum_{j=1}^D |w_j|)$$

似然为:

$$p(y_i|\mathbf{x}_i, \mathbf{W}, \sigma^2) = \mathcal{N}(y_i|\mathbf{w}^T \mathbf{x}_i, \sigma^2) \propto \exp\left(-\frac{1}{2\sigma^2}(y_i - (\mathbf{w}^T \mathbf{x}_i + w_0))^2\right)$$

后验为:

$$p(\mathbf{w}, w_0|\mathbf{X}, \mathbf{y}, \sigma^2) \propto \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - (\mathbf{w}^T \mathbf{x}_i + w_0))^2 - \sum_{j=1}^D \lambda |w_j|\right)$$

极大后验估计 MAP 等价于 L1 正则的线性回归 (Lasso):

$$J(\mathbf{w}) = \underbrace{\sum_{i=1}^N (y_i - (\mathbf{w}^T \mathbf{x}_i + w_0))^2}_{RSS(\mathbf{w})} + \lambda \underbrace{\|\mathbf{w}\|}_{\text{正则项复杂性惩罚}}$$

当 λ 取合适值时, \mathbf{w} 变得稀疏 (有些系数为 0), 但是相比岭回归, 优化计算更复杂。

9.2.5 贝叶斯推断

回顾之前讲过的贝叶斯公式：

$$P(\theta | x) = \frac{P(x | \theta)\pi(\theta)}{P(x)} = \frac{P(x | \theta)\pi(\theta)}{\int P(x | \theta)\pi(\theta)d\theta}$$

若要求一个未知概率分布，该分布由参数 θ 决定，根据经验，若能估计 θ 可能的取值，即 θ 的概率分布，我们就能解决该问题。 θ 的概率分布称之为先验分布 $\pi(\theta)$ ，另外， $P(\theta | x)$ 称为后验分布。当这个后验分布和先验分布是同一个分布时，我们称先验分布和似然函数为共轭分布，也就是我们先验分布假设的比较准确。在介绍共轭分布前我们先介绍一下 Gamma 函数和 Beta 函数。

Gamma 函数 Gamma 函数 $\Gamma(x)$ 定义为

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$$

通过分部积分法，可以很容易证明 Gamma 函数具有如下之递归性质

$$\Gamma(x+1) = x\Gamma(x)$$

也是便很容易发现，它还可以看做是阶乘在实数集上的延拓，即

$$\Gamma(x) = (x-1)!$$

Beta 函数 定义 Beta 函数如下

$$\mathbf{B}(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

Beta 函数的另外一种定义形式为（注意这两种定义是等价的）

$$\mathbf{B}(a, b) = \int_0^1 t^{a-1} (1-t)^{b-1} dt$$

Beta 分布 Beta 分布的概率密度函数 (PDF) 定义为：

$$Beta(\theta|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{a-1} (1-\theta)^{b-1}$$

或

$$Beta(\theta|a, b) = \frac{1}{\mathbf{B}(a, b)} \theta^{a-1} (1-\theta)^{b-1}$$

Beta 分布的均值和方差分别有下面两式给出

$$E[\theta] = \frac{a}{a+b}$$

$$\text{var}[\theta] = \frac{ab}{(a+b)^2(a+b+1)}$$

$\text{Beta}(\theta|a,b) = \frac{1}{B(a,b)}\theta^{a-1}(1-\theta)^{b-1}$ 可见，Beta 分布有两个控制参数 a 和 b ，而且当这两个参数取不同值时，Beta 分布的 PDF 图形可能会呈现出相当大的差异。如下图9.4所示。

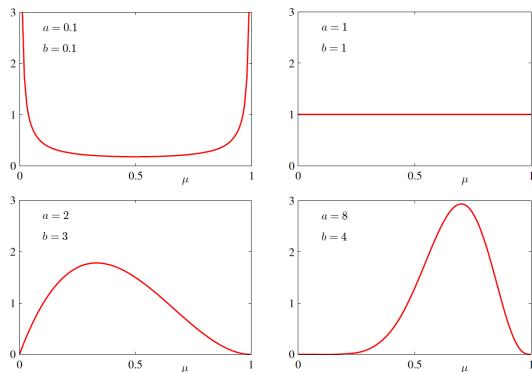


图 9.4: 参数 a 和 b 变化时 Beta 分布的 PDF 图

共轭分布 假如你有一个硬币，它有可能是不均匀的，所以投这个硬币有 θ 的概率抛出正面，有 $(1-\theta)$ 的概率抛出背面。如果抛了五次这个硬币，有三次是正面，有两次是背面，完全根据目前观测的结果来估计 θ ，那么显然你会得出结论 $\theta = \frac{3}{5}$ 。点估计的方法有漏洞。实验次数较少，估计结果可能有较大偏差。如果抛了五次都是正面，以后永远都抛出正面么？

在贝叶斯学派看来，参数 θ 不是一个固定的值，而满足一定的概率分布。在估计 θ 时，我们心中可能有一个根据经验的估计，即先验概率， $P(\theta)$ 。而给定一系列实验观察结果 X 的条件下，我们可以得到后验概率为 $P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$ 在上面的贝叶斯公式中， $P(\theta)$ 就是个概率分布。

这个概率分布可以是任何概率分布，比如高斯分布，或者刚刚提过的 Beta 分布。下图9.5是 Beta(5,2) 的概率分布图。如果我们将这个概率分布作为 $P(\theta)$ ，那么我们在还未抛硬币前，便认为 θ 很可能接近于 0.8。

使用 Beta 分布的原因：

虽然 $P(\theta)$ 可以是任何种类的概率分布，但是如果使用 Beta 分布，会让之后的计算更加方便（稍后说明）。通过调节 Beta 分布中的 a 和 b ，你可以让这个概率分布变成各种你想要的形状！ $P(X|\theta)$ 是个二项分布。继续以前面抛 5 次硬币抛出 3 次正面的观察结果为例， $X =$ 抛 5 次硬币 3 次结果为正面的事件，则 $P(X|\theta) = C_5^2\theta^3(1-\theta)^2$ 。

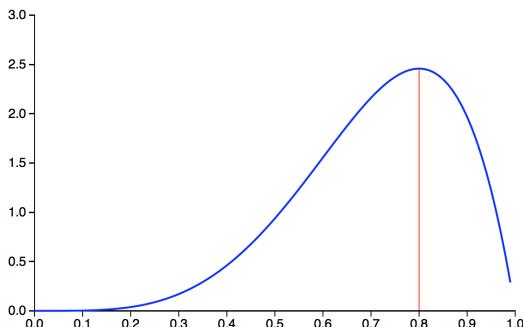


图 9.5: Beta(5,2) 的概率分布图

而如果我们采用 Beta 分布, θ 的概率分布在 $[0,1]$ 之间是连续的, 用积分, 即 $P(X) = \int_0^1 P(X|\theta)P(\theta)d\theta$, $P(\theta)$ 是个 Beta 分布, 那么在观测到“X=抛 5 次硬币中出现 3 个正面”的事件后, $P(\theta|X)$ 依旧是个 Beta 分布! 这是使用 Beta 分布方便计算的原因。

$$\begin{aligned}
 P(\theta|X) &= \frac{P(X|\theta)P(\theta)}{P(X)} = \frac{P(X|\theta)P(\theta)}{\int_0^1 P(X|\theta)P(\theta)d\theta} \\
 &= \frac{C_5^2 \theta^3 (1-\theta)^2 \frac{1}{B(a,b)} \theta^{a-1} (1-\theta)^{b-1}}{\int_0^1 C_5^2 \theta^3 (1-\theta)^2 \frac{1}{B(a,b)} \theta^{a-1} (1-\theta)^{b-1} d\theta} \\
 &= \frac{\theta^{(a+3-1)} (1-\theta)^{(b+2-1)}}{\int_0^1 \theta^{(a+3-1)} (1-\theta)^{(b+2-1)} d\theta} \\
 &= \frac{\theta^{(a+3-1)} (1-\theta)^{(b+2-1)}}{B(a+3, b+2)} \\
 &= Beta(\theta|a+3, b+2)
 \end{aligned}$$

当我们得知 $P(\theta|X) = Beta(\theta|a+3, b+2)$ 后, 我们就只要根据 Beta 分布的特性, 得出 θ 最有可能等于多少了, (即 θ 等于多少时, 观测后得到的 Beta 分布有最大的概率密度)。

例如下图9.6, 仔细观察新得到的 Beta 分布, 和上一图中的概率分布对比, 发现峰值从 0.8 左右的位置移向了 0.7 左右的位置。Bayesian 方法和普通的统计方法不同的地方: 我们结合自己的先验概率和观测结果来给出预测。

共轭性 后验概率分布 (正比于先验和似然函数的乘积) 拥有与先验分布相同的函数形式。这个性质被叫做共轭性 (Conjugacy)。

共轭先验 如果后验概率分布和先验概率分布有相同的形式 (如同为指数族分布), 则后验概率分布和先验概率分布统称共轭分布。那么先验概率 $p(\theta)$ 称为似然函数的共轭先验。

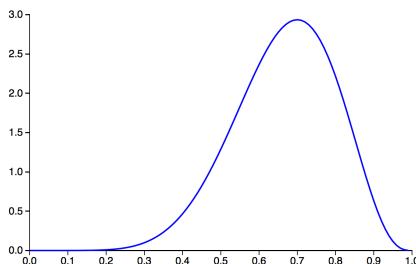


图 9.6: 新的 Beta 分布

共轭先验使得后验概率分布的函数形式与先验概率相同，因此使得贝叶斯分析得到了极大的简化。例如，二项分布的参数之共轭先验就是我们前面介绍的 Beta 分布。多项式分布的参数之共轭先验则是 Dirichlet 分布，高斯分布的均值之共轭先验是另一个高斯分布。

9.3 非参数估计

9.3.1 直方图估计

一种非参数的概率估计方式是直方图

定义 9.3.1. 直方图可以定义为：

$$\hat{f}_n(x) = \begin{cases} \hat{p}_1/h, & x \in B_1 \\ \hat{p}_2/h, & x \in B_2 \\ \dots \\ \hat{p}_m/h, & x \in B_m \end{cases}$$

或者可以写的更简洁：

$$\hat{f}_n(x) = \sum_{j=1}^m \frac{\hat{p}_j}{h} I(x \in B_j)$$

例 9.3.1. 将输入空间划分为 M 个箱子 (bin)，箱子的宽度为 $h = 1/M$ 则这些箱子为 $B_1 = [0, 1/M], B_2 = [1/M, 2/M], \dots, B_M = [(M-1)/M, 1]$ 计算落入箱子 b 中的样本的数目 V_b ，落入箱子 b 的比率为 $\hat{p}_b = V_b/N$ 则直方图估计为

$$\hat{p}(x) = \sum_{b=1}^M \frac{\hat{p}_b}{h} \mathbf{1}(x \in B_b) = \frac{1}{N} \sum_{b=1}^M \frac{v_b}{h} \mathbf{1}(x \in B_b)$$

其中 $\mathbf{1}(x \in B_b)$ 表示当 $x \in B_b$ 时其值为 1，否则为 0 核密度估计。

9.3.2 核密度估计

直方图是不连续的。核密度估计较光滑且比直方图估计较快地收敛到真正的密度。

定义 9.3.2. 给定一个核 K 与一个正数 h , 称作带宽, 核密度估计定义为

$$\hat{f}_n(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x - X_i}{h}\right)$$

其中, 参数 h 称为带宽 (bandwidth), 核函数可为任意平滑的函数 K , 满足

$$\begin{aligned} K(u) &> 0, & \int K(u)du &= 1 \\ \int uK(u)du &= 0, & \sigma_K^2 &= \int u^2 K(u)du > 0 \end{aligned}$$

令 X_1, \dots, X_n 表示观测数据, 它们来自 f 的一个样本。在本章中, 核定义为任意一个光滑函数 K 使得 $K(x) \geq 0$, $\int K(x)dx = 1$, $\int xK(x)dx = 0$ 并且 $\sigma_K^2 = \int x^2 K(x)dx > 0$ 。核的两个例子分别为 Epanechnikov 核

$$K(x) = \begin{cases} \frac{3}{4} \left(\frac{1-x^2}{5} \right) / \sqrt{5}, & |x| < \sqrt{5} \\ 0, & \text{其他} \end{cases}$$

与高斯 (正态) 核

$$K(x) = (2\pi)^{-1/2} e^{-x^2/2}$$

定理 9.3.1. 在 f 和 K 的弱假设下,

$$R(f, \hat{f}_n) \approx \frac{1}{4} \sigma_K^4 h^4 \int (f''(x))^2 dx + \frac{\int K^2(x)dx}{nh}$$

其中, $\sigma_K^2 = \int x^2 K(x)dx$ 。最优的带宽为

$$h^* = \frac{c_1^{-2/5} c_2^{1/5} c_3^{-1/5}}{n^{1/5}}$$

其中, $c_1 = \int x^2 K(x)dx$, $c_2 = \int K(x)^2 dx$ 且 $c_3 = \int (f''(x))^2 dx$

9.3.3 非参数回归估计

考虑点对 $(x_i, Y_i), \dots, (x_n, Y_n)$, 其关系为

$$Y_i = r(x_i) + \epsilon_i$$

其中, $E(\epsilon_i) = 0$, $r(x_i) = E(Y|X)$ 。感兴趣的是如何求出 $r(x_i)$ 。

存在很多非参数回归估计。大多数涉及通过对 Y 取某种加权平均来估计 $r(x)$, 对靠近 x 的点给予更高的权重。一个常用的估计就是所谓 Nadaraya-Watson 核估计。

定义 9.3.3. Nadaraya-Watson 核估计定义为

$$\hat{r}(x) = \sum_{i=1}^n w_i(x) Y_i$$

其中, K 为一个核且其权重 $w_i(x)$ 由下式给出:

$$w_i(x) = \frac{K\left(\frac{x-x_i}{h}\right)}{\sum_{j=1}^n K\left(\frac{x-x_j}{h}\right)}$$

核估计还有另一种写法:

$$r(x) = \mathbb{E}(Y | X = x) = \int y f(y | x) dy = \frac{\int y f(x, y) dy}{\int f(x, y) dy}$$

例 9.3.2. 图 9.7 给出了宇宙微波背景 (CMB) 数据的拟合情况. 该数据包含了 n 对观察值 $(x_1, Y_1), \dots, (x_n, Y_n)$, 其中, x_i 称作多极矩, Y_i 称作温度变化功率谱估计. 所看到的是宇宙微波背景辐射中的声波, 这是从宇宙大爆炸中留下来的. 若令 $r(z)$ 表示真正的功率谱, 则

$$Y_i = r(x_i) + \epsilon_i$$

其中, ϵ_i 是一个均值为 0 的随机误差. $r(z)$ 峰值的位置和大小为了解早期宇宙的状况提供了有价值的线索. 图 9.7 给出了基于交叉验证的拟合, 既有一个欠光滑的拟合也有一个过光滑的拟合. 交叉验证拟合表明了三个定义好的峰值的存在, 恰如大爆炸的物理学理论所预测的那样.

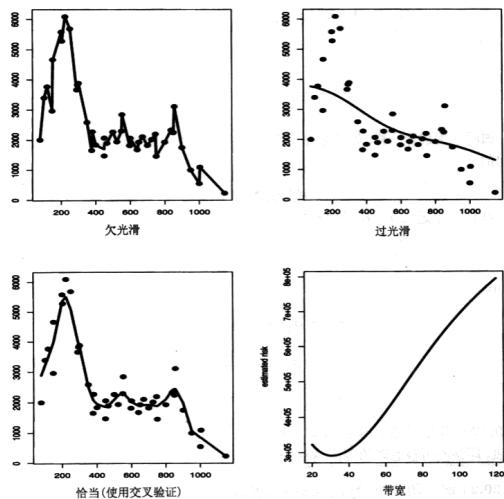


图 9.7: CMB 数据的回归分析

9.3.4 CDF 和统计泛函的估计

令 $X_1, \dots, X_n \sim F$ 为 IID 样本, 其中, F 为实直线上的分布函数, 将用经验分布函数估计 F , 定义如下:

定义 9.3.4. 经验分布函数 E 指在每一个数据点 X_i 上的概率密度为 $\frac{1}{n}$ 的 CDF, 用公式表示为

$$\hat{F}_n(x) = \frac{\sum_{i=1}^n I(X_i \leq x)}{n}$$

其中,

$$I(X_i \leq x) = \begin{cases} 1, & X_i \leq x \\ 0, & X_i > x \end{cases}$$

例 9.3.3. (神经数据) Cox 和 Lewis(1966) 报告了一种神经两次起搏之间的等待时间, 共有 799 个数据。图 9.8 为经验的 $CDF \hat{F}_n$, 数据点以垂直直线体现在图的底部。假设要估计等待时间在 0.4 到 0.6 秒之间的概率, 估计值为 $\hat{F}(0.6) - \hat{F}(0.4) = 0.93 - 0.84 = 0.09$ 。

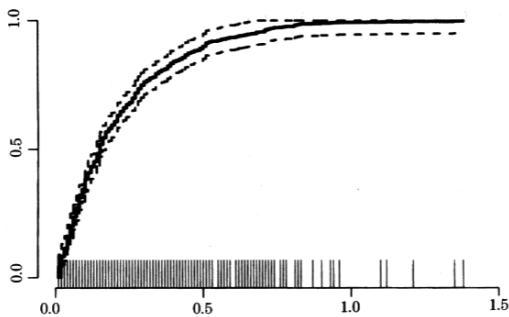


图 9.8: 神经数据

定理 9.3.2. 在任意固定点 x 有

$$\mathbb{E}(\hat{F}_n(x)) = F(x)$$

$$\mathbb{V}(\hat{F}_n(x)) = \frac{F(x)(1-F(x))}{n}$$

$$\text{MSE} = \frac{F(x)(1-F(x))}{n} \rightarrow 0$$

$$\hat{F}_n(x) \xrightarrow{P} F(x)$$

定理 9.3.3. (Glivenko-Cantelli 定理) $X_1, \dots, X_n \sim F$, 则

$$\sup_x |\hat{F}_n(x) - F(x)| \xrightarrow{P} 0$$

草稿勿外传

定理 9.3.4. (Dvoretzky-Kiefer-Wolfowitz(DKW) 不等式) 令 $X_1, \dots, X_n \sim F$, 则对任意 $\epsilon > 0$ 有

$$\mathbb{P} \left(\sup_x |F(x) - \hat{F}_n(x)| > \epsilon \right) \leq 2e^{-2n\epsilon^2}$$

通过 DKW 不等式, 可以按如下方式建立置信集:

定义:

$$L(x) = \max \left\{ \hat{F}_n(x) - \epsilon_n, 0 \right\}$$

$$U(x) = \min \left\{ \hat{F}_n(x) + \epsilon_n, 1 \right\}$$

其中,

$$\epsilon = \sqrt{\frac{1}{2n} \log \left(\frac{2}{\alpha} \right)}$$

对任意 F , 由9.3.4得

$$\mathbb{P}(\text{对所有 } x, L(x) \leq F(x) \leq U(x)) \geq 1 - \alpha$$

例 9.3.4. 图9.8的虚线给出了 95% 置信带, 其中, $\epsilon_n = \sqrt{\frac{1}{2n} \log \left(\frac{2}{0.05} \right)} = 0.048$

统计泛函 $T(F)$ 是 F 的任意函数, 例如, 均值 $\mu = \int x dF(x)$, 方差 $\sigma^2 = \int (x - \mu)^2 dF(x)$, 中位数 $m = F^{-1}(1/2)$ 。

定义 9.3.5. $\theta = T(F)$ 的嵌入式估计量定义为

$$\hat{\theta}_n = T(\hat{F}_n)$$

换言之, 就是用经验分布函数 \hat{F}_n 代替未知函数 F 。

定义 9.3.6. 如果对函数 $r(x)$ 有 $T(F) = \int r(x)dF(x)$, 则称 T 为线性泛函。

函数 $T(F) = \int r(x)dF(x)$ 被称为线性泛函的理由是 T 满足

$$T(aF + bG) = aT(F) + bT(G)$$

因此 T 在它的自变量范围内是线性的。回忆前面的介绍, 在连续情形下, $\int r(x)dF(x)$ 定义为 $\int r(x)df(x)$, 在离散情形下, $\int r(x)dF(x)$ 定义为 $\sum_j r(x_j) f(x_j)$ 。经验 CDF \hat{F}_n 是离散的, 在每一个数据点 X_i 的概率密度为, 因此, 如果 $T(F) = \int r(x)dF(x)$ 为线性泛函, 则有:

定理 9.3.5. 线性泛函 $T(F) = \int r(x)dF(x)$ 的嵌入式估计量为

$$T(\hat{F}_n) = \int r(x)d\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^n r(X_i)$$

有时可以通过计算求得 $T(\hat{F}_n)$ 的估计标准误差 $\hat{s}\epsilon$ 。然而，在有些情况下，标准误差的估计并不是很显而易见的，下一章将讨论求 $\hat{s}\epsilon$ 的一般方法，本章，假设可以求得 $\hat{s}\epsilon$ 。很多情况下，如下结论成立：

$$T(\hat{F}_n) \approx N(T(F), \hat{s}\epsilon^2)$$

容易得到 $T(F)$ 的近似 $1 - \alpha$ 的置信区间为

$$T(\hat{F}_n) \pm z_{\alpha/2} \hat{s}\epsilon$$

称该区间为基于正态的置信区间，对于 95% 的置信区间。 $z_{\alpha/2} = z_{0.05/2} = 1.96 \approx 2$ ，所以区间为

$$T(\hat{F}_n) \pm 2\hat{s}\epsilon$$

例 9.3.5. (均值) 令 $\mu = T(F) = \int x dF(x)$ ，则均值的嵌入式估计量为 $\hat{\mu} = \int x d\hat{F}_n(x)$ ，标准误差 $s\epsilon = \sqrt{V(\bar{X}_n)} = \sigma/\sqrt{n}$ ，如果 $\hat{\sigma}$ 表示 σ 的估计，则估计的标准误差为 $\hat{\sigma}/\sqrt{n}$ ，的基于正态的置信区间为 $\bar{X}_n \pm z_{\alpha/2} s\epsilon$ 。

9.4 概率模型的图语言描述

概率图模型刻画了随机变量间不同的条件独立关系。有向概率图模型通常用来表示随机变量间的因果关系，而无向概率图模型用来建立随机变量间的空间相互关系或者是相互依赖性。有向概率图和无向概率图只能表示随机变量的同一类关系，而混合概率图模型却可以表示不同类型的关系。

9.4.1 条件独立性

一个有向图是由一系列的节点及连接节点的有向边组成的。图9.9给出了一个有向图的例子。图在表示变量间的独立性关系方面是很有用处的，还可以用来代替反事实去表示因果关系。

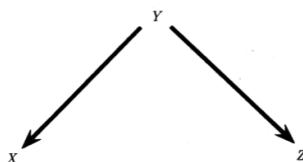


图 9.9: 节点集为 $V = X, Y, Z$ 且边集为 $E = (Y, X), (Y, Z)$ 的一个有向图

一个被赋予某种概率分布的有向图常被称为贝叶斯网络。频率学派或贝叶斯学派的方法都可以用来对有向图进行统计推断，所以贝叶斯网络这个说法是有歧义的。在进行关于有向非循环图(DAGs)的讨论之前，需要先讨论一下条件独立性。

定义 9.4.1. 令 X, Y 和 Z 为随机变量。在给定 Z 的条件下, X 和 Y 称为条件独立的, 记作 $X \perp\!\!\!\perp Y | Z$, 如果下式对于所有的 x, y 和 z 均成立,

$$f_{X,Y|Z}(x,y|z) = f_{X|Z}(x|z)f_{Y|Z}(y|z)$$

一个等价的定义为

$$f(x|y,z) = f(x|z)$$

定理 9.4.1. 下列各蕴含关系成立:

$$\begin{aligned} X \perp\!\!\!\perp Y | Z &\Rightarrow Y \perp\!\!\!\perp X | Z \\ X \perp\!\!\!\perp Y | Z \text{ 且 } U = h(X) &\Rightarrow U \perp\!\!\!\perp Y | Z \\ X \perp\!\!\!\perp Y | Z \text{ 且 } U = h(X) &\Rightarrow X \perp\!\!\!\perp Y | (Z, U) \\ X \perp\!\!\!\perp Y | Z \text{ 且 } X \perp\!\!\!\perp W | (Y, Z) &\Rightarrow X \perp\!\!\!\perp (W, Y) | Z \\ X \perp\!\!\!\perp Y | Z \text{ 且 } X \perp\!\!\!\perp Z | Y &\Rightarrow X \perp\!\!\!\perp (Y, Z). \end{aligned}$$

9.4.2 DAGs

一个有向图 \mathcal{G} 是由节点集 V 及连接一对有序节点的边集 E 组成的。每个节点对应一个随机变量。若 $(X, Y) \in E$, 则存在一条有向边从 X 指向 Y 。见图9.9。

若一条有向边连接两个随机变量 X 和 Y (取任意一个方向), 就称 X 和 Y 是邻接的。若一条有向边从 X 指向 Y , 则称 X 是 Y 的母节点, 而 Y 是 X 的子节点。 X 的所有母节点的集合记作 π_X 或 $\pi(X)$ 。两变量间的一条 c 是由一系列的同方向的有向边构成的, 如下所示:

$$X \rightarrow \cdots \rightarrow Y$$

一个从 X 开始至 Y 结束的邻接节点的序列, 但是忽略其有向边的方向性, 就称该序列为一个无向路。图9.9中的序列 X, Y, Z 就是一个无向路。若存在一条有向路从 X 指向 Y (或 $X = Y$), 则称 X 是 Y 的祖节点。也可以说 Y 是 X 的后裔节点。

如下形式的结构:

$$X \rightarrow Y \leftarrow Z$$

称作在 Y 处相遇。不具有该种形式的结构称作不相遇, 例如,

$$X \rightarrow Y \rightarrow Z$$

相遇的性质是依赖于路的。在图9.9中, Y 是一个在路 X, Y, Z 上的相遇, 但不是在路 X, Y, W 上的一个相遇。当指向相遇的变量不是邻接时, 就说该相遇是无保护的。一条开始和结束都在同一个变量处的有向路是一个圈。若一个有向图没有圈, 则它是非循环的。在这种情况下, 称这种图为一个有向非循环图或 DAG。以后只考虑非循环图。

令 \mathcal{G} 为一个具有节点集 $V = (X_1, \dots, X_k)$ 的 DAG。

定义 9.4.2. 若 P 为 V 的分布，它的概率函数为 f ，就说 P 是关于 \mathcal{G} 是马尔可夫的，或称 \mathcal{G} 表示 P ，若下式成立：

$$f(v) = \prod_{i=1}^k f(x_i | \pi_i)$$

其中， T_i 为 X_i 的母节点。由 \mathcal{G} 表示的分布集记为 $M(\mathcal{G})$ 。

例 9.4.1. 对于图 9.10 中的 DAG 来说， $\mathbb{P} \in M(\mathcal{G})$ 当且仅当其概率函数 f 具有以下形式：

$$f(x, y, z, w) = f(x)f(y)f(z | x, y)f(w | z)$$

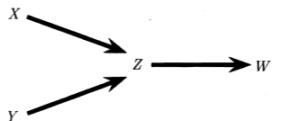


图 9.10: 另一个 DAG

下述定理表明 $\mathbb{P} \in M(\mathcal{G})$ 当且仅当马尔可夫条件成立。粗略地讲，马尔可夫条件意味着每个变量 W 在给定其母节点的情况下与“过去”是独立的。

定理 9.4.2. 一个分布 $\mathbb{P} \in M(\mathcal{G})$ 当且仅当下面的马尔可夫条件成立：对于每个变量 W ，

$$W \perp\!\!\!\perp \tilde{W} \mid \pi_W$$

其中， W 表示除了 W 的母节点和后裔节点以外的所有其他变量。

例 9.4.2. 考虑图 9.11 中的 DAG。在这种情况下，概率函数分解如下：

$$f(a, b, c, d, e) = f(a)f(b | a)f(c | a)f(d | b, c)f(e | d)$$

马尔可夫条件意味着下面的独立性关系：

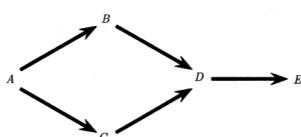


图 9.11: 另一个 DAG

$$D \perp\!\!\!\perp A \mid \{B, C\}, \quad E \perp\!\!\!\perp \{A, B, C\} \mid D \text{ 且 } B \perp\!\!\!\perp C \mid A.$$

在 DAGs 中有两个首先要考虑的估计问题。

第一，给定一个 DAGG 和来自与 \mathcal{G} 相符的分布为 f 的数据 V_1, \dots, V_n ，如何去估计 f ？

第二，给定数据 V_1, \dots, V_n ，又如何去估计 \mathcal{G} ？

第一个问题是一个纯粹的估计问题，而第二个问题则涉及到模型的选择。这些都是非常复杂的问题。这里仅简要介绍其主要思想。

通常，对于每个条件密度，人们常选择用某个参数模型 $f(x | \pi_x; \theta_x)$ ，则其似然函数为

$$\mathcal{L}(\theta) = \prod_{i=1}^n f(V_i; \theta) = \prod_{i=1}^n \prod_{j=1}^m f(X_{ij} | \pi_j; \theta_j)$$

其中， X_{ij} 是对于第 i 个数据点的 X_j 的值， θ_j 是第 j 个条件密度的参数。这样就可以通过极大似然方法来估计参数。

9.4.3 无向图

无向图也可以像有向图一样来表示独立性关系。两者的主要差异是从图中读出独立性关系的规则不同。一个无向图 $G = (V, E)$ 由一个有限节点集 V 和由每对节点组成的边或(弧)集 E 所构成。节点对应着随机变量 X, Y, Z, \dots 而边被记作一些无序对。例如， $(X, Y) \in E$ 表示 X 和 Y 通过一条边连接起来。图9.12给出了一个无向图的例子。

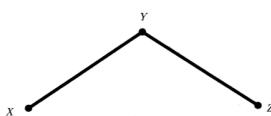


图 9.12: 节点集为 $V = \{X, Y, Z\}$ 的一个图。其边集为 $E = (Y, X), (Y, Z)$

若两个节点之间存在一条边，则称这两个节点是邻接的，记作 $X \sim Y$ 。在图9.12中， X 和 Y 是邻接的但是 X 和 Z 不是邻接的。若对每个 i 都有 $X_{i-1} \sim X_i$ ，则序列 X_0, \dots, X_n 称为一条路。在图9.12中， X, Y, Z 是一条路。若一个图中任意两个节点之间都存在一条边，则称这个图是完全的。一个子节点集 $U \in V$ 连同其边被称作一个子图。

设 A, B 和 C 是 V 的不同子集，若从 A 中的一个变量到 B 中的一个变量的路都相交于 C 中的一个变量，就说 C 分离 A 和 B 。在图9.13中， Y, W 和 Z 被 X, Y 分离。同时， W 和 Z 被 X, Y 分离。

9.4.4 概率与图

令 V 为具有分布 \mathbb{P} 的随机变量集。构造一个图，其每个节点对应 V 中的每个变量。略去一对变量之间的边若它们在给定其余变量的条件下是独立的。即

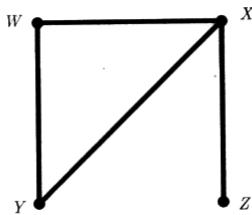


图 9.13: $\{Y, W\}$ 和 $\{Z\}$ 被 $\{X\}$ 分离。而且, W 和 Z 被 $\{X, Y\}$ 分离

X 和 Y 之间没有边 $\Leftrightarrow X \text{II} Y \mid \text{其余变量}$, 其中, “其余变量” 表示除了 X 和 Y 之外的所有其他变量。这样的图称作成对马尔可夫图。如图9.14所示: 图中暗含着一系列的成对条件独立性

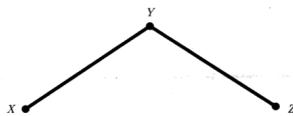


图 9.14: $X \text{II} Y \mid Z$

关系。这些关系可以推出其他的条件独立性关系。如何找到这些关系呢? 幸运的是, 也可以从图中直接读出这些其他的条件独立性关系, 如下面的定理所述。

定理 9.4.3. 令 $\mathcal{G} = (V, E)$ 是一个分布为 \mathbb{P} 的成对马尔可夫图。令 A, B 和 C 为 V 的不相同的子集使得 C 分离 A 和 B , $A \text{II} B \mid C$ 。

定理9.4.3中的独立性条件被称作全局马尔可夫性质。将看到成对和全局马尔可夫性质是等价的。把这个问题表述得更确切些。给定一个图 \mathcal{G} , 令 $M_{pair}(\mathcal{G})$ 表示满足成对马尔可夫性质的分布集, 因此 $P \in M_{pair}(\mathcal{G})$, 在分布 \mathbb{P} 下, 若 $X \text{II} Y \mid \text{其余变量}$ 当且仅当 X 和 Y 之间不存在边。令 $M_{global}(\mathcal{G})$ 为满足全局马尔可夫性质的分布集: 则 $P \in M_{global}(\mathcal{G})$, 在分布 \mathbb{P} 下, 若 AB , $A \text{II} B \mid C$ 当且仅当 C 分离 A 和 B

定理 9.4.4. 令 \mathcal{G} 为一个图, 则 $M_{pair}(\mathcal{G}) = M_{global}(\mathcal{G})$ 。

例 9.4.3. 由图9.15可知 $X \text{II} Y$, $X \text{II} Z$ 和 $X \text{II} (Y, Z)$ 。

9.4.5 团与势

若一个图的变量集中的任意两个对应的节点都是邻接的, 则称该集为一个团。若一个团任意增加一个节点后就不能成为团, 则称之为一个极大团。一个势就是任意一个正函数。在特定

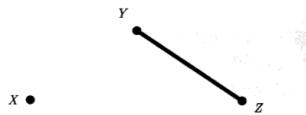


图 9.15: XIIY

的条件下, 可以证明 \mathbb{P} 关于 \mathcal{G} 是马尔可夫的当且仅当其概率函数 f 可以写为

$$f(x) = \frac{\prod_{C \in \mathcal{C}} \psi_C(x_C)}{Z}$$

其中, \mathcal{C} 是一个极大团集, ψ_C 是一个势, 且

$$Z = \sum_x \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

例 9.4.4. 图9.12中的极大团是 $C = X, Y$ 和 $C = Y, Z$ 。因此, 若 \mathbb{P} 关于该图是马尔可夫的, 则其概率函数可以写为

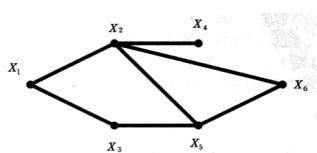
$$f(x, y, z) \propto \psi_1(x, y) \psi_2(y, z)$$

ψ_1 和 ψ_2 是某些正函数。

例 9.4.5. 图9.16中的极大团为

$$\{X_1, X_2\}, \quad \{X_1, X_3\}, \quad \{X_2, X_4\}, \quad \{X_3, X_5\}, \quad \{X_2, X_5, X_6\}$$

因此可以把概率函数写为

图 9.16: 该图的极大团为 $\{X_1, X_2\}, \quad \{X_1, X_3\}, \quad \{X_2, X_4\}, \quad \{X_3, X_5\}, \quad \{X_2, X_5, X_6\}$

$$f(x_1, x_2, x_3, x_4, x_5, x_6) \propto \psi_{12}(x_1, x_2) \psi_{13}(x_1, x_3) \psi_{24}(x_2, x_4) \times \psi_{35}(x_3, x_5) \psi_{256}(x_2, x_5, x_6)$$

9.5 统计决策理论

9.5.1 引言

前面已经考虑了几种点估计, 如极大似然估计、矩估计和后验均值。事实上, 还有许多其他的估计方法。如何选择它们呢? 答案在决策理论中找, 它是比较统计过程的正规理论。

考虑参数空间中的参数 θ . 令是 $\hat{\theta}$ 的估计. 在决策理论的语言中, 点估计有时称为决策规则, 决策规则可能的值称为行动。

用损失函数 $L(\theta, \hat{\theta})$ 来度量和的离散程度。正式地, L 把 $\Theta \times \Theta$ 映射到 R 。下面列出了一些损失函数:

$$L(\theta, \hat{\theta}) = (\theta - \hat{\theta})^2 \quad \text{平方损失,}$$

$$L(\theta, \hat{\theta}) = |\theta - \hat{\theta}| \quad \text{绝对损失,}$$

$$L(\theta, \hat{\theta}) = |\theta - \hat{\theta}|^p \quad L_p \text{损失,}$$

当 $\theta = \hat{\theta}$ 时, $L(\theta, \hat{\theta}) = 0$, 当 $\theta \neq 0$ 时为 1.0 – 1 损失,

$$L(\theta, \hat{\theta}) = \int \log \left(\frac{f(x; \theta)}{f(x; \hat{\theta})} \right) f(x; \theta) dx \quad Kullback - Leibler \text{ 损失.}$$

记住估计 $\hat{\theta}$ 是数据的函数。为了强调这一点, 有时把 $\hat{\theta}$ 记为 $\hat{\theta}(X)$. 为了衡量一个估计, 用平均风险或损失来估计。

定义估计 $\hat{\theta}$ 的风险为:

$$R(\theta, \hat{\theta}) = \mathbb{E}_\theta(L(\theta, \hat{\theta})) = \int L(\theta, \hat{\theta}(x)) f(x; \theta) dx$$

当损失函数为平方误差时, 风险是均方误差 MSE :

$$R(\theta, \hat{\theta}) = \mathbb{E}_\theta(\theta - \hat{\theta})^2 = MSE = \mathbb{V}_\theta + bias_\theta^2(\hat{\theta}).$$

本章后面部分, 如果不专门说明用了哪种损失函数, 就假定使用的是平方损失函数。

9.5.2 比较风险函数

为比较两个估计, 来比较它们的风险函数. 然而, 这并不能提供一个明确的答案说哪一个估计更好. 考虑下面的例子。

例 令 $X \sim N(\theta, 1)$, 假设使用平方损失函数. 考虑两个估计 $\hat{\theta}_1 = X$ 和 $\hat{\theta}_2 = 3$. 风险函数为 $R(\theta, \hat{\theta}) = \mathbb{E}_\theta(X - \hat{\theta})^2 = 1$ 和 $R(\theta, \hat{\theta}) = \mathbb{E}_\theta(3 - \theta)^2 = (3 - \theta)^2$. 如果 $2 < \theta < 4$, 则 $R(\theta, \hat{\theta}_1) < R(\theta, \hat{\theta}_2)$. 没有哪一个估计一定比另一个好, 见下图:

例 令 $X_1, \dots, X_n \sim Bernoulli(p)$. 考虑平方损失函数, 令 $\hat{p}_1 = \bar{X}$. 由于这是无偏的, 就有

$$R(p, \hat{p}_1) = \mathbb{V}(\bar{X}) = \frac{p(1-p)}{n}.$$

另一个估计为

$$\hat{p}_2 = \frac{Y + \alpha}{\alpha + \beta + n},$$

其中, $Y = \sum_{i=1}^n X_i$, α, β 为正常数. 这是使用先验 $Beta(\alpha, \beta)$ 的后验均值. 现在,

$$R(p, \hat{p}_2) = \mathbb{V}_p(\hat{p}_2) + (bias(\hat{p}_2))^2 \quad (9.17)$$

$$= \mathbb{V}_p\left(\frac{Y + \alpha}{\alpha + \beta + n}\right) + \left(\mathbb{E}_p\left(\frac{Y + \alpha}{\alpha + \beta + n}\right) - p\right)^2 \quad (9.18)$$

$$= \frac{np(1-p)}{(\alpha + \beta + n)^2} + \left(\frac{np + \alpha}{\alpha + \beta + n} - p\right)^2. \quad (9.19)$$

令 $\alpha = \beta = \sqrt{n/4}$ 在例 12.12 会解释这样选择的理) 得到估计为

$$\hat{p}_2 = \frac{Y + \sqrt{n/4}}{n + \sqrt{n}},$$

风险函数为

$$R(p, \hat{p}_2) = \frac{n}{4(n + \sqrt{n})^2}$$

风险函数在下图中画出. 正如所看到的, 没有哪一个估计一致的比另一个好。

这些例子说明了风险函数需要比较的要求. 为此, 需要用一个数来描述这个风险函数最大风险和贝叶斯风险就是采用这种形式定义的. 定义 定义最大风险为

$$\bar{R}(\hat{\theta}) = \sup_{\theta} R(\theta, \hat{\theta}) \quad (9.20)$$

贝叶斯风险为

$$r(f, \hat{\theta}) = \int R(\theta, \hat{\theta}) f(\theta) d\theta \quad (9.21)$$

其中, $f(\theta)$ 是 $Theta$ 的先验。

9.5.3 最小最大规则

求最小最大规则比较复杂, 在这里并不能全面讲述这一理论, 但会提到几个关键结果. 这一节传达的主要信息就是: 常数风险函数的贝叶斯估计是最小最大估计。

定理 9.5.1. 令 $\hat{\theta}^f$ 是某一先验 f 的贝叶斯规则,

$$r(f, \hat{\theta}^f) = \inf_{\hat{\theta}} r(f, \hat{\theta}). \quad (9.22)$$

假设对所有的 θ , 有

$$R(f, \hat{\theta}^f) \leq r(f, \hat{\theta}^f). \quad (9.23)$$

则 $\hat{\theta}^f$ 是最小最大估计, f 称为最不利先验。

证明. 假设 $\hat{\theta}^f$ 不是最小最大的, 则存在其他的一个规则 $\hat{\theta}_0$ 使得 $\sup_{\theta} R(\theta, \hat{\theta}_0) < \sup_{\theta} R(\theta, \hat{\theta}^f)$. 由于函数的均值总是小于等于它的最大值, 有 $r(f, \hat{\theta}_0) < \sup_{\theta} R(\theta, \hat{\theta}_0)$. 因此,

$$r(f, \hat{\theta}_0) \leq \sup_{\theta} R(\theta, \hat{\theta}_0) < \sup_{\theta} R(f, \hat{\theta}^f) \leq r(f, \hat{\theta}^f),$$

这和(9.22)矛盾。 □

定理 9.5.2. 假设 $\hat{\theta}$ 是基于先验 f 的贝叶斯估计。进一步假设 $\hat{\theta}$ 的风险为常数 $c: R(\theta, \hat{\theta}) = c$, 则 $\hat{\theta}$ 是最小最大的。

证明. 贝叶斯风险为 $r(f, \hat{\theta}) = \int R(\theta, \hat{\theta}) f(\theta) d\theta = c$, 因此, 对所有的 $\theta, R(\theta, \hat{\theta}) \leq r(\theta, \hat{\theta})$ 。再应用(9.5.1)可得结论。□

例 9.5.1. 考虑损失函数为平方损失的 *Bernoulli* 模型. 在例 12.3 中, 已经证明了估计

$$\hat{p}(X^n) = \frac{\sum_{i=1}^n X_i + \sqrt{n/4}}{n + \sqrt{n}}$$

的风险函数为一常数. 这个估计是后验均值, 因此, 对于 $\alpha = \beta = \sqrt{n/4}$ 的 $Beta(\alpha, \beta)$ 先验, 它也是贝叶斯估计. 因此, 由前面的定理, 这个估计是最小最大的。

例 9.5.2. 再次考虑 *Bernoulli* 模型, 但是它的损失函数为

$$L(p, \hat{p}) = \frac{(p - \hat{p})^2}{p(1-p)}.$$

令

$$\hat{p}(X^n) = \hat{p} = \frac{\sum_{i=1}^n X_i}{n}.$$

风险为

$$R(p, \hat{p}) = E\left(\frac{(p - \hat{p})^2}{p(1-p)}\right) = \frac{1}{p(1-p)} \frac{p(1-p)}{n} = \frac{1}{n},$$

这里, 它作为 p 的函数, 是一个常数. 可以证明, 对于这个损失函数, $\hat{p}(X^n)$ 是在先验 $f(p) = 1$ 下的贝叶斯估计。因此, \hat{p} 是最小最大的。

很自然地会想到一个问题: 什么是正态模型的最小最大估计?

定理 9.5.3. 令 $X_1, \dots, X_n \sim N(\theta, 1)$, 且令 $\hat{\theta} = \bar{X}$, 则 $\hat{\theta}$ 是关于任意优良的损失函数的最小最大规则. 它是具有这种性质的唯一估计。

如果参数空间是有限制的, 则上面的定理不适用, 正如下面的例子说明的。

例 9.5.3. 假设 $X \sim N(\theta, 1)$, 且已知 θ 在区间 $[-m, m]$ 中, 其中, $0 < m < 1$. 在平方损失函数下, 唯一的最小最大估计为

$$\hat{\theta}(X) = m \tanh(mX).$$

其中, $\tanh(z) = (e^z - e^{-z})/(e^z + e^{-z})$. 可以证明, 这是在 m 和 $-m$ 的概率分别为 $1/2$ 为先验条件下的贝叶斯估计. 而且可以证明这个风险不是常数, 但对于所有 θ , 它满足 $R(\theta, \hat{\theta}) \leq r(f, \hat{\theta})$. 见图 12.3. 因此, 由定理(9.5.1)可知 $\hat{\theta}$ 是最小最大的。

9.5.4 极大似然、最小最大和贝叶斯

对于满足弱正则性条件的参数模型，极大似然估计近似最小最大估计。考虑平方损失函数，它是偏差的平方加上方差。在大样本的参数模型中，可以证明方差项远远大于偏差项，所有极大似然估计 $\hat{\theta}$ 约等于方差

$$R(\theta, \hat{\theta}) = \mathbb{V}_\theta(\hat{\theta}) + \text{bias}^2 \approx \mathbb{V}_\theta$$

极大似然估计的方差近似为

$$\mathbb{V}(\hat{\theta}) \approx \frac{1}{nI(\theta)}$$

其中, $I(\theta)$ 是 Fisher 信息量。因此,

$$nR(\theta, \hat{\theta}) \approx \frac{1}{I(\theta)}$$

对于任意其他估计 θ' ，可以证明对于足够大的 n ，有 $R(\theta, \theta') \geq R(\theta, \hat{\theta})$ 。更精确地，这说明在局部大样本的情况下，极大似然 MLE 是最小最大的。可以证明 MLE 近似是贝叶斯规则。

总之，在绝大多数大样本参数模型中，MLE 是近似最小最大的和贝叶斯规则。

9.6 阅读材料

概率学习方法利用（并且要求）关于不同假设的先验概率以及在给定假设时观察到不同数据的概率的知识。贝叶斯方法则提供了概率学习方法的基础。贝叶斯方法还可基于这些先验和数据观察假定，赋予每个候选假设一个后验概率。贝叶斯方法可用于确定在给定数据时最可能的假设—极大后验概率（MAP）假设。它比其他的假设更可能成为最优假设。

贝叶斯最优分类器将所有假设的预测结合起来，并用后验概率加权，以计算对新实例的最可能分类。

朴素贝叶斯分类器是在许多实际应用问题中很有效的一种贝叶斯学习方法。它之所以被称为朴素的（naive）是因为它的简化假定：属性值在给定实例的分类时条件独立。

当该假定成立时，朴素贝叶斯分类器可输出 MAP 分类。即使此假定不成立，在学习分类文本的情况下，朴素贝叶斯分类通常也是很有效的。贝叶斯信念网为属性的子集上的一组条件独立性假定提供了更强的表达能力。

贝叶斯推理框架可对其他不直接应用贝叶斯公式的学习方法的分析提供理论基础。例如，在特定条件下学习一个对应于极大似然假设的实值目标函数时，它可使误差平方最小化。

最小描述长度准则建议选取这样的假设，它使假设的描述长度和给定假设下数据的描述长度的和最小化。贝叶斯公式和信息论中的基本结论可提供此准则的根据。

在许多实际的学习问题中，某些相关的实例变量是不可观察到的。EM 算法提供了一个很通用的方法，当存在隐藏变量时进行学习。该算法开始于一个任意的初始假设。

然后迭代地计算隐藏变量的期望值(假定当前假设是正确的),再重新计算极大似然假设(假定隐藏变量等于第1步中得到的期望值)。这一过程收敛到一个局部的极大似然假设以及隐藏变量的估计值。

在概率和统计方面有许多很好的介绍性文章,如 Casella & Berger: (1990)。几本快速参考类书籍(如 Maisel 1971, Speigel 1991)也对机器学习相关的概率和统计理论提供了很好的阐述。

对贝叶斯分类器和最小平方误差分类器的基本介绍由 Duda & Hart (1973) 给出, Domigos & Pazzani(1996) 分析了在什么条件下朴素贝叶斯方法可输出最优的分类,即使其独立性假定不成立时(关键在于在什么条件下即使相关联的后验概率估计不正确也可输出最优分类)。

Cmnik (1990) 讨论了使用 m-估计来估计概率。

将不同贝叶斯方法与决策树等其他算法进行比较的实验结果可在 Michie et al.(1994) 中找到。 Chauvin & Rumelhart (1995) 提供了基于反向传播算法的神经网络的贝叶斯分析。

对最小描述长度准则的讨论可参考 Rissanen(1983, 1989)。Quinlan & Rivest(1989) 描述了其使用以避免决策树的过度拟合。

统计推断内容在很多书中都有涉及,初等的参考书包括 (DeGroot and Schervish, 2000; Larsen and Marx, 1986), 中水平的参考书推荐读者参考 (Casella and Berger, 2002; Bickel and Doksum, 2000; Rice, 1995), 高级教程包括 (Cox and Hinkley, 2000; Lehmann and Casella, 1998; Lehmann, 1986; van der Vaart, 1998)。

Bootstrap 方法是 Efron(1979) 发明的。到目前为止,已经有一些书是关于这个论题的,包括 (Efron and Tibshirani, 1993; Davision and Hinkley, 1997; Hall, 1992; Shao and Tu, 1995)。同时,见 3.6 节的 (van der Vaart and Wellner, 1996)。

贝叶斯推断的参考书包括 (Carlin and Louis, 1996; Gelman et al., 1995; Lee, 1997; Robert, 1994; Schervish, 1995). 对于非参贝叶斯推断的技巧,见 (Cox, 1993; Diaconis and FVeedman, 1999; Barron et al., 1999; Ghosal et al., 2000; Shen and Wasserman, 2001; Zhao, 2000). Robins-Ritov 例子在 (Robins-Ritov, 1997) 中详细讨论,那里它更确切地被作为非参问题讨论. 例 11.10 来自 Edward George(个人通讯). 关于贝叶斯检验参考 (Berger and Delampady, 1987; Kass and Raftery, 1995). 对于无信息先验,见 (Kass and Wasserman, 1996).

决策理论的讨论可以见文献 (Casella and Berger, 2002; Berger, 1985; Ferguson, 1967; Lehmann and Casella, 1998).

有关线性回归的著作见文献 (Weisberg, 1985). 从数据挖掘角度写的有关回归的书见文献 (Hastie et al., 2002). Akaike 信息准则 (AIC) 见 Akaike(1973) 的著作. 贝叶斯信息准则 (BIC) 见文献 (Schwarz, 1978). Logistic 回归的参考文献 (Agresti, 1990) 和 (Dobson, 2001).

有很多关于 DAGs 的文献包括 Edwsrds (1995) 和 Jordan(2004). 第一个用 DAGs 来表示因果关系的是 Wright(1934) . 一些现代的论述包含在文献 (Spirtes et al., 2000) 和 (Pearl, 2000) 中. Robins 等 (2003) 讨论了从数据中来估计因果结构的问题。

最大似然估计和极大后验估计这两种方法都有很长的发展历史了。最初把贝叶斯方法引入

模式识别领域是 David, 它指出当类条件概率密度函数未知的情况下, 正确地使用训练样本的途径是计算 $P(\omega_i|x, \mathcal{D})$ 。贝叶斯自己也非常看重无信息先验的作用。一个详尽的对不同的先验概率的研究请参见 Harold Jeffreys 和 Dennis Victor Lindley。在 Jose M. Bernardo 中, 详细地列举了这方面的文献资料。Manfred Opper and David Haussler 中, 描述了 Gibbs 算法, 而 David Haussler, Michael Kearns, and Robert Schapire. Bounds 中, 对此进行了深入的分析。

主成分分析是一种经典的多元统计分析方法, 在广泛的工程领域中都得到了重要应用。Geoffrey J.McLachlan 详细而深入地描述了最初由 Fisher 所提出的线性可分性方法, 文献 Herman Chernoff,Pierre A, Devijver ,Keinosuke Fukunaga. 也进行了这方面的论述。

期望最大化算法是由 Dempster 等人提出的。Geoffrey J.McLachlan 对这一方法和其发展历史进行了详细论述。Michael I.Jordan ,D.Michael Tiitcrington 描述了期望最大化算法的在线版本。而专门讨论在丢失数据情况下的处理方法, 则可以参考 Donald B.Rubin, 当然, 这方面的进一步深入论述这超出了本书的范围。

马尔可夫在分析俄国文学家普希金的名著《叶夫盖尼? 奥涅金》的文字的过程中, 提出了后来被称为马尔可夫框架的思想。而 Baum 及其同事则提出/隐马尔可夫模型, 这一思想后来在语音识别领域 (awrence Rabiner) 得到了异常成功的应用。同时, 隐马尔可夫模型在“统计语言学习” (Eugene Charniak, Frederick Jelinek) 以及“序列符号识别” (比如 DNA 序列) (Pierre Baldi, Anders Krogh) 等领域也得到了应用。人们还把隐马尔可夫模型扩展到二维领域, 用于光学字符识别 (Gary E.Kopec)。而其中的解码算法则是由 Viterbi 和他的同事们 (G.David Forney,J Andrew J.Viterbi) 发展起来的。Padhraic Smyth 探讨了隐马尔可夫模型和图论模型 (比如贝叶斯置信网) 之间的联系。

Knuth 的经典著作是最初研究计算复杂度的著作, 他完成了这个领域的大部分工作。而该领域的标准教科书 (Thomas H.Cormen) 对于在计算机领域没有非常强的背景的读者是一本更好的入门性读物 (也为我们的几道课后习题提供了来源)。最后, Christopher M.Bishop, Padhraic Smyth 都是模式识别方面的很好的教材, 虽然采用了与本书有所不同的方式, 但也都值得推荐。

习题

习题 9.1. 随机地取 8 只活塞环, 测得它们的直径为 (以 mm 计)

74.001	74.005	74.003	74.001
74,000	73.998	74.006	74.002

试求总体均值 μ 及方差 σ^2 的矩估计值, 并求样本方差 s^2 .

习题 9.2. 设某种电子器件的寿命 (以 h 计) T 服从双参数的指数分布, 其概率密度为

$$f(t) = \begin{cases} \frac{1}{\theta} e^{-(t-c)/\theta} & t \geq c \\ 0 & \text{其他} \end{cases}$$

其中 $c, \theta (c, \theta > 0)$ 为未知参数. 自一批这种器件中随机地取 n 件进行寿命试验. 设它们的失效时间依次为 $x_1 \leq x_2 \leq \dots \leq x_n$.

(1) 求 θ 与 c 的最大似然估计值.

(2) 求 θ 与 c 的矩估计量

习题 9.3. 设 X_1, X_2, \dots, X_n 是来自概率密度为

$$f(x; \theta) = \begin{cases} \theta x^{\theta-1} & 0 < x < 1 \\ 0 & \text{其他} \end{cases}$$

的总体的样本, θ 未知, 求 $U = e^{-1/\theta}$ 的最大似然估计值.

习题 9.4. 设 x_1, x_2, \dots, x_n 是来自总体 $b(m, \theta)$ 的样本值, 又 $\theta = \frac{1}{3}(1 + \beta)$, 求 β 的最大似然估计值.

习题 9.5. 设总体 X 的概率密度为

$$f(x; \theta) = \begin{cases} \frac{1}{\theta} x^{(1-\theta)/\theta} & 0 < x < 1 \\ 0 & \text{其他} \end{cases} \quad 0 < \theta < +\infty$$

X_1, X_2, \dots, X_n 是来自总体 X 的样本.

(1) 验证 θ 的最大似然估计量是 $\hat{\theta} = \frac{-1}{n} \sum_{i=1}^n \ln X_i$

(2) 证明 $\hat{\theta}$ 是 θ 的无偏估计量.

习题 9.6. (1) 设 $\hat{\theta}$ 是参数 θ 的无偏估计, 且有 $D(\hat{\theta}) > 0$, 试证

$\hat{\theta}^2 = (\hat{\theta})^2$ 不是 θ^2 的无偏估计.

(2) 试证明均匀分布

$$f(x) = \begin{cases} \frac{1}{\theta} & 0 < x \leq \theta \\ 0 & \text{其他} \end{cases}$$

中未知参数 θ 的最大似然估计量不是无偏的.

习题 9.7. 设齐次马氏链的一步转移概率矩阵为

$$P = \begin{bmatrix} q & p & 0 \\ q & 0 & p \\ 0 & q & p \end{bmatrix}, \quad q = 1 - p, 0 < p < 1$$

试证明此链具有遍历性, 并求其平稳分布.

习题 9.8. 设马氏链的一步转移概率矩阵为

$$P = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

试证此链不是遍历的.

习题 9.9. 考虑高斯随机变量 $x \sim \mathcal{N}(x|\mu_x, \Sigma_x)$, 其中 $x \in \mathbb{R}^D$ 。进一步, 我们有

$$\mathbf{y} = \mathbf{Ax} + \mathbf{b} + \mathbf{w}$$

其中 $y \in \mathbb{R}^E$, $\mathbf{A} \in \mathbb{R}^{E \times D}$, $\mathbf{b} \in \mathbb{R}^E$, 并且 $\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mathbf{0}, \mathbf{Q})$ 是独立高斯噪声。

(1) 写出似然函数 $p(\mathbf{y}|x)$

(2) 证明 $p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$ 是高斯分布。计算 μ_y 和协方差 Σ_y

(3) 对随机变量 y 做变换

$$z = Cy + v$$

写出 $p(z|\mathbf{y})$, 计算 $p(z)$, 即均值 μ_z 和协方差 Σ_z

(4) 计算后验概率分布 $p(x|\hat{\mathbf{y}})$

参考文献

[1] AGRESTI,A.(1990).Categorical Data Analysis.Wiley.

[2] AKAIKE, H.(1973).Information theory and an extension of the maximum likelihood principle.Second International Symposium on Information Theory 267-281.

[3] BARRON,A., SCHERVISH, M.J.and WASSERMAN, L.(1999).The consistency of posterior distributions in nonparametric problems.The Annals of Statistics 27 536-561.

[4] BERGER, J.O.(1985).Statistical Decision Theory and Bayesian Analysis (Second Edition).Springer-Verlag.

[5] BERGER, J.O.and DELAMPADY, M.(1987).Testing precise hypotheses (P335-352).Statistical Science 2 317-335.

[6] CARLIN, B.P.and LOUIS, T.A.(1996).Bayes and Empirical Bayes Methods or Data Analysis.Chapman Hall.

[7] COX, D.D.(1993).An analysis of Bayesian inference for nonparametric regression.The Annals of Statistics 21 903-923.

[8] DIAGONIS, P.and FREEDMAN, D.(1986).On inconsistent Bayes estimates of location.The Annals of Statistics 14 68-87.

[9] EDWARDS, D.(1995).Introduction to graphical modelling.Springer-verlag.

[10] FERGUSON, T.(1967).Mathematical Statistics: a Decision Theoretic Approach.Academic

[11] GELMAN, A., CARLIN, J.B., STERN, H.S.and RUBIN, D.B,(1995).Bayesian Data Analysis, Chapman & Hall.

[12] GHOSAL, SM GHOSH, J.K.and VAN DER VAART, A.W.(2000).Convergence rates of posterior distributions.The Annals of Statistics 28 500-531.

[13] JORDAN, M.(2004).Graphical models.In Preparation.

- [14] KASS, R.E.and WASSERMAN, L.(1996).The selection of prior distributions by formal rules (corn 1998 v93 P 412).Journal of American Stati8tical Association 01 1343-1370.
- [15] LEE, P.M.(1997).Bayesian Statistics: An Introduction.Eldward Arnold.
- [16] LEHMANN, E.L.and CASELLA, G.(1998).Theory of Point Estimation.Springer-Verlag.
- [17] PEARL, J.(2000).Casualityi modela, reasoning, and inference.Cambridge University Press.
- [18] ROBINS, J.t SCHEINES, R., SPIRTES, P.and WASSERMAN, L.(2003).Uniform convergence in causal inference.Biometrika to appear.
- [19] LARSEN, R.J.and MARX, M.L.(1986).An Introduction to Mathematical Statistics and Its Applicationa(Second Edition).Prentice Hall.
- [20] DEGROOT, M.and Schervish, M.(2002).Probability and Statistics (Third Edition).AddisonWesley.
- [21] CASELLA, G.and BERGER, R.L.(2002).Statistical Inference.Duxbury Press.
- [22] BICKEL, P.J.and DOKSUM, K.A.(2000).Mathematical Statisticst Basic Ideas and Selected Topics, Vol.I(Second Edition).Prentice Hall.
- [23] RICE, J.A.(1995).Mathematical Statistics and Data Analysis (Second Edition).Duxbury Press.
- [24] COX, D.R.and HINKELEY, D.V.(2000).Theoretical statistics.Chapman& Hall.
- [25] LEHMANN, E.L.(1986).Testing Statistical Hypotheses Second Edition, Wiley.
- [26] VAN DER VAART, A.W.(1998).Asymptotic Statistics.Cambridge University Press.
- [27] EFRON, B.(1979).Bootstrap methods* Another look at the jackknife.The Annals of Statistics 71-26.
- [28] EFRON, B., TISBSHIRANI, R.J.(1993).An Introduction to the Bootstrap.Chapman & Hall.
- [29] DAVISON, A.C.and Hinkley, D.V.(1997).Bootstrap Methods and Their Application.Cambridge University Press.
- [30] HALL, P.(1992).The Bootstrap and Edgeworth Expansion, Springer-Verlag.
- [31] Russell G.Almond.Graphicat Belief Modelling.Chapman & Hall.New York,1995.
- [32] Pierre Baldi,Sorcн Brunak,Yves Chauvin,Jacob Engelbrecht, and Aadcrs Krogh.Hidden Markov models for human genes.In Stephen J.Hanson, Jack D.Cowan, and C.Lee Giles, editors.Advances in Neural Information Processing Systems, volume 6, pages 761-768, Morgan Kaufmann, San Mateo, CA 1994.
- [33] Leonard E.Baum and Ted Petrie.Statistical inference for probabilistic functions of finite state Markov chains Annals of Mathematical Statistics, 37:1554—1563, 1966.
- [34] Leonard E.Baum, Ted Petrie, George Soules, and Norman A maximizaioit technique occurring in the statistical analysis of probabilistic functions of Markov chains.Annals of Mathematical Statistics, 41(1):164-171,1970.
- [35] Jose M. Bernardo and Adrian F.M.Smith.Bayesian Theory.Wiley, New York, 1996.

- [36] Christopher M.Bishop.Neural Networks for Pattern Recognition.Oxford University Press, Oxford, UK.1995.
- [37] David Bravemian.Learning filters for optimum pattern recognition.IRE Transactions on Information Theory, IT8:280-285, 1962.
- [38] Eugene Charniak.Statistical Language Leuming.MIT Press, Cambridge, MA, 1993.
- [39] Herman Chernoff and Lincoln E.Moses.Elementary Decision Theory, Wiley, New York, 1959.
- [40] Thomas H.Cormcn, Charles E.Lciscron, and Ronald L.Rivest.Introduction to Algorithms.MIT Press, Cambridge, MA,1990.
- [41] Arthur P.Dempster, Nan M.Laird, and Donald B.Rubin.Maximum-likelihood from incomplete data via the EM algorithm (wirh discussion).Joumol of the Royal Statistical Society.Series Bt 39:1-38, 1977.
- [42] Pierre A, Devijver aad Josef Kittler.Patum Recog nition: A Statistical Approach, Prentice-Hall, London,1982.
- [43] Ronald A.Fisher.The use of itniltipte mcasuremcrus in taxonomic problems.Annals of Eugenics,7 Part II: 179-188, 1936.
- [44] G.David Forney, Jr.The Viterbi aigorithm.Proceedings of the IEEE, 61:268-278, 1973.
- [45] Keinosuke Fukunaga.Introduction to Stanstical Pattern Recognition.Academic Press, New Yor,second edition, 1990.
- [46] David Haussler, Michael Kearns, and Robert Schapire.Bounds on the sample complexity of Bayesian learning using informalion theory and the VC dimension.Machine Learnings 14:84-114, 1994.
- [47] Harold Jeffreys.Theory of Probability.Oxford University Press.Oxford, UKT 1961 reprint edition, 1939.
- [48] Frederick Jclinek.Statistical Methods for Speech Recognition MIT Press, Cambridge, MA, 1997.
- [49] Ian T.Jolliffc.Principal Component Analysis.SpringerVerlag, New York, 1986.
- [50] Michael I.Jordan and Robert A.Jacobs.Hierarchical mixtures of experts and the EM algorithm.Neural Computation, 6(2):181-214,1994.
- [51] Donald E.Knuih.The Art of Computer Programming volume 1.Addison-Wesley, Reading, MA, first edition,1973.
- [52] Gary E.Kopec and Phil A.Chou.Document image decoding using Markov source models.IEEE Transactions on Pattern Analysis and Machine Intelligence,16(6):602-617, 1994.
- [53] Anders Krogh, Michael Brown, I.Saira Mian, Kimmen Sjolander, and David Haussler.Hidden Markov models in computational biology: Applications to protein modelling.Journal of Molecular Biology, 235:1501-1531, 1994.

[54] Dennis Victor Lindley.The use of prior probability distributions in statistical inference and decision.In Jerzy Neyman and Elizabeth L.Scou, editors.Proceedings Fourth Berkeley Symposium on Mathematical Statistics and Probability,pages 453-468, University of California Press, Berkeley, CA, 1961.

草稿请勿外传

草稿请勿
使用

第十章 优化基础

根据第 1 章提及的统计学习理论中的经验风险最小化准则，我们知道数据科学、人工智能和机器学习的很多问题都归结为一个优化问题。对优化问题的求解已然成为大部分数据分析和机器学习算法的核心组成部分。而且机器学习算法都是在计算机上操作的，其数学公式就表示为数值优化算法。因为来源于实际应用的优化问题是如此的多样和复杂，所以在我们介绍各种具体的数值优化算法之前，我们将安排两章内容，也即本章和下一章，来理清我们所面对的各种优化问题以及其可解的条件，然后在第 12 章，我们会详细介绍各种具体的数值优化求解算法。本章主要介绍优化的基础理论。我们在第 5 章已经看到，普通的最小二乘问题可以用标准线性代数工具求解。在这种情况下，最小化问题的解可以被有效找到并且是整体最优解，也即，除了最小二乘最优解外没有其他更优的解。这些令人满意的特性实际上可扩展到一类更广泛的优化问题，而实现优化求解的关键特性就是所谓的“凸性”性质。因此在本章中，将主要介绍：优化问题的定义、优化问题的分类、数据科学中常见的优化问题、凸集和凸函数的定义和判别方法以及保凸运算、凸优化问题的定义和标准形式、具有特定结构的某些类型的凸优化模型，例如线性、凸二次或凸二次曲线模型等等。并介绍数据科学中常见的典型凸优化问题。

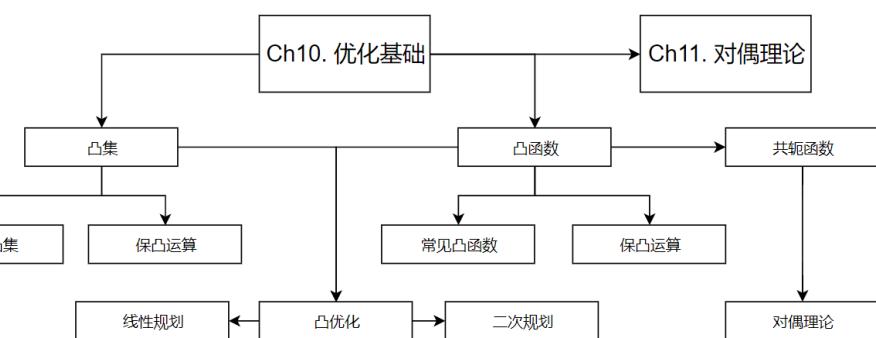


图 10.1: 本章导图

10.1 优化简介

在标准算法理论中，设计一个有效的算法来解决手头的问题是算法设计者的主要责任。自计算机科学引入的几十年以来，人们为各种任务设计了很多优美的算法，这些任务包括查找图中的最短路径、计算网络中的最佳流、压缩包含由数码相机拍摄的图像的计算机文件以及替换文本文档中的字符串等。

这些设计方法虽然对许多任务都很有用，但并没有解决更复杂的问题，例如在位图格式的图像中识别特定的人，或者将文本从英语翻译成中文。对于上述任务，可能有一个很好的算法，但是算法设计方案可能是不容易扩展的。

正如图灵在他的论文中所提倡的那样，我们要教计算机学习如何解决一个任务，而不是教给它特定任务的解决方案。实际上，这就是我们在学校中所做的，教会大家如何学习。我们希望教会计算机如何学习。这就是人工智能的思想，其核心是机器学习，并且主要就是从数据中来进行学习。

解决问题的机器学习方法有一个自动学习算法的机制。比如，我们考虑一个图像数据，将图像分为两类的问题：包含汽车的图像和包含椅子的图像（假设世界上只有两种类型的图像）。在机器学习中，我们训练（教导）一台机器以实现所需的功能，同一台机器可以潜在地解决任何算法任务，并且不同于一个任务到另一个任务只能由一组参数来决定机器的功能。这很像计算机芯片中的电线决定了它的功能。事实上，目前最流行的机器学习方法之一是人工神经网络。

机器学习的数学优化方法是将机器训练过程看作一个优化问题。如果我们把 $\theta \in \mathbb{R}^d$ 作为机器的参数（也即模型，确定了参数就确定了模型），它被限制在某个集合 $\mathcal{K} \subseteq \mathbb{R}^d$ 中，如果函数 f 成功地度量了将实例映射到它们的正确标签，那么这个训练过程可以用数学优化问题来描述：

$$\min_{\theta \in \mathcal{K}} f(\theta) \quad (10.1)$$

这是本书关注的主要问题，并且将特别强调机器学习中出现的具有特殊结构的函数，以便设计有效的算法。

事实上，根据度量的准则不同和参数模型的不一样，机器学习中会有很多各种具有特殊结构的优化问题。例如，我们在第 1 章中提到，在确定了训练集 \mathcal{D} 、假设空间 \mathcal{F} 以及学习准则后，如何找到最优的模型 $f(x, \theta)$ 就成了一个最优化（Optimization）问题。注意这里 $f(x, \theta)$ 类似于优化问题(1.1)中的 θ ，其中 x 是输入实例，它对应的输出记为 y ，它们一起形成训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}.$$

根据模型是否含有概率以及学习准则的不同，我们有如下四大类优化问题有待求解。

首先是经验风险最小化问题，求最优模型就是求解最优化问题：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)), \quad (10.2)$$

其中， \mathcal{F} 是假设空间， L 是损失函数，如平方损失函数等。

有时，为了避免过拟合，需引入结构风险最小化。结构风险最小化的策略认为结构风险最小的模型是最优的模型，也就是要求解最优化问题：

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f), \quad (10.3)$$

其中 $J(f)$ 为模型的复杂度，是定义在假设空间 \mathcal{F} 上的泛函。

上面两类优化主要针对函数类模型，当我们使用概率分布来为实际问题建模，我们会求解最大似然估计，也即最小化如下负对数似然问题：

$$\min_{\theta} \mathcal{L}(\theta), \quad (10.4)$$

其中 $\mathcal{L}(\theta) = -\log p(\mathbf{y}|\mathbf{X}, \theta) = -\sum_{n=1}^N \log p(y_n|\mathbf{x}_n, \theta)$ 。

如果我们有关于参数 θ 的分布的先验知识，则我们会求解一个最大后验估计，也即最小化如下负对数后验问题：

$$\min_{\theta} -\log p(\theta|\mathbf{x}), \quad (10.5)$$

其中 $p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} \propto p(\mathbf{x}|\theta)p(\theta)$ 。

机器学习的训练过程其实就是最优化问题的求解过程。在机器学习中，优化又可以分为参数优化和超参数优化。模型 $f(x; \theta)$ 中的 θ 称为模型的参数，可以通过优化算法进行学习，上面提及的四类优化问题都属于参数优化问题。除了可学习的参数 θ 之外，还有一类参数是用来定义模型结构或优化策略的，这类参数叫做超参数（Hyper-Parameter）。在贝叶斯方法中，超参数可以理解为参数的参数，即控制模型参数分布的参数。常见的超参数包括：聚类算法中的类别个数、梯度下降法的步长、正则项的系数、神经网络的层数、支持向量机中的核函数等。超参数的选取一般都是组合优化问题，很难通过优化算法来自动学习。因此，超参数优化是机器学习中一个经验性很强的技术，通常是按照人的经验设定，或者通过搜索的方法对一组超参数组合进行不断试错调整。

本书我们主要以参数优化为主。下面我们介绍两个具体的常见的机器学习优化问题的例子。

10.1.1 数据科学与机器学习中最优化问题的例子

线性分类与垃圾邮件处理

我们从第 1 章已经知道，监督学习中最基本的优化问题之一是用模型拟合数据或样本，也称为基于经验风险最小化的优化问题。

线性分类的监督学习范式就是这样的一个例子。在这个模型中，学习者面对的是一个概念的积极和消极的样本。每个样本用向量 \mathbf{a}_i 表示其在欧几里德空间中对应的 d 维特征向量。例如，垃圾邮件分类问题中电子邮件的常见表示是欧几里德空间中的二进制向量，其中空间的维数是语料中的单词数。第 i 封电子邮件是一个向量 \mathbf{a}_i ，其中邮件中出现过的单词在向量 \mathbf{a}_i 中对应的

位置为 1，否则为 0。此外，每个样本都有一个标签 $b_i \in \{-1, +1\}$ ，对应于电子邮件是否被标记为垃圾邮件/非垃圾邮件。

我们的目标是找到一个超平面来分离两类向量：带正标签的向量和带负标签的向量。如果不存在这样一个根据标签完全分离训练集的超平面，则目标是找到一个以最小错误数实现训练集分离的超平面。

从数学上讲，给定一组 m 个样本来训练，我们寻找 $\mathbf{x} \in \mathbb{R}^d$ ，它最小化了错误分类的样本的数量，即

$$\min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{m} \sum_{i \in [m]} \delta(\text{sign}(\mathbf{x}^T \mathbf{a}_i) \neq b_i)$$

其中， $\text{sign}(x) \in \{-1, +1\}$ 是符号函数，而 $\delta(z) \in \{0, 1\}$ 是指示函数，如果条件 z 满足，则取值 1，否则为 0。

上述线性分类的数学公式是数学优化问题(1.1)的特例，其中

$$f(x) = \frac{1}{m} \sum_{i \in [m]} \delta(\text{sign}(\mathbf{x}^T \mathbf{a}_i) \neq b_i) = \mathbf{E}_{i \sim [m]}[l_i(x)]$$

上式中为了简单，我们使用了期望算子，其中 $l_i(x) = \delta(\text{sign}(\mathbf{x}^T \mathbf{a}_i) \neq b_i)$ 。由于上面的优化问题是非凸的、非光滑的，所以通常采用凸松弛并用凸损失函数代替 $l_i(x)$ 。典型的选择包括均方误差函数和铰链损失函数

$$l_{a_i, b_i}(x) = \max\{0, 1 - b_i \cdot \mathbf{x}^T \mathbf{a}_i\}$$

在二分类的背景下，导致了著名的软间隔支持向量机问题。

另一个重要的优化问题是训练用于二分类的深层神经网络。例如，考虑一个图像数据集，它以位图格式表示，用 $\{\mathbf{a}_i \in \mathbb{R}^d | i \in [m]\}$ 表示，即 m 个图像除以 n 个像素。我们想找到一个从图像到汽车和椅子这两类 $\{b_i \in \{0, 1\}\}$ 的映射。映射由机器学习模型的一组参数给出，比如神经网络中的权重，或者支持向量机的值。因此，我们试图找出把 \mathbf{a}_i 匹配到 b_i 的最佳参数，也即求解如下数学优化问题

$$\min_{\mathbf{w} \in \mathbb{R}^d} f(w) = \mathbf{E}_{a_i, b_i}[l(f_w(a_i), b_i)].$$

矩阵补全和推荐系统

随着互联网的出现和在线媒体商店的兴起，媒体推荐已经发生了重大变化。收集到的大量数据能够有效地聚类和准确预测用户对各种媒体的偏好。一个众所周知的例子是所谓的“Netflix 挑战”——一个从用户的电影偏好的大数据集中进行推荐的自动化工具的竞赛。正如 Netflix 挑战中所证明的，自动化推荐系统最成功的方法之一是矩阵补全，它的最简单问题形式可以描述如下。

我们把整个用户-媒体偏好数据集看成是一个部分观测矩阵。矩阵中的每一行表示每个人，每一列表示一个媒体项（电影）。为了简单起见，让我们把观察结果看作是二元的，也即一个人

要么喜欢要么不喜欢某部电影。因此，我们有一个矩阵 $\mathbf{M} \in \{0, 1, *\}^{n \times m}$ ，其中 n 是考虑的总人数， m 是的电影数目，0/1 和 * 分别表示“不喜欢”、“喜欢”和“未知”：

$$M_{i,j} = \begin{cases} 0, & \text{第 } i \text{ 个人不喜欢第 } j \text{ 个电影} \\ 1, & \text{第 } i \text{ 个人喜欢第 } j \text{ 个电影} \\ *, & \text{偏好未知} \end{cases}$$

因为有很多用户和很多电影，这个矩阵通常非常大，只有部分位置有数值。一个自然的目标是补全矩阵，即正确地将 0 或 1 分配给未知项。到目前为止，这个问题是不适当的，因为任何补全都是一样好（或坏）的，而且对补全没有任何限制。

对补全的常见限制是“真”矩阵具有低秩。回想一下，如果矩阵 $\mathbf{X} \in \mathbb{R}^{n \times m}$ 的秩 $k \leq p = \min\{n, m\}$ 那么它可以写成

$$\mathbf{X} = \mathbf{U}\mathbf{V}, \quad \mathbf{U} \in \mathbb{R}^{n \times k}, \mathbf{V} \in \mathbb{R}^{k \times m}$$

这个性质的直观解释是 \mathbf{M} 中的每个条目只能用 k 个数字来解释。在矩阵补全中，这意味着，直觉上，只有 k 个因素决定一个人对电影的偏好，比如类型、导演、演员等等。

在这样的约束下，简单的矩阵补全问题可以很好地表述为在下面的数学优化。用 $\|\cdot\|_{OB}$ 表示仅在 \mathbf{M} 的观测（非星号）项上的欧几里得范数，即

$$\|\mathbf{X}\|_{OB}^2 = \sum_{M_{ij} \neq *} X_{ij}^2$$

$$s.t. \quad \text{rank}(\mathbf{X}) \leq k$$

则矩阵补全的数学优化问题可以描述如下：

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times m}} = \frac{1}{2} \|\mathbf{X} - \mathbf{M}\|_{OB}^2$$

$$s.t. \quad \text{rank}(\mathbf{X}) \leq k$$

10.1.2 其他常见的优化问题举例

词向量相关的优化问题

- 词向量：

$$\arg \max_{\mathbf{w}, b} \prod_{i=1}^m (h_{\mathbf{w}, b}(x_i)^{y_i} * (1 - h_{\mathbf{w}, b}(x_i)^{1-y_i}))$$

$$\arg \min_{\mathbf{w}, b} - \sum_{i=1}^m (y_i \log h_{\mathbf{w}, b}(x_i)) + (1 - y_i) \log(1 - h_{\mathbf{w}, b}(x_i))$$

- 连续词袋模型

$$\min_{u, v} -u_c^T \hat{v} + \log \sum_{j=1}^{|V|} \exp(u_j^T \hat{v})$$

外传
阅读
或
草稿

- 跳格模型

$$\min - \sum_{j=0, j \neq m}^{2m} u_{c-m+j}^T v_c + 2m \log \sum_{k=1}^{|V|} \exp(u_k^T v_c)$$

推荐系统中的优化问题

- 我们讨论了在推荐系统中的优化问题

$$\begin{aligned} & \min_X \quad \text{rank}(X) \\ & s.t. \quad X_{ij} = M_{ij} \quad \forall i, j \in \mathbb{E} \end{aligned}$$

或者转化为限定在秩为 r 的条件下, 求矩阵使得观测到的评分与预测的评分最接近:

$$\begin{aligned} & \min_X \quad \sum_{ij} (X_{ij} - M_{ij})^2 \quad \forall i, j \in \mathbb{E} \\ & s.t. \quad \text{rank}(X) = r \end{aligned}$$

与正交矩阵相关的优化问题

- 正交 Procrustes 问题

$$\arg \max_{Q \in O(n)} \text{Tr}(SQ)$$

- Wahba 问题

$$\arg \min_{R \in SO(n)} \sum_{t=1}^T \frac{1}{2} \|Rx_t - y_t\|^2 = \arg \max_{R \in SO(n)} \text{Tr}(SR)$$

低秩矩阵相关优化问题

- 鲁棒 PCA

$$\min_{A,E} \|A\|_* + \lambda \|E\|_1, s.t. X = A + E$$

- 低秩矩阵补全

$$\min_A \|A\|_* s.t. P_\Omega(A) = P_\Omega(D)$$

- 低秩矩阵表示

$$\min_Z \|Z\|_* s.t. D = BZ$$

以及

$$\min_{Z,E} \|Z\|_* + \lambda \|E\|_{2,1} s.t. D = DZ + E$$

最小二乘问题相关优化问题

- 最小二乘问题

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2 \tag{10.6}$$

- 加权最小二乘

$$\min \| \mathbf{A}_w \mathbf{x} - \mathbf{y}_w \|_2^2$$

- 约束最小二乘

$$\begin{aligned} & \min_{\mathbf{x}} \frac{1}{2} \| \mathbf{A} \mathbf{x} - \mathbf{b} \|_2^2 \\ & s.t. \quad \mathbf{B} \mathbf{x} = \mathbf{f} \end{aligned}$$

- 总体最小二乘

$$\begin{aligned} \text{TLS: } & \min_{\Delta \mathbf{A}, \Delta \mathbf{b}, \mathbf{x}} \| \Delta \mathbf{A} \|_{\text{F}}^2 + \| \Delta \mathbf{b} \|_2^2 \\ & \text{subject to} \quad (\mathbf{A} + \Delta \mathbf{A}) \mathbf{x} = \mathbf{b} + \Delta \mathbf{b} \end{aligned}$$

机器学习中的优化问题

- 逻辑回归:

$$\min_{\mathbf{w}} \sum_{i=1}^N [y_i(\mathbf{w}^T \mathbf{x}_i) - \log(1 + \exp(\mathbf{w}^T \mathbf{x}_i))]$$

- 线性回归模型

$$\min_{(\mathbf{w}, b)} \sum_{i=1}^m (\mathbf{w}^T \mathbf{x}_i + b - y_i)^2$$

- 感知机

$$\min_{\mathbf{w}, b} - \sum_{x_i \in M} y_i(\mathbf{w}^T \mathbf{x}_i + b)$$

- 支持向量机

$$\min_{\mathbf{w}, b} \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^N L_{0/1}(y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1),$$

$$\min_{\mathbf{w}, b} \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^N \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$$

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^N \xi_i$$

$$s.t. \quad y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, 2, \dots, N$$

$$\xi_i \geq 0, i = 1, 2, \dots, N$$

- 非线性支持向量机

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i$$

$$s.t. \quad \sum_{i=1}^N \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, \dots, N$$

草稿请勿外传

- PCA

$$\min_{\mathbf{W}} \text{Tr}(-\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W})$$

$$s.t. \mathbf{W}^T \mathbf{W} = \mathbf{I}$$

- k 均值聚类

$$C^* = \arg \min_C W(C) = \arg \min_C \sum_{l=1}^k \sum_{C(i)=l} \|Vx_i - \bar{x}_l\|^2$$

- 谱聚类

$$\min_{\mathbf{x}} \mathbf{x}^T \mathbf{L} \mathbf{x}$$

$$s.t. \mathbf{x}^T \mathbf{1} = 0$$

10.1.3 优化问题的一般形式

下面我们给出数学优化问题的一般形式以及相关的概念。

优化问题的一般形式

最优化问题或者说优化问题的一般形式表示为:

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq b_i, i = 1, \dots, m \\ & && h_j(x) = 0, j = 1, \dots, p \end{aligned} \tag{10.7}$$

其中, 向量 $\mathbf{x} = (x_1, \dots, x_n)$ 称为问题的优化变量, 函数 $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ 称为目标函数, 函数 $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, 被称为不等式约束函数, 常数 b_1, \dots, b_m 称为约束上限或者约束边界, $f_i(x) \leq b_i (i = 1, \dots, m)$ 称为不等式约束, 函数 $h_j : \mathbb{R}^n \rightarrow \mathbb{R}$, 被称为等式约束函数, $h_j(x) = 0, j = 1, \dots, p$ 称为等式约束。

称满足所有约束条件的向量 \mathbf{x} 为可行解或可行点, 全体可行点的集合称为可行集, 记为 D , 其表示为:

$$D = \{\mathbf{x} | h_j(x) = 0, j = 1, \dots, p, f_i(x) \leq b_i, i = 1, \dots, m\}$$

若 $h_j(x)$ 和 $f_i(x)$ 是连续函数, 则 D 是闭集。

在可行集中找一点 x^* , 使目标函数 $f_0(x)$ 在该点取最小值, 即满足: $f_0(x^*) = \min f_0(x)$, 使得 $f_i(x^*) \leq b_i$ 和 $h_j(x^*) = 0$ 的过程即为最优化的求解过程。 x^* 称为问题的最优点或最优解, $f_0(x^*)$ 称为最优值。

优化问题(1.7)可以看成在向量空间 \mathbb{R}^n 的一集备选解中选择最好的解。用 x 表示备选解, $f_i(x) \leq b_i$ 和 $h_j(x) = 0$ 表示 x 必须满足的条件, 目标函数 $f_0(x)$ 表示选择 x 的成本(同理也可以认为 $-f_0(x)$ 表示选择 x 的效益或者效用)。优化问题(1.7)的解即为满足约束条件的所有备选解中成本最小(或者效用最大)的那个解。

在数据拟合中，人们需要在一族候选模型中选择最符合观测数据与先验知识的模型。此时，变量为模型中的参数，约束可以是先验知识以及参数限制（比如说非负性）。目标函数可能是与真实模型的偏差或者是观测数据与估计模型的预测值之间的偏差，也有可能是参数值的似然度和置信度的统计估计。优化问题(1.7)此时即为寻找合适的模型参数值，使之符合先验知识，且与真实模型之间的偏差或者预测值与观测值之间的偏差最小（或者在统计意义上更加相似）。

局部最优和整体最优

定义 10.1.1. 整体（全局）最优解：若 $x^* \in D$ ，对于一切 $x \in D$ ，恒有 $f_0(x^*) \leq f_0(x)$ ，则称 x^* 是最优化问题(1.7)的整体最优解。

定义 10.1.2. 局部最优解：若 $x^* \in D$ ，存在某个领域 $N_\varepsilon(x^*)$ ，使得对于一切 $x \in N_\varepsilon(x^*) \cap D$ ，恒有 $f_0(x^*) \leq f_0(x)$ ，则称 x^* 是最优化问题(1.7)的局部最优解。其中 $N_\varepsilon(x^*) = \{x | \|x - x^*\| < \varepsilon, \varepsilon > 0\}$

严格最优解：当 $x \neq x^*$ ，有 $f_0(x^*) < f_0(x)$ 则称 x^* 为优化问题(1.7)的严格最优解。

由上述定义可知，局部最优解 x^* 使 f_0 最小，但仅对可行集上的邻近点。此时目标函数的值不一定是问题的（全局）最优值。局部最优解可能对用户没有实际意义。因此局部最优解的存在在一般优化问题中是一个挑战，因为大多数算法往往被困在局部极小，如果存在的话，从而不能产生期望的全局最优解。

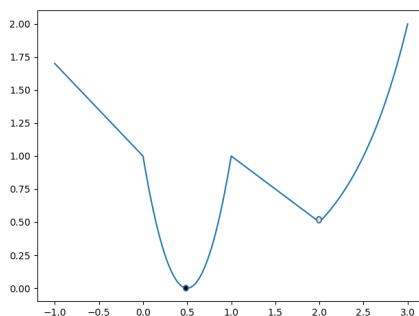


图 10.2：局部（灰色）与全局（黑色）最小值。最佳集是单重态 $X_{opt} = 0.5$ ，点 $x=2$ 是局部最小值。

易处理优化问题和不易处理优化问题

并非所有的优化问题都是平等的。一些问题类，如寻找一组有限的线性等式或不等式的解，可以用有效可靠的方法数值求解。相反，对于其他的一些问题，没有可靠有效的求解算法。

不讨论优化问题的计算复杂性，在这里，我们称之为“可处理的”所有那些优化模型，对于这些模型，可以用可靠的方式（在任何问题实例中）在数值上找到全局最优解，并且随着问

题的大小（非正式地，问题的大小由模型中的决策变量和/或约束的数量来衡量）。其他问题被称为“困难”，然而对于其他问题，计算复杂性是未知的。

这本书的重点是可处理的模型，一个关键的信息是，可以以线性代数问题的形式或以凸的形式来表达的模型，通常是可处理的。此外，如果凸模型具有一些特殊结构，那么，可以使用一些现有的可靠的数值求解器进行求解。

问题变换

优化问题(1.7)的形式是非常灵活的，并允许许多变换，这就有利于我们将一个给定的问题处理为一个易于处理的问题。例如，优化问题

$$\min_x \sqrt{(x_1 + 1)^2 + (x_2 - 2)^2} \quad \text{s.t. : } x_1 \geq 0$$

与

$$\min_x (x_1 + 1)^2 + (x_2 - 2)^2 \quad \text{s.t. : } x_1 \geq 0$$

是等价的，而第二个优化问题的目标函数是可微的。有些情况下，还可以使用变量替换。例如，给定一个优化问题

$$\max_x x_1 x_2^3 x_3 \quad \text{s.t. : } x_i \geq 0, i = 1, 2, 3, x_1 x_2 \leq 2, x_2^3 x_3 \leq 1$$

令新变量 $z_i = \log x_i, i = 1, 2, 3$ ，在取目标的对数之后，该问题可以等价地写为

$$\max_z z_1 + 3z_2 + z_3 \quad z_1 + z_2 \leq \log 2, 2z_2 + z_3 \leq 0.$$

优点是替换后的目标函数和约束函数都是线性的。

10.1.4 优化问题的分类

优化问题种类繁多，因而分类的方法也有许多。可以按变量的性质分类，按有无约束条件分类，按目标函数的个数分类等等。一般来说，变量可以分为确定性变量，随机变量和系统变量等等，相对应的最优化问题分别称为：普通最优化问题，统计最优化问题和系统最优化问题。

根据输入变量 x 的值域是否连续，数学优化问题可以分为离散优化问题和连续优化问题。

离散优化

离散优化 (Discrete Optimization) 问题是目标函数的输入变量为离散变量，比如为整数或有限集合中的元素。离散优化问题主要有三个分支：

1. 整数规划 (Integer Programming)：输入变量 $x \in \mathbb{Z}^d$ 为整数向量。常见的整数规划问题通常为整数线性规划 (Integer Linear Programming, ILP)。整数线性规划的一种最直接的求解方法是：(1) 去掉输入必须为整数的限制，将原问题转换为一般的线性规划问题，这个线性规划问题为原问题的松弛问题；(2) 求得相应松弛问题的解；(3) 把松弛问题的解四舍五入到最接近的整数。但是这种方法得到的解一般都不是最优的，因为原问题的最优解不一定在松弛问题最优解的附近。另外，这种方法得到的解也不一定满足约束条件。

2. 混合整数规划 (Mixed Integer Programming, MIP), 即自变量既包含整数也有连续变量。

3. 组合优化 (Combinatorial Optimization): 其目标是从一个有限集合中找出使得目标函数最优的元素。在一般的组合优化问题中, 集合中的元素之间存在一定的关联, 可以表示为图结构。典型的组合优化问题有旅行商问题、最小生成树问题、图着色问题等。很多机器学习问题都是组合优化问题, 比如特征选择、聚类问题、超参数优化问题以及结构化学习 (Structured Learning) 中标签预测问题等。

从这个意义上讲, 组合优化是整数规划的子集。的确, 绝大多数组合优化问题都可以被建模成 (混合) 整数规划模型来求解。

离散优化问题的求解一般都比较困难, 优化算法的复杂度都比较高。

连续优化

连续优化 (Continuous Optimization) 问题的目标函数的输入变量为连续变量 $\mathbf{x} \in \mathbb{R}^d$, 即目标函数为实函数。一般认为, 在深度学习或机器学习中, 模型中要学习的参数是连续变量。本书中主要讲解连续优化问题内容为主。

在连续优化问题中, 根据是否有变量的约束条件, 可以将优化问题分为无约束优化问题和约束优化问题。

1. 无约束优化问题 (Unconstrained Optimization) 的可行域为整个实数域 $D = \mathbb{R}^d$ 。在优化问题(1.7)中, 当我们把不等式约束 $f_i(x) \leq b_i$ 和等式约束 $h_j(x) = 0$ 去掉时, 即退化为无约束优化问题

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad (10.8)$$

其中 $\mathbf{x} \in \mathbb{R}^d$ 为输入变量, $f : \mathbb{R}^d \rightarrow \mathbb{R}$ 为目标函数。我们前面提到的最小二乘问题和低秩近似问题都属于无约束优化问题。

2. 约束优化问题 (Constrained Optimization) 中变量 \mathbf{x} 需要满足一些等式或不等式的约束。在优化问题(1.7)中, 当不等式约束 $f_i(x) \leq b_i$ 和等式约束 $h_j(x) = 0$ 只要有一个成立, 其即被称为约束优化问题。我们之前提到的最大方差问题属于约束优化问题。

此外, 在连续优化问题中, 根据函数的线性性质, 可以将优化问题分为线性规划 (线性优化) 和非线性规划 (非线性优化)。

1. 在优化问题(1.7)中, 当目标函数和所有的约束函数都为线性函数, 则该问题为线性规划问题 (Linear Programming)。

线性规划问题的解并没有一个简单的解析表达形式 (和最小二乘问题不同), 然而, 存在很多非常有效的求解线性规划问题的方法, 这当中包括 Dantzig 的单纯形法以及上世纪 80 年代发展起来的内点法。和最小二乘问题一样, 处理极大规模的线性规划问题或者在很短时间内实时解决线性规划问题还是具有一定难度的。但是和最小二乘问题的情况类似, 我们可以说求解 (大部分) 线性规划问题是一项成熟的技术。线性规划的求解程序可以 (并已经) 嵌入到很多工具箱和应用软件中。虽然一些应用可以直接表述为线性规划的形式, 但是在很多情况下, 原始的优化

问题并不是线性规划问题的标准形式，这可以利用技巧转化为一个等价的线性规划问题（然后进行求解）。

2. 在优化问题(1.7)中，如果目标函数或任何一个约束函数为非线性函数，则该问题为非线性规划问题（Nonlinear Programming）。

对于一般的非线性规划问题，目前还没有有效的求解方法。有时看似简单的问题，变量个数可能不到 10，却非常难以求解，更不用说上百变量的非线性优化问题。因此，现有的用于求解一般非线性规划的问题都是在放宽某些指标条件下，采取不同的途径进行求解。

更进一步，根据函数的凸性，也即是否为凸函数，我们还可以把优化问题分为凸优化（Convex Programming）和非凸优化。

1. 在凸优化问题中，变量 \mathbf{x} 的可行域为凸集，即对于集合中任意两点，它们的连线全部位于集合内部。目标函数 f 也必须为凸函数，即满足

$$f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y}), \forall \alpha \in [0, 1]$$

凸优化问题是一种特殊的约束优化问题，需满足目标函数为凸函数，并且等式约束函数为线性函数，不等式约束函数为凸函数。

最小二乘问题和线性规划问题实质上都是凸优化问题的特殊形式。在非线性规划问题中，有些是凸优化问题，有些是非凸优化问题。

凸优化问题的解并没有一个解析表达式，但是，和线性规划问题类似，存在很多有效的算法求解凸优化问题，如内点法，它可以在多项式时间内以给定精度求解这些凸优化问题。内点法几乎总可以在 10 步到 100 步之间解决凸优化问题。如果不考虑特殊结构的凸优化问题（如稀疏结构），每一步需要的操作次数和下述变量成正比： $\max\{n^3, n^2m, F\}$ ，其中 F 是计算目标函数和约束函数 f_0, \dots, f_m 的一阶导数和二阶导数所需要的计算量。我们可以使用目前的台式计算机轻易地解决包含数百变量、数千约束的凸优化问题，计算时间不超过一百秒。如果问题本身具有一些特殊结构（如稀疏结构），则可以解决含有数千变量以及约束的更大规模的问题。如果是一般的非线性凸优化问题，则并不容易，目前只对几类重要问题，如二阶锥规划和几何规划问题等，内点法正在逐渐成为一项成熟的技术。

类似于线性规划问题，理论上，对于大部分优化问题只要能把问题表述为凸优化问题，我们就能迅速有效地进行求解。然而，不像判断某个问题是否为最小二乘问题或线性规划问题非常直接，凸优化问题的识别比较困难。此外，较之线性规划问题，转换为凸优化问题的过程中存在更多的技巧。因此，判断某个问题是否属于凸优化问题或识别那些可以转换为凸优化问题的问题是具有挑战性的工作。

2. 非凸优化对应于标准形式(1.7)中的一个或多个目标函数或约束函数不具有凸性的问题。一般来说，这样的问题很难解决。实际上，这个类包含组合优化：如果一个变量 x_i 必须是布尔型的（即 $x_i \in \{0 \square 1\}$ ），我们可以将其建模为一对约束，其中第二个约束包含一个非凸函数。一般的非凸问题很难解决的原因之一是它们可能出现局部极小值然而，需要注意的是，并不是每

一个非凸优化问题都难以求解。例如，最大方差和低秩逼近问题是非凸问题，可以使用线性代数的特殊算法可靠地求解。

除了上述分类，还有按目标函数的个数分类：单目标最优化问题，多目标最优化问题；以及按约束条件和目标函数是否是时间的函数分类：静态最优化问题和动态最优化问题（动态规划）。本书主要考虑单目标静态最优化问题为主。

由上述讨论可知，目前对于线性规划和凸优化问题有成熟的求解方法，对于其它问题的求解思路主要想办法把它转化为线性规划或凸优化问题。因此接下来我们重点介绍凸优化问题有关的凸分析基础，包括凸集、凸函数以及凸优化问题的性质等等。

10.2 凸集

本节我们将首先给出凸集的定义并介绍一些相关的例子，然后给出保持凸集的一些基本运算，最后给出在机器学习中常用的分离超平面和支撑超平面。

10.2.1 仿射集合和凸集

在给出凸集的定义之前，我们首先回顾线段和仿射集合的定义。

直线与线段

定义 10.2.1. 设 $x_1 \neq x_2$ 为 \mathbb{R}^n 空间中的两个点，那么具体有下列形式的点

$$y = \theta x_1 + (1 - \theta) x_2, \theta \in \mathbb{R}$$

组成一条穿越 x_1 和 x_2 的直线。参数 $\theta = 0$ 对应 $y = x_2$ ，而 $\theta = 1$ 对应 $y = x_1$ 。参数 θ 的值在 0 和 1 之间变动，构成了 x_1 和 x_2 之间的(闭)线段。

仿射集合

定义 10.2.2. 如果通过集合 $C \subseteq \mathbb{R}^n$ 中任意两个不同点的直线上的点仍然在集合 C 中，那么，集合 C 被称为仿射集合。

也就是说， $C \subseteq \mathbb{R}^n$ 是仿射集合等价于：对于任意 $x_1, x_2 \in C$ 及 $\theta \in \mathbb{R}$ 有 $\theta x_1 + (1 - \theta) x_2 \in C$ 。换而言之， C 包含了 C 中任意两点的系数之和为 1 的线性组合。

这个概念可以扩展到多个点的情况。如果 $\theta_1 + \theta_2 + \dots + \theta_k = 1$ ，我们称具有 $\theta_1 x_1 + \dots + \theta_k x_k$ 形式的点为 x_1, \dots, x_k 的仿射组合。利用仿射集合的定义（即仿射集合包含其中任意两点的仿射组合），我们可以归纳出以下结论：一个仿射集合包含其中任意点的仿射组合，即如果 C 是一个仿射集合， $x_1, \dots, x_k \in C$ ，并且 $\theta_1 + \dots + \theta_k = 1$ ，那么 $\theta_1 x_1 + \dots + \theta_k x_k$ 仍然在 C 中。

例 10.2.1. 线性方程组的解集。线性方程组的解集 $C = \{x | Ax = b\}$ 是一个仿射集合，其中 $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ 。为说明这点，任取 $x_1, x_2 \in C$ ，则有 $Ax_1 = b$, $Ax_2 = b$ 。对于任意 θ ，我们有

$$A(\theta x_1 + (1 - \theta)x_2) = \theta Ax_1 + (1 - \theta)Ax_2 = \theta b + (1 - \theta)b = b$$

这表明，任意的仿射组合 $\theta x_1 + (1 - \theta)x_2$ 也在仿射集合 C 中。

我们称由集合 $C \subseteq \mathbb{R}^n$ 中的点的所有仿射组合组成的集合为 C 的仿射包，记为 $\text{aff } C$ ：

$$\text{aff } C = \{\theta_1 x_1 + \dots + \theta_k x_k | x_1, \dots, x_k \in C, \theta_1 + \dots + \theta_k = 1\}$$

仿射包是包含 C 的最小的仿射集合，也就是说：如果 S 是满足 $C \subseteq S$ 的仿射集合，那么 $\text{aff } C \subseteq S$ 。

仿射维数与相对内部

我们定义集合 C 的仿射维数为其仿射包的维数。仿射维数在凸分析及凸优化中十分有用，但它与其他维数的定义常常不相容。作为一个例子，考虑 \mathbb{R}^2 上的单位圆环 $\{x \in \mathbb{R}^2 | x_1^2 + x_2^2 = 1\}$ 。它的仿射包是全空间 \mathbb{R}^2 ，所以其仿射维数为 2。但是，在其他大多数维数的定义下， \mathbb{R}^2 上的单位圆环的维数为 1。

如果集合 $c \subseteq \mathbb{R}^n$ 的仿射维数小于 n ，那么这个集合在仿射集合 $\text{aff } C \neq \mathbb{R}^n$ 中。我们定义集合 C 的相对内部为 $\text{aff } C$ 的内部，记为 $\text{relint } C$ ，即

$$\text{relint } C = \{x \in C | B(x, r) \cap \text{aff } C \subseteq C \text{ 对于某些 } r > 0\}$$

其中 $B(x, r) = \{y | \|y - x\| \leq r\}$ ，即半径为 r ，中心为 x 并由范数 $\|\cdot\|$ 定义的球（这里的 $\|\cdot\|$ 可以是任意范数，并且所有范数定义了相同的相对内部）。我们于是可以定义集合 C 的相对边界为 $\text{cl } C \setminus \text{relint } C$ ，此处 $\text{cl } C$ 表示 C 的闭包。

例 10.2.2. 考虑 \mathbb{R}^3 处于 (x_1, x_2) 平面的一个正方形，定义

$$C = \{x \in \mathbb{R}^3 | -1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1, x_3 = 0\}$$

其仿射包为 (x_1, x_2) -平面，即 $\text{aff } C = \{x \in \mathbb{R}^3 | x_3 = 0\}$ 。 C 的内部为空，但其相对内部为

$$\text{relint } C = \{x \in \mathbb{R}^3 | -1 < x_1 < 1, -1 < x_2 < 1, x_3 = 0\}$$

C (在 \mathbb{R}^3 中) 的边界是其自身，而相对边界是其边框，

$$\{x \in \mathbb{R}^3 | \max\{|x_1|, |x_2|\} = 1, x_3 = 0\}$$

下面我们给出凸集的定义。

凸集

定义 10.2.3. 如果集合 $C \subseteq \mathbb{R}^n$ 中任意两点间的线段仍在 C 中，即对于任意 $x_1, x_2 \in C$ 和满足 $0 \leq \theta \leq 1$ 的 θ ，都有

$$\theta x_1 + (1 - \theta)x_2 \in C$$

那么，集合 C 是凸集。

换句话说，凸集中的每一点都可以被其他点沿着它们之间一条无阻碍的路径看见，所谓无障碍，是指整条路径都在集合中。由于仿射集包含穿过集合中任意不同两点的整条直线，任意不同两点间的线段自然也在集合中。因而仿射集是凸集。下图显示了 \mathbb{R}^2 空间中一些简单的凸和非凸集合。

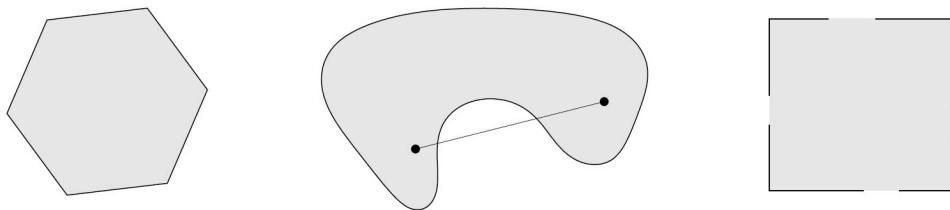


图 10.3: 一些简单的凸和非凸集合。(左) 包含其边界的六边形是凸的；(中) 肾形集合不是凸的，因为图中所示集合中两点间的线段不为集合所包含；(右) 仅包含部分边界的正方形不是凸的。

我们称点 $\theta_1x_1 + \dots + \theta_kx_k$ 为点 x_1, \dots, x_k 的一个凸组合，其中 $\theta_1 + \dots + \theta_k = 1$ ，并且 $\theta_i \geq 0, i = 1, \dots, k$ 。与仿射集合类似，一个集合是凸集等价于该集合包含其中所有点的凸组合。点的凸组合可以看做是这些点的混合或加权平均， θ_i 代表混合时 x_i 所占的份数。

我们称集合 C 中所有点的凸组合的集合为其凸包，记为 $\text{conv}C$:

$$\text{conv}C = \{\theta_1x_1 + \dots + \theta_kx_k | x_i \in C, \theta_i \geq 0, i = 1, \dots, k, \theta_1 + \dots + \theta_k = 1\}$$

顾名思义，凸包总是凸的。它是包含 C 的最小的凸集。也就是说，如果 \mathbb{B} 是包含 C 的凸集，那么 $\text{conv}C \subseteq \mathbb{B}$ 。下图显示了凸包的定义。

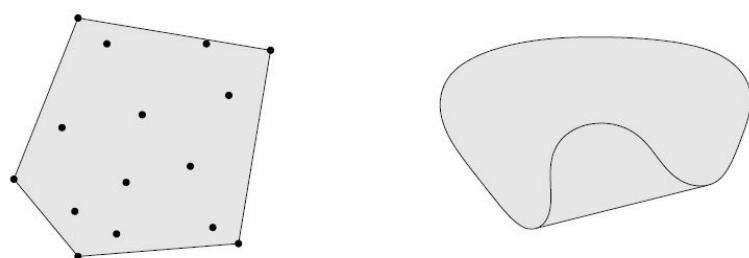


图 10.4: \mathbb{R}^2 上两个集合的凸包。(左) 十五个点的集合的凸包是一个五边形(阴影所示);(右) 肾形集合的凸包是阴影所示的集合

锥

在集合中，有一类特殊的集合，称为锥，我们可以把凸集的定义推广到锥集合上，形成凸锥。

定义 10.2.4. 如果对于任意 $x \in C$ 和 $\theta \geq 0$ ，都有 $\theta x \in C$ ，我们称集合 C 是锥或者非负齐次。如果集合 C 是锥，并且是凸的，则称 C 是凸锥，即对于任意 $x_1, x_2 \in C$ 和 $\theta_1, \theta_2 \geq 0$ ，都有

$$\theta_1 x_1 + \theta_2 x_2 \in C$$

在几何上，具有此类形式的点构成了二维的扇形，这个扇形以 0 为定点，边通过 x_1 和 x_2 ，如图 1.5 所示：

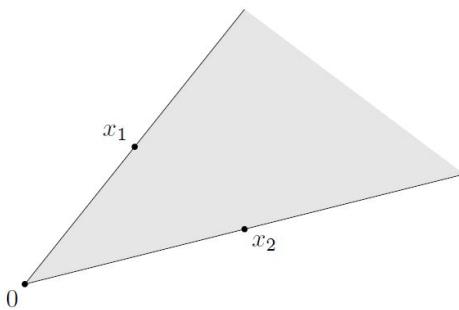


图 1.5：扇形显示了所有具有形式 $\theta_1 x_1 + \theta_2 x_2$ 的点，其中 $\theta_1, \theta_2 \geq 0$ 。扇形的顶点 ($\theta_1 = \theta_2 = 0$) 在 O 处，其边界 ($\theta_1 = 0$ 或 $\theta_2 = 0$) 穿过点 x_1 或 x_2

具有 $\theta_1 x_1 + \cdots + \theta_k x_k, \theta_1, \dots, \theta_k \geq 0$ 形式的点称为 x_1, \dots, x_k 的锥组合(或非负线性组合)。如果 x_i 均属于凸锥 C ，那么， x_i 的每一个锥组合也在 C 中。反言之，集合 C 是凸锥的充要条件是它包含其元素的所有钱组合。

我们称集合 C 中所有元素的锥组合的集合为其锥包，即：

$$\{\theta_1 x_1 + \cdots + \theta_k x_k | x_i \in C, \theta_i \geq 0, i = 1\},$$

它是包含 C 的最小的凸锥。

10.2.2 重要的凸集例子

本节将描述一些重要的凸集，这些凸集在本书的后续部分将多次遇见。

简单的例子

- 空集、任意一个点（即单点集） $\{x_0\}$ 、空间 \mathbb{R}^n 都是凸集。

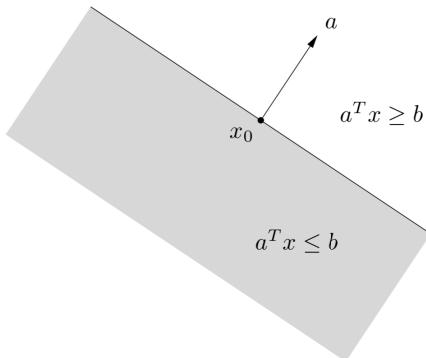


图 10.6: \mathbb{R}^2 上的 $a^T x = b$ 定义的超平面决定了两个半空间。由 $a^T x \geq b$ 决定的半空间（无阴影）是向 a 拓展的半空间。由 $a^T x \leq b$ 决定的半空间（阴影所示）是向 $-a$ 拓展的半空间。向量 a 是这个半空间向外的法向量。

- 任意一条直线是仿射集合，也是凸集。如果直线通过零点，则是子空间，因此，也是凸锥
- 任意一条线段是凸集，但不是仿射集合（除非退化为一个点）。
- 任意一条射线，即具有形式 $\{x_0 + \theta v | \theta \geq 0\}, v \neq 0$ 的集合，是凸集，但不是仿射集合。如果射线的基点 x_0 是 0，则它是凸锥。
- 任意子空间是仿射集合，也是凸锥。

超平面与半空间

超平面是具有下面形式的集合

$$\{x | a^T x = b\}$$

其中 $a \in \mathbb{R}^n, a \neq 0$ 且 $b \in \mathbb{R}$ 。解析地，超平面是关于 x 的非平凡线性方程的解空间（因此是一个仿射集合）。几何上，超平面 $\{x | a^T x = b\}$ 可以解释为与给定向量 a 的内积为常数的点的集合；也可以看成法线方向为 a 的超平面，而常数 $b \in \mathbb{R}$ 决定了这个平面离原点的偏移。

一个超平面将 \mathbb{R}^n 划分为两个半空间，分别是 $\{x | a^T x \leq b\}$ 和 $\{x | a^T x \geq b\}$ 。半空间是凸集，但不是仿射集合，如图 1.6 所示。

半空间 $\{x | a^T x \leq b\}$ 也可以表示为，

$$\{x | a^T(x - x_0) \leq 0\}$$

其中 x_0 是超平面上 $a^T x = b$ 的任意一点，即满足 $a^T x_0 = b$ 。半空间 $\{x | a^T(x - x_0) \leq 0\}$ 的边界是超平面 $\{x | a^T x = b\}$ 。集合 $\{x | a^T x < b\}$ 是半空间 $\{x | a^T x \leq b\}$ 的内部，称为开半空间。

Euclid 球与范数球

\mathbb{R}^n 空间中的 Euclid 球(或简称为球)具有如下形式,

$$B(x_c, r) = \{x | \|x - x_c\|_2 \leq r\} = \{x | (x^T - x_c^T)(x - x_c) \leq r^2\}$$

其中 $r \geq 0, \|\cdot\|_2$ 表示 Euclid 范数, 即 $\|u\|_2 = (u^T u)^{\frac{1}{2}}$ 。向量 x_c 是球心, 标量 r 为半径。 $B(x_c, r)$ 由与球心 x_c 距离不超过 r 的所有点组成。Euclid 球的另一个常见表达式为:

$$B(x_c, r) = \{x_c + ru | \|u\|_2 \leq 1\}$$

Euclid 球是凸集, 即如果 $\|x_1 - x_c\|_2 \leq r, \|x_2 - x_c\|_2 \leq r$, 并且 $0 \leq \theta \leq 1$, 那么

$$\begin{aligned} \|\theta x_1 + (1 - \theta)x_2 - x_c\|_2 &= \|\theta(x_1 - x_c) + (1 - \theta)(x_2 - x_c)\|_2 \\ &\leq \theta\|(x_1 - x_c)\|_2 + (1 - \theta)\|(x_2 - x_c)\|_2 \\ &\leq r \end{aligned}$$

设 $\|\cdot\|$ 是 \mathbb{R}^n 中的任意范数。那么, 集合 $\{x | \|x - x_c\| \leq r\}$ 是以 r 为半径, x_c 为球心的范数球。根据范数的三角不等式性质可知, 范数球是一个凸集。

椭球

椭球是具有如下的形式的凸集:

$$\mathcal{E} = \{x | (x - x_c)^T \mathbf{P}^{-1} (x - x_c) \leq 1\}$$

其中 $\mathbf{P} = \mathbf{P}^T \succ 0$, 即 \mathbf{P} 是对称正定矩阵。向量 $x_c \in \mathbb{R}^n$ 是椭球的中心。矩阵 \mathbf{P} 决定了椭球从 x_c 向各个方向扩展的幅度。 \mathcal{E} 的半轴长度由 $\sqrt{\lambda_i}$ 给出, 其中 λ_i 是 \mathbf{P} 的特征值。球可以看做 $\mathbf{P} = r^2 \mathbf{I}$ 的椭球。

多面体

多面体被定义为有限个线性等式和不等式的解集。

$$P = \{x | a_i^T x \leq b_i, i = 1, \dots, m, c_j^T x = d_j, j = 1, \dots, p\}$$

或使用紧凑形式:

$$P = \{x | \mathbf{A}x \preceq \mathbf{b}, \mathbf{C}x = \mathbf{d}\}$$

其中

$$\mathbf{A} = \begin{pmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_m^T \end{pmatrix}, \mathbf{C} = \begin{pmatrix} \mathbf{c}_1^T \\ \vdots \\ \mathbf{c}_p^T \end{pmatrix},$$

这里 \preceq 表示的是 \mathbb{R}^m 上的分量不等式, 如果 $\mathbf{u} \preceq \mathbf{v}$, 那么向量 \mathbf{u} 和 \mathbf{v} 的每个分量都有 $u_i \preceq v_i, i = 1, \dots, m$ 。

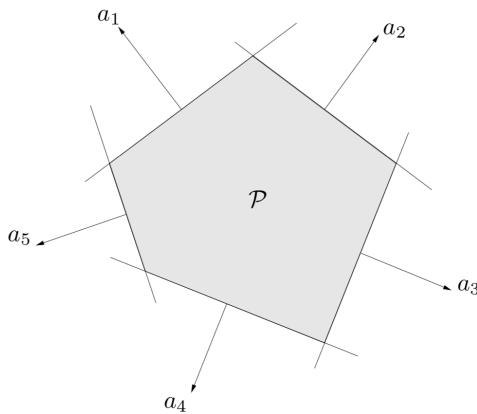


图 10.7: 多面体 \mathcal{P} (阴影所示) 是外法向量为 a_1, \dots, a_5 的五个半空间的交集。

因此, 多面体是有限个半空间和超平面的交集。仿射集合(例如子空间、超平面、直线)、射线、线段和半空间都是多面体。显而易见, 多面体是凸集。有界的多面体有时也称为多胞形, 但也有一些作者反过来使用这两个概念(即用多胞形表示具有上面形式的集合, 而当其有界时称为多面体)。图 1.7 显示了由五个半空间的交集组成的多面体。

单纯形

单纯形是一类重要的多面体。设 $k+1$ 个点 $v_0, \dots, v_k \in \mathbb{R}^n$ 仿射独立, 则 $v_1 - v_0, \dots, v_k - v_0$ 线性独立。那么, 这些点决定了一个单纯形, 如下所示:

$$C = \text{conv}\{v_0, \dots, v_k\} = \{\theta_0 v_0 + \dots + \theta_k v_k \mid \theta \geq 0, \mathbf{1}^T \theta = 1\}$$

其中 $\mathbf{1}$ 表示所有分量均为 1 的向量。

例 10.2.3. 一些常见的单纯形。 1 维单纯形是一条线段; 2 维单纯形是一个三角形(包含其内部); 3 维单纯形是一个四面体。

单位单纯形是由零向量和单位向量 $\mathbf{0}, e_1, \dots, e_n \in \mathbb{R}^n$ 决定的 n 维单纯形。它可以表示为满足下列条件的向量的集合,

$$x \succeq 0, \quad \mathbf{1}^T x \leq 1$$

概率单纯形是有单位向量 $e_1, \dots, e_n \in \mathbb{R}^n$ 决定的 $n-1$ 维单纯形。它是满足下列条件的向量的集合,

$$x \succeq 0, \quad \mathbf{1}^T x = 1$$

概率单纯形中的向量对应于含有 n 个元素的集合的概率分布, x_i 可理解为第 i 个元素的概率。

有限集合 $\{v_1, \dots, v_k\}$ 的凸包是

$$\text{conv}\{v_1, \dots, v_k\} = \{\theta_1 v_1 + \dots + \theta_k v_k | \theta \succeq 0, \mathbf{1}^T \theta = 1\}$$

它是一个有界的多面体，可以用线性等式和不等式的集合表示。

半正定锥

我们用 S^n 表示对称 $n \times n$ 矩阵的集合，即

$$S^n = \{\mathbf{X} \in \mathbb{R}^{n \times n} | \mathbf{X} = \mathbf{X}^T\}$$

这是一个维数为 $n(n+1)/2$ 的向量空间。我们用 S_+^n 表示对称半正定矩阵的集合：

$$S_+^n = \{\mathbf{X} \in S^n | \mathbf{X} \succeq 0\}$$

用 S_{++}^n 表示对称正定矩阵集合：

$$S_{++}^n = \{\mathbf{X} \in S_+^n | \mathbf{X} \succ 0\}$$

集合 S_+^n 是一个凸锥：如果 $\theta_1, \theta_2 \geq 0$ 并且 $A, B \in S_+^n$ ，那么 $\theta_1 A + \theta_2 B \in S_+^n$ 。从半正定矩阵的定义可以直接得到：对于任意 $x \in \mathbb{R}^n$ ，如果 $A \succeq 0, B \succeq 0$ ，那么，就有

$$x^T(\theta_1 A + \theta_2 B)x = \theta_1 x^T A x + \theta_2 x^T B x \geq 0$$

10.2.3 保持凸集的运算

本节将描述一些保持凸集的运算，利用它们，我们可以从一个凸集构造出其他凸集。

集合运算

交集 交集运算是保凸的：如果 S_1 和 S_2 是凸集，那么 $S_1 \cap S_2$ 也是凸集。这个性质可以扩展到无穷个集合的交：如果对于任意 $\alpha \in \mathcal{A}$ 都有 S_α 是凸集，那么， $\cap_{\alpha \in \mathcal{A}} S_\alpha$ 也是凸集。（子空间和仿射集合对于任意交运算也是封闭的。）作为一个简单的例子，多面体是半空间和超平面（它们都是凸集）的交集，因而是凸的。

例 10.2.4. 半正定锥 S_+^n 可以表示为，

$$\bigcap_{z \neq 0} \{X \in S^n | z^T X z \geq 0\}.$$

对于任意 $z \neq 0$ ， $z^T X z$ 是关于 X 的（不恒等于零的）线性函数，因此集合

$$\{X \in S^n | z^T X z \geq 0\}$$

实际上就是 S^n 的半空间。由此可见，半正定锥是无穷个半空间的交集，因此是凸的。

例 10.2.5. 考虑集合

$$S = \{x \in \mathbb{R}^m | |p(t)| \leq 1 \text{ 对于 } |t| \leq \pi/3\}$$

其中 $p(t) = \sum_{k=1}^m x_k \cos(kt)$ 。集合 S 可以表示为无穷多个平板的交集： $S = \bigcap_{|t| \leq \pi/3} S_t$ ，其中

$$S_t = \{x | -1 \leq (\cos(t), \dots, \cos(mt))x \leq 1\}$$

因此， S 是凸的。对于 $m = 2$ 的情况，它的定义和集合可见图 1.8 和图 1.9。

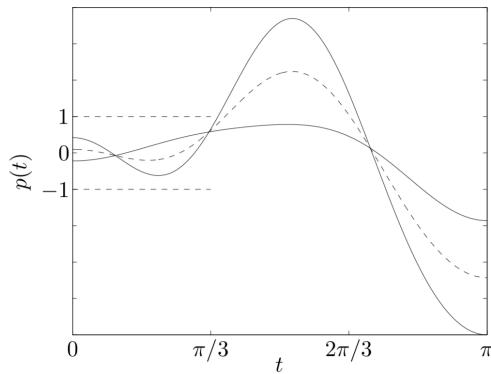


图 10.8: 对应于 $m = 2$ 中的点的三角多项式. 虚线所示的三角多项式是另外两个的平均.

草稿请勿外传

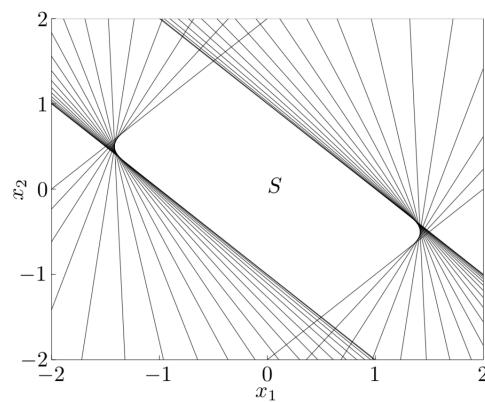


图 10.9: 图中央的白色区域显示了 $m = 2$ 情况下例定义的集合 s . 这个集合是无限多个 (图中显示了其中 20 个) 平板的交集, 所以是凸的。

在上述例子中，通过将集合表示为（可能无穷多个）半空间的交集来证明集合的凸性。反过来，我们也可以看到：每个闭凸集 S 是（通常为无限多个）半空间的交集。事实上，一个闭凸集 S 是包含它的所有半空间的交集：

$$S = \bigcap \{\mathcal{H} | \mathcal{H} \text{是半空间}, S \subseteq \mathcal{H}\}$$

和运算 两个集合的和可以定义为：

$$S_1 + S_2 = \{x + y | x \in S_1, y \in S_2\}$$

如果 S_1 和 S_2 是凸集，那么， $S_1 + S_2$ 是凸的。可以看出，如果 S_1 和 S_2 是凸的，那么其直积或 Cartesian 乘积

$$S_1 \times S_2 = \{(x_1, x_2) | x_1 \in S_1, x_2 \in S_2\}$$

也是凸集。

我们也可以考虑 $S_1, S_2 \in \mathbb{R}^n \times \mathbb{R}^m$ 的部分和，定义为

$$S = \{(x, y_1 + y_2) | (x, y_1) \in S_1, (x, y_2) \in S_2\}$$

其中 $x \in \mathbb{R}^n, y_i \in \mathbb{R}^m$ 。 $m = 0$ 时，部分和给出了 S_1 和 S_2 的交集； $n = 0$ ，部分和等于集合之和。凸集的部分和仍然是凸集。

集合投影 一个凸集向它的某几个坐标的投影是凸的，即：如果 $S \subseteq \mathbb{R}^m \times \mathbb{R}^n$ 是凸集，那么是凸集。

函数映射

仿射函数 假设 $S \subseteq \mathbb{R}^n$ 是凸集，并且 $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 是仿射函数，那么， S 在 f 下的像

$$f(S) = \{f(x) | x \in S\}$$

是凸的。类似地，如果 $f : \mathbb{R}^k \rightarrow \mathbb{R}^n$ 是仿射函数，那么， S 在 f 下的原像

$$f^{-1}(S) = \{x | f(x) \in S\}$$

是凸的。

两个简单的例子是伸缩和平移。如果 $S \subseteq \mathbb{R}^n$ 是凸集， $\alpha \in \mathbb{R}$ 并且 $a \in \mathbb{R}^n$ ，那么，集合 αS 和 $S + a$ 是凸的，其中

$$\alpha S = \{\alpha x | x \in S\}, \quad S + a = \{x + a | x \in S\}$$

例 10.2.6. 线性矩阵不等式的解。条件

$$A(x) = x_1 A_1 + \cdots + x_n A_n \preceq B$$

称为关于 x 的线性矩阵不等式（LMI），其中， $B, A_i \in S^m$ 。（注意它与有序线性不等式

$$a^T x = x_1 a_1 + \cdots + x_n a_n \leq b$$

的相似性，其中 $b, a_i \in \mathbb{R}$ 。

线性矩阵不等式的解 $\{x | A(x) \succeq B\}$ 是凸集。事实上，它是半正定锥在由 $f(x) = B - A(x)$ 给定的仿射映射 $f : \mathbb{R}^n \rightarrow S^m$ 下的原像。

透视函数

我们定义 $P : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$, $P(z, t) = z/t$ 为透视函数, 其定义域为 $\mathbb{K} = \mathbb{R}^n \times \mathbb{R}_+$ 。透视函数对向量进行伸缩, 或称为规范化, 使得最后一维分量为 1 并舍弃之。

如果 $C \subseteq \mathbb{K}$ 是凸集, 那么它的象

$$P(C) = \{P(x) | x \in C\}$$

也是凸集。这个结论很直观: 通过小孔观察一个凸的物体, 可以得到凸的像。为解释这个事实, 下面我们将说明在透视函数作用下, 线段将被映射成线段。

假设 $x = (\tilde{x}, x_{n+1}), y = (\tilde{y}, y_{n+1}) \in \mathbb{R}^{n+1}$ 并且 $x_{n+1} > 0, y_{n+1} > 0$ 。那么, 对于 $0 \leq \theta \leq 1$ 。

$$P(\theta x + (1 - \theta)y) = \frac{\theta \tilde{x} + (1 - \theta)\tilde{y}}{\theta x_{n+1} + (1 - \theta)y_{n+1}} = \mu P(x) + (1 - \mu)P(y)$$

其中,

$$\mu = \frac{\theta x_{n+1}}{\theta x_{n+1} + (1 - \theta)y_{n+1}} \in [0, 1]$$

θ 和 μ 之间的关系是单调的: 当 θ 在 0, 1 间变化时 (形成线段 $[x, y]$), μ 也在 0, 1 间变化 (形成线段 $[P(x), P(y)]$)。这说明 $P([x, y]) = [P(x), P(y)]$ 。

现在假设 C 是凸的, 并且有 $C \subseteq \mathbb{K}$, 即对于所有 $x \in C, x_{n+1} > 0$ 及 $x, y \in C$ 。为显示 $P(C)$ 的凸性, 我们需要说明线段 $[P(x), P(y)]$ 在 $P(C)$ 中。这条线段是线段 $[x, y]$ 在 P 的象, 因而属于 $P(C)$ 。

一个凸集在透视函数下的原象也是凸的: 如果 $C \subseteq \mathbb{R}^n$ 为凸集, 那么

$$P^{-1}(C) = \{(x, t) \in \mathbb{R}^{n+1} | x/t \in C, t > 0\}$$

是凸集。

为证明这点, 假设 $(x, t) \in P^{-1}(C), (y, s) \in P^{-1}(C), 0 \leq \theta \leq 1$ 。我们需要说明

$$\theta(x, t) + (1 - \theta)(y, s) \in P^{-1}(C)$$

即

$$\frac{\theta x + (1 - \theta)y}{\theta t + (1 - \theta)s} \in C$$

(显然地, $\theta t + (1 - \theta)s > 0$)。这可从下式看出,

$$\frac{\theta x + (1 - \theta)y}{\theta t + (1 - \theta)s} = \mu(x/t) + (1 - \mu)(y/s)$$

其中,

$$\mu = \frac{\theta t}{\theta t + (1 - \theta)s} \in [0, 1]$$

线性分式函数

线性分式函数由透视函数和仿射函数复合而成。设 $g : \mathbb{R}^n \rightarrow \mathbb{R}^{m+1}$ 是仿射的，即

$$g(x) = \begin{bmatrix} A \\ c^T \end{bmatrix} x + \begin{bmatrix} b \\ d \end{bmatrix}$$

其中 $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$ 并且 $d \in \mathbb{R}$ 。则由 $f = P \circ g$ 给出的函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$

$$f(x) = (Ax + b)/(c^T x + d), \mathbb{K} = \{x | c^T x + d > 0\}$$

称为线性分式(或投射)函数。如果 $c = 0, d > 0$ ，则 f 的定义域为 \mathbb{R}^n ，并且 f 是仿射函数。因此，我们可以将仿射和线性函数视为特殊的线性分式函数。

类似于透视函数，线性分式函数也是保凸的。如果 C 是凸集并且在 f 的定义域中(即任意 $x \in C$ 满足 $c^T x + d > 0$)，那么 C 的象 $f(C)$ 也是凸集。根据前述的结果可以直接得到这个结论： C 在仿射映射下的象是凸的，并且在透视函数 P 下的映射(即 $f(C)$)是凸的。类似地，如果 $C \subseteq \mathbb{R}^m$ 是凸集，那么其原象 $f^{-1}(C)$ 也是凸的。

这个例子不是很恰当，换成 beyod 的 Figure 2.16 对应的例子

例 10.2.7. 条件概率。设 u 和 v 是分别在 $\{1, \dots, n\}$ 和 $\{1, \dots, m\}$ 中取值的随机变量，并且 p_{ij} 表示概率 $\text{prob}(u = i, v = j)$ 。那么条件概率 $f_{ij} = \text{prob}(u = i | v = j)$ 由下式给出

$$f_{ij} = \frac{p_{ij}}{\sum_{k=1}^n p_{kj}}$$

因此， f 可以通过一个线性分式映射从 p 得到。可以知道，如果 C 是一个关于 (u, v) 的联合密度的凸集，那么相应的 u 的条件密度(给定 v)的集合也是凸集。

10.2.4 分离与支撑超平面

分离超平面

本节中我们将阐述一个在之后非常重要的想法：用超平面或仿射函数将两个不相交的凸集分离开来。

定义 10.2.5. 假设 C 和 D 是两个不相交的凸集，即 $C \cap D = \emptyset$ 。如果仿射函数 $a^T x - b$ 在 C 中非正，而在 D 中非负，那么，超平面 $\{x | a^T x = b\}$ 被称为集合 C 和 D 的分离超平面，或者说超平面分离了集合 C 和 D 。参见图 1.10。

这一结果可以表述为如下超平面分离定理。

定理 10.2.1. 超平面分离定理：假设 C 和 D 是两个不相交的凸集，即 $C \cap D = \emptyset$ 。那么，存在 $a \neq 0$ 和 b ，使得对于所有 $x \in C$ ，都有 $a^T x \leq b$ ，并且对于所有 $x \in D$ ，都有 $a^T x \geq b$ 。

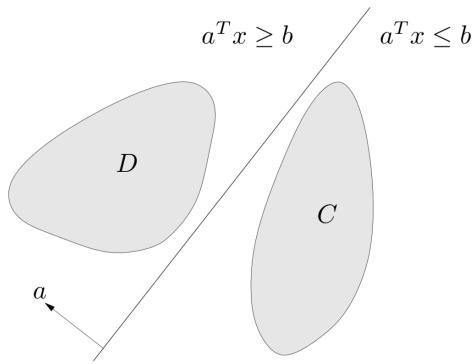


图 10.10: 超平面 $\{x | a^T x = b\}$ 分离了两个不相交的凸集 C 和 D 。仿射函数 $a^T x - b$ 在 C 上非正而在 D 上非负。

证明. 这里我们对一个特殊的情况给予证明。我们假设 C 和 D 的 (Euclid) 距离为正，这里的距离定义为

$$\text{dist}(C, D) = \inf\{\|u - v\|_2 | u \in C, v \in D\}$$

并且存在 $c \in C$ 和 $d \in D$ 达到这个最小距离，即 $\|c - d\|_2 = \text{dist}(C, D)$

定义

$$a = d - c, b = \frac{\|d\|_2^2 - \|c\|_2^2}{2}$$

我们将显示仿射函数

$$f(x) = a^T x - b = (d - c)^T (x - (1/2)(d + c))$$

在 C 中非正而在 D 中非负，即超平面 $\{x | a^T x = b\}$ 分离了 C 和 D 。这个超平面与连接 c 和 d 之间的线段相垂直并且穿过其中点。

我们首先证明 f 在 D 中非负。关于 f 在 C 中非正的证明是相似的 (只需将 C 和 D 交换并考虑 $-f$ 即可)。假设存在一个点 $u \in D$ ，并且

$$f(u) = (d - c)^T (u - (1/2)(d + c)) < 0$$

我们可以将 $f(u)$ 表示为

$$f(u) = (d - c)^T (u - d + (1/2)(d - c)) = (d - c)^T (u - d) + (1/2) \|d - c\|_2^2$$

这意味着 $(d - c)^T (u - d) \leq 0$ 。于是，我们观察到

$$\left. \frac{d}{dt} \|d + t(u - d) - c\|_2^2 \right|_{t=0} = 2(d - c)^T (u - d) \leq 0$$

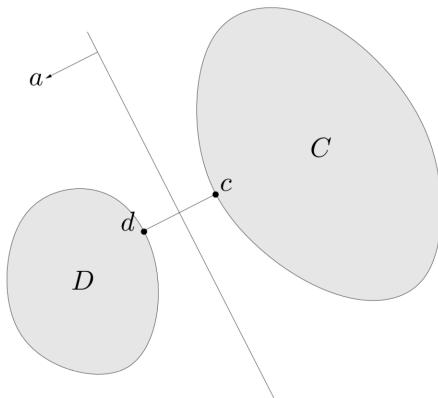


图 10.11: 两个凸集分离超平面的构造。 $c \in C$ 和 $d \in D$ 是两个集合中最靠近彼此的一个点对。分离超平面垂直并等分 c 和 d 直接的线段。

因此, 对于足够小的 $t > 0$ 及 $t \leq 1$ 我们有

$$\|d + t(u - d) - c\|_2 \leq \|d - c\|_2$$

即点 $d + t(u - d)$ 比 d 更靠近 c 。因为 D 是包含 d 和 u 的凸集, 我们有 $d + t(u - d) \in D$ 。但这是不可能的, 因为根据假设, d 应当是 D 中离 C 最近的点。□

严格分离 如果之前构造的分离超平面满足更强的条件. 即对于任意 $x \in C$ 有 $a^T x < b$, 并且对于任意 $x \in D$ 有 $a^T x > b$. 则称其为集合 C 和 D 的严格分离。

支撑超平面

设 $C \subseteq \mathbb{R}^n$ 而 x_0 是其边界 $\text{bd}C$ 上的一点, 即

$$x_0 \in \text{bd}C = \text{cl}C \setminus \text{int}C.$$

如果 $a \neq 0$. 并且对任意 $x \in C$ 满足 $a^T x \leq a^T x_0$. 那么称超平面 $\{x | a^T x = a^T x_0\}$ 为集合 C 在点 x_0 处的支撑超平面。这等于说点 x_0 与集合 C 被超平面所分离。其几何解释是超平面 $\{x | a^T x = a^T x_0\}$ 与 C 相切于点 x_0 , 而且半空间 $\{x | a^T x \leq a^T x_0\}$ 包含 C 。

一个基本的结论, 称为支撑超平面定理, 表明对于任意非空的凸集 C 和任意 $x_0 \in \text{bd}C$. 在 x_0 处存在 C 的支撑超平面。支撑超平面定理从超平面分离定理很容易得到证明。需要区分两种情况. 如果 C 的内部非空, 对于 $\{x_0\}$ 和 $\text{int}C$ 应用超平面分离定理可以直接得到所需的结论。如果 C 的内部是空集, 则 C 必处于小于 n 维的一个仿射集合中, 并且任意包含这个仿射集合的超平面一定包含 C 和 x_0 , 这是一个(平凡的)支撑超平面。

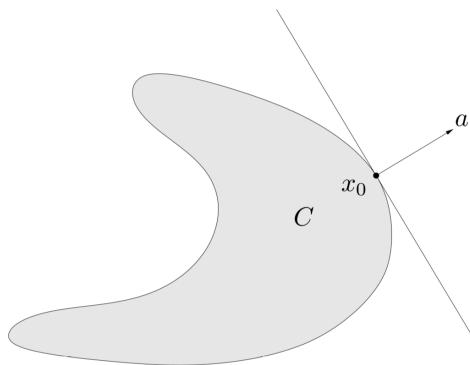


图 10.12: 超平面 $\{\mathbf{x} | \mathbf{a}^T \mathbf{x} = \mathbf{a}^T \mathbf{x}_0\}$ 在 x_0 处支撑 \mathcal{C}

10.3 凸函数

本节我们将首先给出凸函数的定义和相关的例子，然后给出凸函数的判定条件，最后给出保持凸函数的一些基本运算。

10.3.1 凸函数的定义和基本性质

凸函数定义

定义 10.3.1. 如果函数 $f : \mathbb{K} \rightarrow \mathbb{R}$ 的定义域 \mathbb{K} 是凸集，且对于任意 $x, y \in \mathbb{K}$ 和任意 $0 \leq \theta \leq 1$ ，有

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y) \quad (10.9)$$

那么称函数 f 是 \mathbb{K} 上的凸函数。

从几何意义上讲，上述不等式意味着点 $(x, f(x))$ 和 $(y, f(y))$ 之间的线段，即从 x 到 y 的弦，在函数 f 的图像上方（如图 1.13 所示）。称函数 f 是严格凸函数，如果式子中的不等式当 $x \neq y$ 以及 $0 < \theta < 1$ 时严格成立。称函数 f 是凹函数，如果函数 $-f$ 是凸函数。称函数 f 是严格凹函数的，如果 $-f$ 严格凸函数。

对于仿射函数，上面的不等式总成立，因此，所有的仿射函数（包括线性函数）既是凸函数又是凹函数。反之，若某个函数是既凸又凹的，则一定是仿射函数。

函数是凸的，当且仅当其在与其定义域相交的任何直线上都是凸的。换言之，函数 f 是凸的，当且仅当对于任意 $x \in \mathbb{K}$ 和任意向量 v ，函数 $g(t) = f(x+tv)$ 是凸的（其定义域为 $\{t | x+tv \in \mathbb{K}\}$ ）。这个性质非常有用，因为它容许我们通过将函数限制在直线上来判断其是否是凸函数。

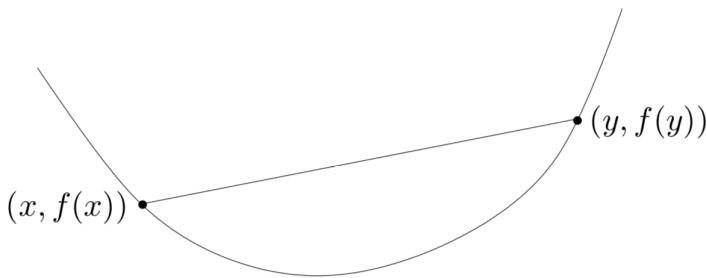


图 10.13: 凸函数示意图。图上任意两点之间的弦（即线段）都在函数图像之上。

凸函数定义的扩展

通常可以定义凸函数在定义域 \mathbb{K} 外的值为 ∞ ，从而将这个凸函数延伸至全空间 \mathbb{R}^n 。

定义 10.3.2. 如果 f 是凸函数，按照如下方式定义它的扩展函数 $\tilde{f}: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$

$$\tilde{f} = \begin{cases} f(x) & x \in \mathbb{K} \\ \infty & x \notin \mathbb{K} \end{cases}$$

扩展函数 \tilde{f} 是定义在全空间 \mathbb{R}^n 上的，取值集合为 $\mathbb{R} \cup \{\infty\}$ 。我们也可以从扩展函数 \tilde{f} 的定义中确定原函数 f 的定义域，即 $\mathbb{K} = \{x | \tilde{f} \leq \infty\}$ 。

类似地，可以通过定义凹函数在定义域外的取值为 $-\infty$ 对其进行延伸。

基本性质-Jessen 不等式

如图 1.13 所示，对于任意 $x, y \in \mathbb{K}$ 和任意 $0 \leq \theta \leq 1$ ，凸函数 f 满足不等式

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$

这个不等式被为 Jessen 不等式。此不等式可以很方便地扩展至更多点的凸组合：如果函数 f 是凸函数， $x_1, \dots, x_k \in \mathbb{K}, \theta_1, \dots, \theta_k \geq 0$ 且 $\theta_1 + \dots + \theta_k = 1$ ，则下式成立

$$f(\theta_1 x_1 + \dots + \theta_k x_k) \leq \theta_1 f(x_1) + \dots + \theta_k f(x_k)$$

考虑凸集时，此不等式可以扩展至无穷项和、积分以及期望。例如，如果在 $S \subseteq \mathbb{K}$ 上 $p(x) \geq 0$ 且 $\int_S p(x)dx = 1$ ，则当相应的积分存在时，下式成立

$$f\left(\int_S p(x)xdx\right) \leq \int_S f(x)p(x)dx$$

扩展到更一般的情况，我们可以采用其支撑属于 \mathbb{K} 的任意概率测度。如果 x 是随机变量，事件 $x \in \mathbb{K}$ 发生的概率为 1，函数 f 是凸函数，当相应的期望存在时，我们有

$$f(\mathbf{E}x) \leq \mathbf{E}f(x)$$

设随机变量 x 的可能取值为 $\{x_1, x_2\}$ ，相应的取值概率为 $\text{prob}(x = x_1) = \theta, \text{prob}(x = x_2) = 1 - \theta$ ，则由一般形式 $f(\mathbf{E}x) \leq \mathbf{E}f(x)$ 可以得到 $f(\theta x_1 + (1 - \theta)x_2) \leq \theta f(x_1) + (1 - \theta)f(x_2)$ 。所以 $f(\mathbf{E}x) \leq \mathbf{E}f(x)$ 可以刻画凸性：如果函数 f 不是凸函数，那么存在随机变量 $x, x \in \mathbb{K}$ 以概率 1 发生，使 $f(\mathbf{E}x) > \mathbf{E}f(x)$ 。

上述所有不等式均被称为 **Jensen 不等式**。

10.3.2 凸函数举例

前文已经提到所有的线性函数和仿射函数均为凸函数（同时也是凹函数），本节给出更多的凸函数和凹函数的例子。

实数集 \mathbb{R} 上常见的凸函数

首先考虑 \mathbb{R} 上的一些函数，其自变量为 x 。用 \mathbb{R}_{++} 表示正实数，用 \mathbb{R}_+ 表示非负实数，以后同。

- **指数函数**。对任意 $a \in \mathbb{R}$ ，函数 e^{ax} 在 \mathbb{R} 上是凸的。
- **幂函数**。当 $a \geq 1$ 或 $a \leq 0$ 时， x^a 是在 \mathbb{R}_{++} 上的凸函数，当 $0 \leq a \leq 1$ 时 x^a 是在 \mathbb{R}_{++} 上的凹函数。
- **绝对值幂函数**。当 $p \geq 1$ 时，函数 $|x|^p$ 在 \mathbb{R} 上是凸函数。
- **对数函数**。函数 $\log x$ 在 \mathbb{R}_{++} 上的凹函数。
- **负熵**。函数 $x \log x$ 在其定义域上是凸函数。

空间 \mathbb{R}^n 上常见的凸函数

下而我们给出 \mathbb{R}^n 上的一些例子。

- **范数**。 \mathbb{R}^n 上的任意范数均为凸函数。
- **最大值函数**。函数 $f(x) = \max\{x_1, \dots, x_n\}$ 在 \mathbb{R}^n 上是凸的。
- **几何平均**。几何平均是其定义域上面的凹函数。

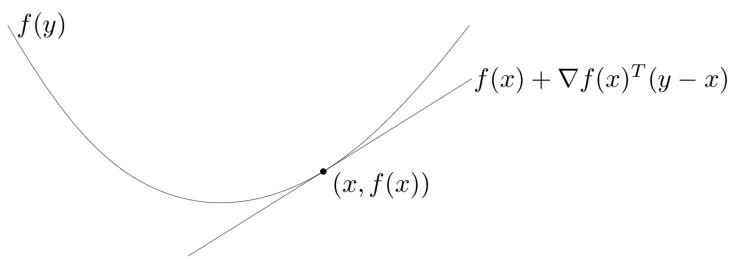


图 10.14: 如果函数 f 是凸的且可微, 那么对于任意 $x, y \in \mathbb{K}$ 有 $f(x) + \nabla f(x)^T(y - x) \leq f(y)$

10.3.3 凸函数的判定条件

一阶条件: f 可微

假设 f 可微 (即其梯度 ∇f 在开集 \mathbb{K} 内处处存在), 则函数 f 是凸函数的充要条件是 \mathbb{K} 是凸集, 且对于任意 $x, y \in \mathbb{K}$, 都有

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) \quad (10.10)$$

成立。图 1.14 描述了上述不等式。

由 $f(x) + \nabla f(x)^T(y - x)$ 得出的关于 y 的仿射函数恰好是函数 f 在点 x 附近的一阶 Taylor 近似。

不等式 (1.10) 表明, 根据凸函数的局部信息 (即它在某点的函数值及其导数), 可以得到一些全局信息。这也许是凸函数最重要的信息, 由此可以解释凸函数及凸优化问题的一些非常重要的性质。由不等式可以知道, 如果 $\nabla f(x) = 0$, 那么对于所有的 $y \in \mathbb{K}$, 都有 $f(y) \geq f(x)$, 也就是说, x 是函数 f 的全局极小值点。

严格凸性同样可以由一阶条件刻画: 函数 f 严格凸的充要条件是 \mathbb{K} 是凸集且对于任意 $x, y \in \mathbb{K}, x \neq y$, 有

$$f(y) > f(x) + \nabla f(x)^T(y - x)$$

对于凹函数, 亦存在与之对应的一阶条件: 函数 f 是凹函数的充要条件是 \mathbb{K} 是凸集且对于任意 $x, y \in \mathbb{K}$, 下式成立

$$f(y) \leq f(x) + \nabla f(x)^T(y - x)$$

证明. 为了证明上式, 先考虑 $n = 1$ 的情况: 我们证明可微函数 $f: \mathbb{R} \rightarrow \mathbb{R}$ 是凸函数的充要条件是对于 \mathbb{K} 内的任意 x 和 y , 有:

$$f(y) \leq f(x) + f'(x)(y - x)$$

首先假设 f 是凸函数, 且 $x, y \in \mathbb{K}$ 。因为 \mathbb{K} 是凸集(某个区间), 对于任意 $0 < t \leq 1$, 我们有 $x + t(y - x) \in \mathbb{K}$, 由函数 f 的凸性可得:

$$f(x + t(y - x)) \leq (1 - t)f(x) + tf(y)$$

将上式两端同除 t 可得

$$f(y) \geq f(x) + \frac{f(x + t(y - x)) - f(x)}{t}$$

令 $t \rightarrow 0$, 可以得到不等式。

为了证明充分性, 假设对 \mathbb{K} (某个区间) 内的任意 x 和 y 。函数满足不等式

$$f(y) \leq f(x) + f'(x)(y - x)$$

选择任意 $x \neq y, 0 \leq \theta \leq 1$, 令 $z = \theta x + (1 - \theta)y$ 。两次应用不等式可得:

$$f(x) \geq f(z) + f'(z)(x - z)$$

$$f(y) \geq f(z) + f'(z)(y - z)$$

将第一个不等式乘以 θ , 第二个不等式乘以 $(1 - \theta)$, 并将两者相加可得

$$\theta f(x) + (1 - \theta)f(y) \geq f(z)$$

从而说明了函数 f 是凸的。

现在来证明一般情况, 即 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 。设 $x, y \in \mathbb{R}^n$, 考虑过这两点的直线上的函数 f , 即函数 $g(t) = f(ty + (1 - t)x)$, 此函数对 t 求导可得 $g'(t) = \nabla f(ty + (1 - t)x)^T(y - x)$ 。

首先假设函数 f 是凸的, 则函数 g 是凸的, 由前面的讨论可得 $g(1) \geq g(0) + g'(0)$, 即

$$f(y) \geq f(x) + \nabla f(x)^T(y - x)$$

再假设此不等式对于任意 x 和 y 均成立, 因此若 $ty + (1 - t)x \in \mathbb{K}$ 以及 $\bar{t}y + (1 - \bar{t})x \in \mathbb{K}$, 我们有

$$f(ty + (1 - t)x) \geq f(\bar{t}y + (1 - \bar{t})x) + \nabla f(\bar{t}y + (1 - \bar{t})x)^T(y - x)(t - \bar{t})$$

即 $g(t) \geq g(\bar{t}) + g'(\bar{t})(t - \bar{t})$, 说明了函数 g 是凸的。 \square

一阶条件: f 不可微

添加次梯度的定义以及一阶条件, 参考 Algorithms for Convex Optimization 的 Definition 3.4 (Subgradient) 以及 Lemma 3.5

二阶条件

现在假设函数 f 二阶可微, 即对于开集 \mathbb{K} 内的任意一点, 它的 Hessian 矩阵或者二阶导数 $\nabla^2 f$ 存在, 则函数 f 是凸函数的充要条件是: 其 Hessian 矩阵是半正定阵, 即对于所有的 $x \in \mathbb{K}$, 都有

$$\nabla^2 f(x) \geq 0$$

对于 \mathbb{R} 上的函数，上述条件可以简化为： $f''(x) \geq 0$ (\mathbb{K} 是凸的，即一个区间)，此条件说明函数 f 的导数是非减的。条件 $\nabla^2 f(x) \geq 0$ 从几何上可以理解为函数图像在点 x 处有具有正(向上)的曲率。关于二阶条件的证明作为习题留给读者完成。添加二阶条件的证明，参考 Algorithms for Convex Optimization 的 Theorem 3.6 (Second-order notion of convexity)

例 10.3.1. 二次函数。考虑二次函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ，其定义域为 $\mathbb{K} = \mathbb{R}^n$ ，其表达式为

$$f(x) = (1/2)x^T Px + q^T x + r$$

其中 $P \in S^n, q \in \mathbb{R}^n, r \in \mathbb{R}$ 。对于任意 x , $\nabla^2 f(x) = P$ 。因此，函数 f 是凸函数，当且仅当 $P \succeq 0$ (反之， f 是凹函数，当且仅当 $P \preceq 0$)。

对于任意二次函数，严格凸比较容易表达；函数 f 是严格凸的，当且仅当 $P \succ 0$ (函数是严格凹的当且仅当 $P \prec 0$)。

类似地，函数 f 是凹函数的充要条件是， \mathbb{K} 是凸集且对于任意 $x \in \mathbb{K}, \nabla^2 f(x) \leq 0$ 。严格凸的条件可以部分由二阶条件刻画。如果对于任意的 $x \in \mathbb{K}$ 有 $\nabla^2 f(x) > 0$ ，则函数 f 是严格凸。反过来则不一定成立：例如，函数 $f : \mathbb{R} \rightarrow \mathbb{R}$ 其表达式为 $f(x) = x^4$ ，它是严格凸的，但是在 $x = 0$ 处，二阶导数为零。

10.3.4 保凸运算

非负加权求和

显而易见，如果函数 f 是凸函数且 $\alpha \geq 0$ ，则函数 αf 也是凸函数，如果函数 f_1 和 f_2 都是凸函数，则他们的和 $f_1 + f_2$ 也是凸函数。将非负伸缩以及求和运算结合起来，可以看出，凸函数的非负加权求和仍然是凸函数，即如果 $f_i, i = 1, \dots, m$ 是凸函数， $w_i \geq 0, i = 1, \dots, m$ ，那么，函数

$$f = w_1 f_1 + \dots + w_m f_m$$

也是凸函数。类似地，凹函数的非负加权求和仍然是凹函数。严格凸(凹)函数的非负、非零加权求和是严格凸(凹)函数。

这个性质可以扩展至无限项的求和以及积分的情形。例如，如果固定任意 $y \in \mathcal{A}$ ，函数 $f(x, y)$ 关于 x 是凸函数，且对任意 $y \in \mathcal{A}$ ，有 $w(y) \geq 0$ ，则函数

$$g(x) = \int_{\mathcal{A}} w(y) f(x, y) dy$$

关于 x 是凸函数(若此积分存在)。

逐点最大和逐点上、下确界函数

逐点最大函数 如果函数 f_1 和 f_2 均为凸函数，则二者的逐点最大函数

$$f(x) = \max\{f_1(x), f_2(x)\}$$

仍然是凸函数，定义域为 $\mathbb{K} = \mathbb{K}_1 \cap \mathbb{K}_2$ 。这个性质可以很容易验证：任取 $0 \leq \theta \leq 1$ 以及 $x, y \in \mathbb{K}$ ，有

$$\begin{aligned} f(\theta x + (1 - \theta)y) &= \max\{f_1(\theta x + (1 - \theta)y), f_2(\theta x + (1 - \theta)y)\} \\ &\leq \max\{\theta f_1(x) + (1 - \theta)f_1(y), \theta f_2(x) + (1 - \theta)f_2(y)\} \\ &\leq \theta \max\{f_1(x), f_2(x)\} + (1 - \theta) \max\{f_1(y), f_2(y)\} \\ &= \theta f(x) + (1 - \theta)f(y) \end{aligned}$$

从而说明了函数 f 的凸性。同样地，如果函数 f_1, \dots, f_m 为凸函数，则它们的逐点最大函数

$$f(x) = \max\{f_1(x), \dots, f_m(x)\}$$

仍然是凸函数。

例 10.3.2. 分片线性函数. 函数

$$f(x) = \max\{a_1^T x + b_1, \dots, a_L^T x + b_L\}$$

定义了一个分片线性（实际上是仿射）函数（具有 L 个或者更少的子区域）。因为它是一系列仿射函数的逐点最大函数，所以它是凸函数。

反之亦成立：任意具有 L 个或者更少子区域的分片线性凸函数都可以表述成上述形式。

逐点上确界函数 逐点最大的性质可以扩展至无限个凸函数的逐点上确界。如果对于任意 $y \in \mathcal{A}$ ，函数 $f(x, y)$ 关于 x 都是凸的，则函数 g

$$g(x) = \sup_{y \in \mathcal{A}} f(x, y)$$

关于 x 亦是凸的。此时，函数 g 的定义域为

$$\mathbf{dom} g = \{x | (x, y) \in \mathbb{K}, \forall y \in \mathcal{A}, \sup_{y \in \mathcal{A}} f(x, y) < \infty\}$$

类似地，一系列凹函数的逐点下确界仍然是凹函数。

例 10.3.3. 集合的支撑函数。令集合 $C \subseteq \mathbb{R}^n$ ，且 $C \neq \emptyset$ ，定义集合 C 的支撑函数 S_C 为

$$S_C(x) = \sup\{x^T y | y \in C\}$$

其定义域为 $\mathbf{dom} S_C = \{x | \sup_{y \in C} x^T y < \infty\}$ 。对于任意 $y \in C$ ， $x^T y$ 是 x 的线性函数，所以 S_C 是一系列线性函数的逐点上确界函数，因此是凸函数。

例 10.3.4. 矩阵范数。 考虑函数 $f(X) = \|X\|_2$ ，定义域为 $\mathbb{K} = \mathbb{R}^{p \times q}$ ，其中， $\|\cdot\|_2$ 表示谱函数或者最大奇异值。函数 f 则可以重新写为

$$f(X) = \sup\{u^T X v | \|u\|_2 = 1, \|v\|_2 = 1\}$$

由于它是 X 的一族线性函数的逐点上确界，所以是凸函数。

上述例子表明，一个建立函数凸性的好方法是将其表示为一族仿射函数的逐点上确界。几乎所有的凸函数都可以表示成一族仿射函数的逐点上确界。

定理 10.3.1. 如果函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 是凸函数，其定义域为 $\mathbb{K} = \mathbb{R}^n$ ，我们有

$$f(x) = \sup\{g(x) | g \text{ 是仿射函数}, g(z) \leq f(z), \forall z\}$$

换言之，函数 f 是它所有的仿射全局下估计的逐点上确界。

为了证明该定理，首先引入函数图像和上境图定义：

定义 10.3.3. 函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 的图像定义为：

$$\{(x, f(x)) | x \in \mathbb{K}\}$$

它是 \mathbb{R}^{n+1} 空间的一个子集。函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 的上境图定义为：

$$\text{epi } f = \{(x, t) | x \in \mathbb{K}, f(x) \leq t\}$$

它也是 \mathbb{R}^{n+1} 空间上的一个子集。

下面，我们开始证明定理 1.3.1：

证明. 设函数 f 是凸函数，定义域为 $\mathbb{K} = \mathbb{R}^n$ ，显然下面的不等式成立

$$f(x) \geq \sup\{g(x) | g \text{ 是仿射函数}, g(z) \leq f(z), \forall z\}$$

因为函数 g 是函数 f 的任意仿射下估计，我们有 $g(x) \leq f(x)$ 。为了建立等式，我们说明，对任意 $x \in \mathbb{R}^n$ ，存在仿射函数 g 是函数 f 的全局下估计，并且满足 $g(x) = f(x)$ 。

毫无疑问，函数 f 的上境图是凸集，因此我们在点 $(x, f(x))$ 处可以找到此凸集的支撑超平面，即存在 $a \in \mathbb{R}^n, b \in \mathbb{R}$ 且 $(a, b) \neq 0$ ，使得对任意 $(z, t) \in \text{epi } f$ ，有

$$\begin{bmatrix} a \\ b \end{bmatrix}^T \begin{bmatrix} x - z \\ f(x) - t \end{bmatrix} \leq 0$$

由于 $(z, t) \in \text{epi } f$ 等价于存在 $s \geq 0$ ，使得 $t = f(z) + s$ 。因此，对任意 $z \in \mathbb{K} = \mathbb{R}^n$ 以及所有 $s \geq 0$ ，都有

$$a^T(x - z) + b(f(x) - f(z) - s) \leq 0 \quad (10.11)$$

为了保证不等式(1.11)对所有的 $s \geq 0$ 均成立，必须要 $b \geq 0$ 。如果 $b = 0$ ，对所有的 $z \in \mathbb{R}^n$ ，不等式(1.11)可以简化为 $a^T(x - z) \leq 0$ 。这意味着 $a = 0$ ，于是和假设 $(a, b) \neq 0$ 矛盾。因此， $b > 0$ ，即支撑超平面不是竖直的。

在 $b > 0$ 的情况下，对任意 z ，令 $s = 0$ ，式(1.11)可以重新表述为

$$g(z) = f(x) + (a/b)^T(x - z) \leq f(z)$$

由此说明函数 g 是函数 f 的一个仿射下估计，并且满足 $g(x) = f(x)$ 。 \square

逐点下确界函数 一些特殊形式的最小化同样可以得到凸函数。如果函数 f 关于 (x, y) 是凸函数，集合 C (这里需要将符号 C 全部改为 \mathcal{A} ，与逐点最大化函数的集合符号统一起来)是非空凸集，定义函数

$$g(x) = \inf_{y \in C} f(x, y) \quad (10.12)$$

如果对任意的 x ，都有 $g(x) > -\infty$ ，那么，函数 g 关于 x 是凸函数。此时，函数 g 的定义域是 \mathbb{K} 在 x 方向上的投影，即

$$\mathbf{dom} g = \{x | \exists y \in C, s.t. (x, y) \in \mathbb{K}\}$$

可以利用 Jensen 不等式来证明函数 g 的凸性。

证明. 任取 $x_1, x_2 \in \mathbf{dom} g$ ，令 $\varepsilon > 0$ ，则存在 $y_1, y_2 \in C$ ，使 $f(x_i, y_i) \leq g(x_i) + \varepsilon (i = 1, 2)$ 。设 $\theta \in [0, 1]$ 。我们有

$$\begin{aligned} g(\theta x_1 + (1 - \theta)x_2) &= \inf_{y \in C} f(\theta x_1 + (1 - \theta)x_2, y) \\ &\leq f(\theta x_1 + (1 - \theta)x_2, \theta y_1 + (1 - \theta)y_2) \\ &\leq \theta f(x_1, y_1) + (1 - \theta)f(x_2, y_2) \\ &\leq \theta g(x_1) + (1 - \theta)g(x_2) + \varepsilon \end{aligned}$$

因为上式对任意 $\varepsilon > 0$ 均成立，所以不等式

$$g(\theta x_1 + (1 - \theta)x_2) \leq \theta g(x_1) + (1 - \theta)g(x_2) \quad (10.13)$$

成立。结论得证。 \square

例 10.3.5. 点到某一集合的距离。 某点 x 到集合 $S \subseteq \mathbb{R}^n$ 的距离定义为

$$dist(x, S) = \inf_{y \in S} \|x - y\|$$

函数 $\|x - y\|$ 关于 (x, y) 是凸的。所以，若集合 S 是凸集，则 $dist(x, S)$ 是关于 x 的凸函数。

例 10.3.6. Schur 补。 给定对称矩阵 A 和 C ，假设二次函数

$$f(x, y) = x^T Ax + 2x^T By + y^T Cy$$

关于 (x, y) 是凸函数，即

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succeq 0$$

令 $g(x) = \inf_y f(x, y)$ ，则 $g(x)$ 还可以表述为

$$g(x) = x^T (A - BC^\dagger B^T)x$$

其中， C^\dagger 是矩阵 C 的伪逆。根据极小化的性质， $g(x)$ 是凸函数，因此 $A - BC^\dagger B^T \succeq 0$ 。

如果矩阵 C 可逆，即 $C \succ 0$ ，则矩阵 $A - BC^{-1}B^T$ 称为 C 在矩阵

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$$

中的 Schur 补。

复合函数

本节给定函数 $h : \mathbb{R}^k \rightarrow \mathbb{R}$ 以及 $g : \mathbb{R}^n \rightarrow \mathbb{R}^k$, 定义复合函数 $f = h \circ g : \mathbb{R}^n \rightarrow \mathbb{R}$ 为

$$f(x) = h(g(x)), \quad \mathbb{K} = \{x \in \text{dom } g | g(x) \in \text{dom } h\}$$

我们考虑当函数 f 保凸或者保凹时, 函数 h 和 g 必须满足的条件。

标量复合

考虑 $k = 1$ 的情况, 即 $h : \mathbb{R} \rightarrow \mathbb{R}, g : \mathbb{R}^n \rightarrow \mathbb{R}$ 。

当 $n = 1$ 时, (事实上, 将函数限定在与其定义域相交的任意直线上得到的函数决定了原函数的凸性) 为了找出复合规律, 假设函数 h 和 g 是二次可微的, 且 $\text{dom } g = \text{dom } h = \mathbb{R}$ 。在上述假设下, 函数 f 是凸的等价于 $f'' \geq 0$ (即对所有的 $x \in \mathbb{R}, f''(x) \geq 0$)。复合函数 $f = h \circ g$ 的二阶导数为

$$f''(x) = h''(g(x))g'(x)^2 + h'(g(x))g''(x) \quad (10.14)$$

假设函数 g 是凸函数 ($g'' \geq 0$), 函数 h 是凸函数且非减(即 $h'' \geq 0$ 且 $h' \geq 0$), 从式(1.14)可以得出 $f'' \geq 0$, 即函数 f 是凸函数。类似地, 由式(1.14)可以得出如下结论

- 如果 h 是凸函数且非减, g 是凸函数, 则 f 是凸函数
 - 如果 h 是凸函数且非增, g 是凹函数, 则 f 是凸函数
 - 如果 h 是凹函数且非减, g 是凹函数, 则 f 是凹函数
 - 如果 h 是凹函数且非增, g 是凸函数, 则 f 是凹函数
- (10.15)

当 $n > 1$ 时, 即 $\text{dom } g = \mathbb{R}^n, \text{dom } h = \mathbb{R}$, 此时, 不再假设函数 h 和 g 可微, 相似的复合规则仍然成立:

- 如果 h 是凸函数且 \tilde{h} 非减, g 是凸函数, 则 f 是凸函数
 - 如果 h 是凸函数且 \tilde{h} 非增, g 是凹函数, 则 f 是凸函数
 - 如果 h 是凹函数且 \tilde{h} 非减, g 是凹函数, 则 f 是凹函数
 - 如果 h 是凹函数且 \tilde{h} 非增, g 是凸函数, 则 f 是凹函数
- (10.16)

其中 \tilde{h} 是 h 的扩展函数。这些结论和(1.15)不同之处在于要求扩展函数 \tilde{h} 在整个 \mathbb{R} 上非增或者非减。

下面开始证明如下结论: 如果 h 是凸函数且 \tilde{h} 非减, g 是凸函数, 则 $f = h \circ g$ 是凸函数。
(1.16)的其它结论可类似证明。

证明. 假设 $x, y \in \mathbb{K}, 0 \leq \theta \leq 1$ 。由于 $x, y \in \mathbb{K}$, 我们有 $x, y \in \text{dom } g$, 且 $g(x), g(y) \in \text{dom } h$ 。因为 $\text{dom } g$ 是凸集, 则有 $\theta x + (1 - \theta)y \in \text{dom } g$ 。由函数 g 的凸性可得

$$g(\theta x + (1 - \theta)y) \leq \theta g(x) + (1 - \theta)g(y) \quad (10.17)$$

由 $g(x), g(y) \in \text{dom } h$ 可得 $\theta g(x) + (1 - \theta)g(y) \in \text{dom } h$ 。即式(1.17)的右端在 $\text{dom } h$ 内。根据假设 \tilde{h} 是非减的，可以理解为其定义域在负方向上无限延伸。式(1.17)的右端在 $\text{dom } h$ 内，我们知道其左侧仍在定义域内，即 $g(\theta x + (1 - \theta)y) \in \text{dom } h$ ，因此 \mathbb{K} 是凸集。

根据前提条件， \tilde{h} 非减，利用不等式(1.17)，我们有

$$h(g(\theta x + (1 - \theta)y)) \leq h(\theta g(x) + (1 - \theta)g(y)) \quad (10.18)$$

由函数 h 的凸性，可得

$$h(\theta g(x) + (1 - \theta)g(y)) \leq \theta h(g(x)) + (1 - \theta)h(g(y)) \quad (10.19)$$

综合式(1.18)和式(1.19)，可得

$$h(g(\theta x + (1 - \theta)y)) \leq \theta h(g(x)) + (1 - \theta)h(g(y))$$

结论得证。 \square

下面介绍几个复合函数的例子。

- 如果 g 是凸函数，则 $\exp g(x)$ 是凸函数。
- 如果 g 是凹函数且大于 0，则 $\log g(x)$ 是凹函数。
- 如果 g 是凹函数且大于零，则 $1/g(x)$ 是凸函数。
- 如果 g 是凸函数且不小于零， $p \geq 1$ 则 $g(x)^p$ 是凸函数。
- 如果 g 是凸函数，则 $-\log(-g(x))$ 在 $\{x|g(x) \leq 0\}$ 上是凸函数。

向量复合

考虑 $k > 1$ 的情况， $f(x) = h(g(x)) = h(g_1(x), \dots, g_k(x))$ ，其中， $h : \mathbb{R}^k \rightarrow \mathbb{R}$, $g_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, k$ 。

和上节一样，首先假设 $n = 1$ 。和 $k = 1$ 的情形类似，为了得到复合规则，假设函数二次可微，且 $\text{dom } g = \mathbb{R}$, $\text{dom } h = \mathbb{R}^k$ 。对函数 f 进行二次微分，可得

$$f''(x) = g'(x)^T \nabla^2 h(g(x)) g'(x) + \nabla h(g(x))^T g''(x) \quad (10.20)$$

上式可以看成是式(1.14)对应的向量形式。此时，需要判断在什么条件下对所有 x ，有 $f''(x) \geq 0$ （或者对所有 x ，有 $f(x)'' \leq 0$ ，则 f 是凹函数）。利用式(1.20)，得到如下复合规则：

如果 h 是凸函数且在每维分量上 h 非减， g_i 是凸函数，则 f 是凸函数；

如果 h 是凸函数且在每维分量上 h 非增， g_i 是凹函数，则 f 是凸函数；

如果 h 是凹函数且在每维分量上 h 非减， g_i 是凹函数，则 f 是凹函数。

然后考虑 $n > 1$ 的情况，不假设 h 或 g 可微，类似的复合结论仍然成立。此时，不仅 h 满足单调性条件，其扩展函数 \tilde{h} 同样必须满足。

为了更好地理解扩展函数 \tilde{h} 单调性的含义，考虑凸函数 $h : \mathbb{R}^k \rightarrow \mathbb{R}$ ，且 \tilde{h} 非减，即对任意 $u \leq v$ ，有 $\tilde{h}(u) \leq \tilde{h}(v)$ 。这说明了如果 $v \in \text{dom } h$ ，则 $u \in \text{dom } h$: h 的定义域在方向 $-\mathbb{R}_+^k$ 上必须无限延伸。这个条件可以紧凑地描述为 $\text{dom } h - \mathbb{R}_+^k = \text{dom } h$ 。

下面介绍几个复合函数的例子。

- 函数 $h(z) = \log(\sum_{i=1}^k e^{z_i})$ 是凸函数且在每一维分量上非减，因此只要 g_i 是凸函数，那么 $\log(\sum_{i=1}^k e^{g_i})$ 就是凸函数。
- 对 $0 < p \leq 1$ ，定义在 \mathbb{R}_+^k 上的函数 $h(z) = (\sum_{i=1}^k z_i^p)^{1/p}$ 是凹的，且其扩展值延伸（当 $z \not\geq 0$ 时为 $-\infty$ ）在每维分量上非减，则若 g_i 是凹函数且非负， $f(x) = \left(\sum_{i=1}^k g_i(x)^p\right)^{1/p}$ 是凹函数。
- 设 $p \geq 1, g_1, \dots, g_k$ 是凸函数且非负，则函数 $\left(\sum_{i=1}^k g_i(x)^p\right)^{1/p}$ 是凸函数。
- 几何平均数 $h(z) = \left(\prod_{i=1}^k z_i\right)^{1/k}$ ，定义域为 \mathbb{R}_+^k ，它是凹函数，且其扩展值延伸在每维分量上非减，因此若 g_1, \dots, g_k 是非负凹函数，他们的集合平均 $\left(\prod_{i=1}^k g_i\right)^{1/k}$ 也是非负凹函数。

透视函数

给定函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ，则 f 的透视函数 $g : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ 定义为

$$g(x, t) = tf(x/t),$$

其定义域为

$$\text{dom } g = \{(x, t) | x/t \in \mathbb{K}, t > 0\}$$

透视运算是保凸运算：如果函数 f 是凸函数，则其透视函数 g 也是凸函数。类似地，若 f 是凹函数，则 g 亦是凹函数。

可以从多个角度来证明此结论，从上境图的角度，当 $t > 0$ ，我们有

$$\begin{aligned} (x, t, s) \in \text{epi } g &\iff tf(x/t) \leq s \\ &\iff f(x/t) \leq s/t \\ &\iff (x/t, s/t) \in \text{epi } f \end{aligned}$$

因此， $\text{epi } g$ 是透视映射下 $\text{epi } f$ 的原像，此透视映射将 (u, v, w) 映射为 $(u, w)/v$ 。因此， $\text{epi } g$ 是凸集， g 是凸函数。

例 10.3.7. Euclid 范数平方。 \mathbb{R}^n 上的凸函数 $f(x) = x^T x$ 的透视函数定义为

$$g(x, t) = t(x/t)^T (x/t) = \frac{x^T x}{t}$$

当 $t > 0$ 时，它关于 (x, t) 是凸函数。

例 10.3.8. 负对数。考虑 \mathbb{R}_{++} 上的凸函数 $f(x) = -\log x$, 其透视函数为

$$g(x, t) = -t \log(x/t) = t \log(t/x) = t \log t - t \log x$$

在 \mathbb{R}_{++}^2 上它是凸函数。函数 g 被称为关于 t 和 x 的相对熵。当 $x = 1$ 时, g 为负熵函数。根据函数 g 的凸性, 可以推导出其它函数的凸性或凹性。例如: 定义两个向量 $u, v \in \mathbb{R}_{++}^n$ 的相对熵

$$\sum_{i=1}^n u_i \log(u_i/v_i)$$

因为它可以转化为 (u, v) 的相对熵和线性函数的求和, 所以是凸函数。

例 10.3.9. 设 $f : \mathbb{R}^m \rightarrow \mathbb{R}$ 是凸函数, $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n, d \in \mathbb{R}$ 。定义

$$g(x) = (c^T x + d) f((Ax + b)/(c^T x + d))$$

其定义域为

$$\text{dom } g = \{x | c^T x + d > 0, (Ax + b)/(c^T x + d) \in \mathbb{K}\}$$

则 g 是凸函数。

10.3.5. 共轭函数

本节介绍一个运算, 它将在后续章节发挥重要的作用。

定义

定义 10.3.4. 设函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$, 定义函数 $f^* : \mathbb{R}^n \rightarrow \mathbb{R}$ 为

$$f^*(y) = \sup_{x \in \mathbb{K}} (y^T x - f(x))$$

此函数称为函数 f 的共轭函数, 定义域是由使上述上确界有限的 y 组成的, 即差值 $y^T x - f(x)$ 在 \mathbb{K} 有界。

图 1.15 描述了此定义。 f^* 是凸函数, 这是因为它是一系列关于 y 的凸函数(实质上是仿射函数)的逐点上确界。无论 f 是否是凸函数, f^* 都是凸函数。

上一些常见凸函数的共轭函数

我们从一些简单的例子开始描述共轭函数的一些规律。在此基础上我们可以写出很多常见凸函数的共轭函数的解析形式。

- 仿射函数

$f(x) = ax + b$ 。作为 x 的函数, 当且仅当 $y = a$, 即为常数时, $yx - ax - b$ 有界。因此, 共轭函数 f^* 的定义域为单点集 a , 且 $f^*(a) = -b$ 。

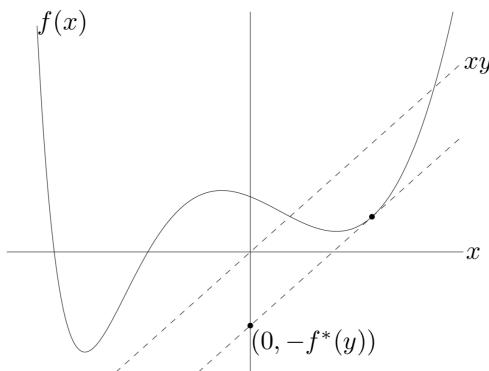


图 10.15: 函数 $f: \mathbb{R} \rightarrow \mathbb{R}$ 以及某一 $y \in \mathbb{R}$ 。共轭函数 $f^*(y)$ 是线性函数 yx 和 $f(x)$ 之间的最大差值, 见图中虚线所示。如果 f 可微, 在满足 $f'(x) = y$ 的点 x 处差值最大。

- 负对数函数

$f(x) = -\log x$, 定义域为 $\mathbb{K} = \mathbb{R}_{++}$ 。当 $y \geq 0$ 时, 函数 $xy + \log x$ 无上界, 当 $y < 0$ 时, 在 $x = -\frac{1}{y}$ 处函数达到最大。因此, 定义域为 $\mathbb{K}^* = \{y | y < 0\} = -\mathbb{R}_{++}$, 共轭函数为 $f^*(y) = -\log(-y) - 1(y < 0)$ 。

- 指数函数

$f(x) = e^x$ 当 $y < 0$ 时, 函数 $xy - e^x$ 无界。当 $y > 0$ 时, 函数 $xy - e^x$ 在 $x = \log y$ 处达到最大值。因此, $f^*(y) = y \log y - y$ 。当 $y = 0$ 时, $f^*(y) = \sup_x -e^x = 0$ 。综合起来, $\mathbb{K}^* = \mathbb{R}_+$, $f^*(y) = y \log y - y$ (规定 $0 \log 0 = 0$)。

- 负熵函数

$f(x) = x \log x$, 定义域为 $\mathbb{K} = \mathbb{R}_+$ 。对所有 y , 函数 $xy - x \log x$ 关于 x 在 \mathbb{R}_+ 上有界, 因此 $\mathbb{K}^* = \mathbb{R}$ 。在 $x = e^{y-1}$ 处, 函数达到最大值。因此 $f^*(y) = e^{y-1}$ 。

- 反函数

$f(x) = \frac{1}{x}, x \in \mathbb{R}_{++}$ 。当 $y > 0$ 时, $yx - 1/x$ 无上界。当 $y = 0$ 时, 函数有上确界 0; 当 $y < 0$ 时, 在 $x = (-y)^{-1/2}$ 处达到上确界。因此, $f^*(y) = -2(-y)^{1/2}$ 且 $\mathbb{K}^* = -\mathbb{R}_+$ 。

- 严格凸的二次函数

考虑函数 $f(x) = \frac{1}{2}x^T Qx$, $Q \in S_{++}^n$ 。对于所有的 y , x 的函数 $y^T x - \frac{1}{2}x^T Qx$ 都有上界并在 $x = Q^{-1}y$ 处达到上确界, 因此, $f^*(y) = \frac{1}{2}y^T Q^{-1}y$ 。

- 指示函数

设 I_S 是某个集合 $S \subseteq \mathbb{R}^n$ (不一定是凸集) 的示性函数, 即当 x 在 $\text{dom } I_S = S$ 内时, $I_S(x) = 0$ 。示性函数的共轭函数为 $I_S^*(y) = \sup_{x \in S} y^T x$, 它是集合 S 的支撑函数。

- 范数平方

考虑函数 $f(x) = (1/2)\|x\|^2$, 其中 $\|\cdot\|$ 是范数, 对偶范数为 $\|\cdot\|_*$ 。此函数的共轭函数为 $f^*(y) = (1/2)\|y\|_*^2$ 。一方面, 由 $y^T x \leq \|y\|_* \|x\|$ 可知, 对任意 x 下式成立

$$y^T x - 1/2\|x\|^2 \leq \|y\|_* \|x\| - 1/2\|x\|^2$$

上式右端是关于 $\|x\|$ 的二次函数, 其最大值为 $1/2\|y\|_*^2$ 。因此, 对任意 x , 我们有

$$y^T x - (1/2)\|x\|^2 \leq (1/2)\|y\|_*^2$$

即 $f^*(y) \leq (1/2)\|y\|_*^2$ 。另一方面, 任取满足 $y^T x = \|y\|_* \|x\|$ 的向量 x , 对其进行伸缩, 使得 $\|x\| = \|y\|_*$ 。此时,

$$y^T x - (1/2)\|x\|^2 \leq (1/2)\|y\|_*^2$$

因此, $f^*(y) \geq (1/2)\|y\|_*^2$ 。

基本性质

- Fenchel 不等式

从共轭函数的定义可以知道, 对任意 x 和 y , 不等式

$$f(x) + f^*(y) \geq x^T y$$

成立, 这就是 Fenchel 不等式(当 f 可微时, 亦称为 Young 不等式)。

- 共轭的共轭

“共轭”的名称隐含了凸函数的共轭函数的共轭函数是原函数。也即: 如果函数 f 是凸函数且 f 是闭的, 则 $f^{**} = f$ 。

- 可微函数

可微函数 f 的共轭函数亦称为 f 的 Legendre 变换。

设函数 f 是凸函数且可微, 其定义域为 $\mathbb{K} = \mathbb{R}^n$, 使 $y^T x - f(x)$ 取最大值的 x^* 满足 $y = \nabla f(x^*)$; 反之, 若 x^* 满足 $y = \nabla f(x^*)$, 则 $y^T x - f(x)$ 在 x^* 处取最大值。因此, 如果 $y = \nabla f(x^*)$, 我们有

$$f^*(y) = x^{*T} \nabla f(x^*) - f(x^*)$$

给定任意 y , 可以求解梯度方程 $y = \nabla f(z)$, 从而得到 y 处的共轭函数 $f^*(y)$ 。

我们亦可以换一个角度理解。任选 $z \in \mathbb{R}^n$, 令 $y = \nabla f(z)$, 则

$$f^*(y) = z^T \nabla f(z) - f(z)$$

- 伸缩变换和复合仿射变换

若 $a > 0$ 以及 $b \in \mathbb{R}$, $g(x) = af(x) + b$ 的共轭函数为 $g^*(y) = af^*(y/a) - b$ 。

设 $A \in \mathbb{R}^{n \times n}$ 非奇异, $b \in \mathbb{R}^n$, 则函数 $g(x) = f(Ax + b)$ 的共轭函数为

$$g^*(y) = f^*(A^{-T}y) - b^T A^{-T}y$$

其定义域为 $\text{dom } g^* = A^T \text{dom } f^*$ 。

- 独立函数求和

如果函数 $f(u, v) = f_1(u) + f_2(v)$, 其中 f_1 和 f_2 是凸函数, 共轭函数分别是 f_1^* 和 f_2^* , 则
 $f^*(w, z) = f_1^*(w) + f_2^*(z)$

换言之, 独立凸函数求和的共轭是各个凸函数的共轭求和。(“独立”的含义是各个函数具有不同的变量。)

10.4 凸优化

10.4.1 优化问题

优化问题的标准形式

前面我们已经介绍了优化问题的一般形式, 这里, 首先将该优化问题转化为标准形式:

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq 0, i = 1, \dots, m \\ & && h_j(x) = 0, j = 1, \dots, p \end{aligned} \tag{10.21}$$

按照惯例, 设不等式和等式约束的右端为零。这一点总可以通过对任何非零右端进行减法得到, 类似地, 我们将 $f_i(x) \geq 0$ 表示为 $-f_i(x) \leq 0$ 。函数 f 和 g 的下标要用不用的符号表示, 这里, 我将 h 的下标改成 j , 后面需要全部再修改一下

通常, 我们主要考虑极小化问题。这是因为极大化问题

$$\begin{aligned} & \text{maximize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq 0, i = 1, \dots, m \\ & && h_j(x) = 0, j = 1, \dots, p \end{aligned} \tag{10.22}$$

可以通过在同样的约束下极小化 $-f_0$ 得到求解。

实例

这里可以总结一下本书前面提到的一些优化问题。(10.1 节简单文字介绍前面提到的优化问题, 这里放几个对应的公式)

10.4.2 凸优化问题

凸优化问题的标准形式

定义 10.4.1. 形如

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq 0, \quad i = 1, \dots, m \\ & && a_i^T x = b_i, \quad i = 1, \dots, p \end{aligned} \tag{10.23}$$

的问题称为凸优化问题，其中 f_0, \dots, f_m 为凸函数。特别地，当 $m = p = 0$ 时，式(1.23)被称为无约束凸优化问题。

对比凸优化问题(1.23)和一般优化问题的标准形式问题(1.21)，可以看出，凸优化问题有三个附加要求：

- 目标函数必须是凸的，
- 不等式约束函数必须是凸的，
- 等式约束函数 $h_i(x) = a_i^T x - b_i$ 必须是仿射的。

由凸函数的性质可知：凸优化问题的可行集是凸的。因此，凸优化问题本质上是在一个凸集上极小化一个凸的目标函数。

凹最大化问题

定义 10.4.2. 形如

$$\begin{aligned} & \text{maximize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq 0, \quad i = 1, \dots, m \\ & && a_i^T x = b_i, \quad i = 1, \dots, p \end{aligned} \tag{10.24}$$

的优化问题被称为凹最大化问题，其中，目标函数 f_0 是凹函数，不等式约束函数 f_1, \dots, f_m 是凸函数。

有时候，这个凹最大化问题也被称为凸优化问题，这是因为它可以简单地通过极小化凸目标函数 $-f_0$ 得以求解。对于最小化问题的所有结果、结论及算法都可以简单地转换用于解决最大化问题。因此，本书接下来的部分只考虑最小化问题。

局部最优解与全局最优解

凸优化问题的一个基本性质是其任意局部最优解也是(全局)最优解。为理解这点，设 x 是凸优化问题的局部最优解，即 x 是可行的，并且对于某些 $R > 0$ ，满足以下条件

$$f_0(x) = \inf\{f_0(z) | z \text{ 可行}, \|z - x\|_2 \leq R\} \tag{10.25}$$

现在假设 x 不是全局最优解，即存在一个可行的 y ，使得 $f_0(y) < f_0(x)$ 。显然 $\|y - x\|_2 > R$ ，否则有 $f_0(x) \leq f_0(y)$ 。令

$$z = (1 - \theta)x + \theta y, \quad \theta = \frac{R}{2\|y - x\|_2}$$

则有 $\|z - x\|_2 = R/2 < R$ 。根据可行集的凸性， z 是可行的。根据 f_0 的凸性，我们有

$$f_0(z) \leq (1 - \theta)f_0(x) + \theta f_0(y) < f_0(x)$$

这与式(1.25)矛盾。因此不存在满足 $f_0(y) < f_0(x)$ 的可行解 y ，即 x 是全局最优解。

可微函数 f_0 的最优性准则

有约束凸优化的最优性准则

定理 10.4.1. 设凸优化问题的目标函数 f_0 是可微的, 即对于所有的 $x, y \in \mathbb{K}_0$, 有

$$f_0(y) \geq f_0(x) + \nabla f_0(x)^T(y - x) \quad (10.26)$$

令 X 表示可行集, 即

$$X = \{x | f_i(x) \leq 0, i = 1, \dots, m, a_j^T x = b_j, j = 1, \dots, p\}$$

那么, x 是最优解, 当且仅当 $x \in X$ 且

$$\nabla f_0(x)^T(y - x) \geq 0, \quad \forall y \in X. \quad (10.27)$$

证明. 首先假设 $x \in X$ 满足式(1.27)。那么, 如果 $y \in X$, 根据式(1.26), 我们有 $f_0(y) \geq f_0(x)$ 。这表明 x 是问题(1.21)的一个最优解。

反之, 设 x 是最优解但条件(1.27)不成立, 即对于某些 $y \in X$, 有

$$\nabla f_0(x)^T(y - x) < 0$$

考虑点 $z(t) = ty + (1 - t)x$, 其中 $t \in [0, 1]$ 为参数。因为 $z(t)$ 在 x 和 y 之间的线段上, 而可行集是凸集, 因此 $z(t)$ 可行。我们可断言对于小正数 t , 有 $f_0(z(t)) < f_0(x)$, 这证明了 x 不是最优的。为说明这一点, 注意

$$\frac{d}{dt} f_0(z(t))|_{t=0} = \nabla f_0(x)^T(y - x) < 0$$

所以, 对于小正数 t , 我们有 $f_0(z(t)) < f_0(x)$

□

这个最优性准则可以从几何上进行理解: 如果 $\nabla f_0(x) \neq 0$, 那么意味着 $-\nabla f_0(x)$ 在 x 处定义了可行集的一个支撑超平面。

等式约束凸优化的最优性准则 考虑只含等式约束而没有不等式约束的问题, 即

$$\text{maximize } f_0(x)$$

$$\text{subject to } Ax = b,$$

其可行集是仿射的。假设定义域非空, 可行解 x 的最优性条件为: 对任意满足 $Ay = b$ 的 y ,

$$\nabla f_0(x)^T(y - x) \geq 0$$

都成立。因为 x 可行, 每个可行解 y 都可以写作 $y = x + v$ 的形式, 其中 $v \in \mathcal{N}(A)$ 。**符号 $\mathcal{N}(A)$ 是否在前面定义过, 如果没有, 这边需要说明** 因此, 最优性条件可表示为

$$\nabla f_0(x)^T v \geq 0, \quad \forall v \in \mathcal{N}(A)$$

如果一个线性函数在子空间中非负, 则它在子空间上必恒等于零。因此, 对于任意 $v \in \mathcal{N}(A)$, 我们有 $\nabla f_0(x)^T v = 0$, 换言之

$$\nabla f_0(x) \perp \mathcal{N}(A)$$

利用 $\mathcal{N}(A)^\perp = \mathcal{R}(A^T)$, 最优性条件可以表示为 $\nabla f_0(x) \in \mathcal{R}(A^T)$, 即存在 $v \in \mathbb{R}^p$, 使得

$$\nabla f_0(x) + A^T v = 0$$

同时考虑 $Ax = b$ 的要求 (即要求 x 可行), 这是经典的 Lagrange 乘子最优性条件。

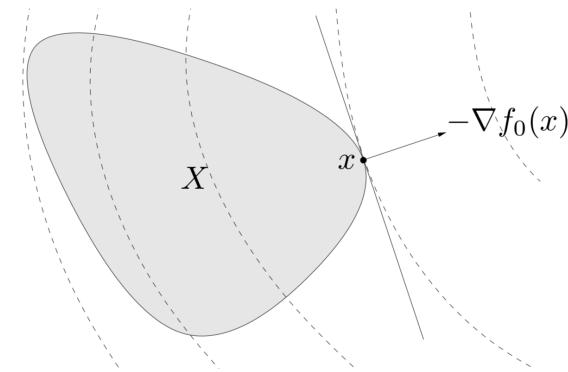


图 10.16: 最优化条件(1.27)的几何解释。可行集 X 由阴影显示。 f_0 的某些等值曲线由虚线显示。点 x 是最优解: $-\nabla f_0(x)$ 定义了 X 在 x 处的一个支撑超平面。

无约束凸优化的最优化准则 对于无约束凸优化问题 (即 $m = p = O$), x 是最优解的条件可以简化为

$$\nabla f_0(x) = 0 \quad (10.28)$$

下面我们看一下它是如何从条件(1.27)中推到出的。设 x 为可行解, 即 $x \in \mathbb{K}_0$ 并且对于所有可行的 y , 都有 $\nabla f_0(x)^T(y - x) \geq 0$ 。因为 f_0 可微, 其定义域是开的, 因此所有充分靠近 x 的点都是可行的。我们取 $y = x - t\nabla f_0(x)$, 其中 $t \in \mathbb{R}$ 为参数。当 t 为小的正数时, y 是可行的, 因此

$$\nabla f_0(x)^T(y - x) = -t\|\nabla f_0(x)\|_2^2 \geq 0$$

从中可知 $\nabla f_0(x) = 0$ 。

无约束凸优化问题的最优解依赖于式(1.28)解的数量。如果式(1.28)无解, 那么, 该问题没有最优解, 即问题无下界或最优值有限但不可达; 如果式(1.28)有多个解, 那么, 这些解就是使得目标函数取最小值的解。

例 10.4.1. 无约束二次规划。考虑极小化二次函数

$$f_0(x) = (1/2)x^T Px + q^T x + r$$

其中, $P \in S_+^*$ (保证了 f_0 为凸函数)。 x 为 f_0 的最优解的充要条件是:

$$\nabla f_0(x) = Px + q = 0$$

根据这个 (线性) 方程无解、有唯一解或多解的不同, 有几种可能的情况

- 如果 $q \notin R(p)$, 则无解。此类情况 f_0 无下界。
- 如果 $P \succ 0$ (意味着 f_0 严格凸的), 则存在唯一的最小解 $x^* = -P^{-1}q$ 。
- 如果 P 奇异但 $q \in R(P)$, 则最优解集合为 (仿射) 集合 $X_{opt} = -P^\dagger q + \mathcal{N}(P)$, 其中 P^\dagger 表示 P 的伪逆。

凸优化问题的等价问题

如果从一个问题的解，很容易得到另一个问题的解，并且反之亦然，我们称两个问题是等价的。例如，考虑问题

$$\begin{aligned} & \text{minimize} && \tilde{f}(x) = \alpha_0 f_0(x) \\ & \text{subject to} && \tilde{f}_i(x) = \alpha_i f_i(x) \leq 0, i = 1, \dots, m \\ & && \beta_j a_j^T x = \beta_j b_j, j = 1, \dots, p \end{aligned} \quad (10.29)$$

其中 $\alpha_i > 0, i = 0, \dots, m$ 且 $\beta_i \neq 0, i = 1, \dots, p$ 。这个问题是通过将优化问题(1.23)的目标函数和不等式约束函数乘以正的常数，并将等式约束函数乘以非零常数得到的。因此，问题(1.29)的可行集与原问题(1.23)是相同的。显然， x 是原问题(1.23)的最优解，当且仅当它也是问题(1.29)的最优解。综上所述，这两个问题是等价的。事实上，这两个问题(1.29)和(1.23)是不同的（除非 α_i 和 β_i 都是 1），因为目标函数和约束函数都不同。

下面介绍几种产生等价问题的方法。

消除等式约束 一个凸问题的等式约束必须是线性的，即具有 $Ax = b$ 的形式。在这种情况下，可以通过寻找 $Ax = b$ 的一个特解 x_0 和域为 A 的零空间的矩阵 F 来消除这些等式约束，从而得到关于 z 的优化问题，

$$\begin{aligned} & \text{minimize} && f_0(Fz + x_0) \\ & \text{subject to} && f_i(Fz + x_0) \leq 0, \quad i = 1, \dots, m \end{aligned}$$

因为凸函数和仿射函数的复合依然是凸的，消除等式约束可以保持问题的凸性。而且消除等式约束的过程（以及从变换后问题的解重构出原问题的解）只需利用标准的线性代数运算。

至少在理论上，这意味着我们可以集中精力于不含等式约束的凸优化问题。但是，在很多情况下，由于消除等式约束会使得问题更难理解和求解，甚至使得求解它的算法失效。因此，有时候还是会保留等式约束。例如，当变量 x 维数很高时，消除等式约束确实有可能破坏问题的稀疏性或其它结构信息。

引入等式约束 在凸优化问题中引入新的变量和等式约束，前提是等式约束是线性的，所得的优化问题仍然是凸的。例如，如果目标函数或约束函数具有 $f_i(A_i x + b_i)$ 的形式，其中 $A_i \in \mathbb{R}^{k_i \times n}$ ，那么，可以引入新的变量 $y_i \in \mathbb{R}^{k_i}$ ，用 $f_i(y_i)$ 替换 $f_i(A_i x + b_i)$ 并添加线性等式约束 $y_i = A_i x + b_i$ 。

添加松弛变量 通过在不等式约束中引入松弛变量 $s_i, i = 1, \dots, m$ ，可以得到新的等式约束 $f_i(x) + s_i = 0$ 。因为凸优化问题中的等式约束必须是仿射的，所以 f_i 必须是仿射函数。换言之，可以在线性不等式约束中引入松弛变量，从而保持原问题的凸性。

优化部分变量 已知等式

$$\inf_{x,y} f(x,y) = \inf_x \tilde{f}(x)$$

成立, 其中 $\tilde{f}(x) = \inf_y f(x,y)$ 。换言之, 可以通过先优化一部分变量再优化另一部分变量来达到函数优化的目的。这个简单而普适的原则可以用来将原问题转换为其等价形式。对于一般形式, 其描述冗长而不直观, 因此, 我们这里仅用一个例子来进行说明。

设变量 $x \in \mathbb{R}^n$ 被分为 $x = (x_1, x_2)$, 其中 $x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}$, 并且 $n_1 + n_2 = n$ 。考虑问题

$$\begin{aligned} & \text{minimize} && f_0(x_1, x_2) \\ & \text{subject to} && f_i(x_1) \leq 0, \quad i = 1, \dots, m_1 \\ & && \tilde{f}_i(x_2) \leq 0, \quad i = 1, \dots, m_2 \end{aligned} \tag{10.30}$$

其约束相互独立, 也就是说每个约束函数只与 x_1 或 x_2 有关。首先优化 x_2 , 定义函数 \tilde{f}_0 为

$$\tilde{f}_0(x_1) = \inf\{f_0(x_1, z) | \tilde{f}_i(z) \leq 0 \quad i = 1, \dots, m_2\}$$

则原问题(1.30)等价于

$$\begin{aligned} & \text{minimize} && \tilde{f}_0(x_1) \\ & \text{subject to} && f_i(x_1) \leq 0, \quad i = 1, \dots, m_1. \end{aligned} \tag{10.31}$$

例 10.4.2. 考虑严格凸的二次目标问题

$$\begin{aligned} & \text{minimize} && x_1^T P_{11} x_1 + 2x_1^T P_{12} x_2 + x_2^T P_{22} x_2 \\ & \text{subject to} && f_i(x_1) \leq 0, i = 1, \dots, m \end{aligned}$$

其中, P_{11} 和 P_{22} 均为对称矩阵, 并且变量 x_2 不受约束。这里, 我们可以解析地优化 x_2 :

$$\inf_{x_2} (x_1^T P_{11} x_1 + 2x_1^T P_{12} x_2 + x_2^T P_{22} x_2) = x_1^T (P_{11} - P_{12} P_{22}^{-1} P_{12}^T) x_1$$

我觉得这个等式是怎么来的可以写得再清楚一点, 就是推导一下 x_2 取什么值时, 得到这个极小值。因此原问题等价于

$$\begin{aligned} & \text{minimize} && x_1^T (P_{11} - P_{12} P_{22}^{-1} P_{12}^T) x_1 \\ & \text{subject to} && f_i(x_1) \leq 0, i = 1, \dots, m \end{aligned} \tag{10.32}$$

注意, 最优化凸函数的部分变量将保持凸性不变。如果问题(1.30)中的目标函数 f_0 是关于变量 x_1 和 x_2 的联合凸函数, 并且 $f_i, i = 1, \dots, m_1$ 和 $\tilde{f}_i, i = 1, \dots, m_2$ 都是凸函数, 那么, 其等价问题(1.31)也是凸的。

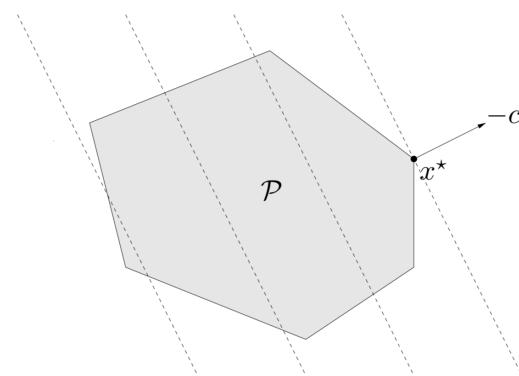


图 10.17: 线性规划的几何解释。可行集 \mathcal{P} 是多面体, 如阴影所示。目标 $c^T x$ 是线性的, 所以其等位曲线是与 c 正交的超平面 (如虚线所示)。点 x^* 是最优的, 它是 \mathcal{P} 中在方向 $-c$ 上最远的点。

10.4.3 常见的凸优化问题

线性规划问题

当目标函数和约束函数都是仿射时, 问题称作线性规划 (LP)。一般的线性规划具有以下形式

$$\begin{aligned} & \text{minimize} && c^T x + d \\ & \text{subject to} && Gx \leq h \\ & && Ax = b \end{aligned} \tag{10.33}$$

公式中关于向量的不等式符号用 \leq 还是 \preceq , 前后都要保持一致, 所以需要全部修改一遍 其中, $G \in \mathbb{R}^{m \times n}, A \in \mathbb{R}^{p \times n}$ 。线性规划问题都是凸优化问题。

有时会将目标函数中的常数 d 省略, 因为它不影响最优解 (以及可行解) 集合。考虑到极大化目标函数 $c^T x + d$ 等价于极小化 $-c^T x - d$ (仍然是凸的), 因此, 具有仿射目标函数和约束函数的最大化问题也被称为线性规划问题。

线性规划的几何解释可以见图 1.17: 线性规划(1.33)的可行集是多面体 P ; 这一问题是在 P 上极小化仿射函数 $c^T x + d$ (或者极小化线性函数 $c^T x$)。

线性规划的标准形和不等式形 线性规划(1.33)的两种特殊情况已经被广泛深入地研究, 以至于分别被赋予了特殊的名称: 标准形和不等式形。标准形的线性规划

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && Ax = b \\ & && x \geq 0 \end{aligned} \tag{10.34}$$

中仅有的不等式都是关于变量的非负性约束 $x \geq 0$ 。如果线性规划没有等式约束, 则称为不等式形的线性规划, 常写作

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && Ax \leq b \end{aligned} \tag{10.35}$$

有时需要将一般的线性规划(1.33)转化为标准形式。第一步是为不等式引入松弛变量 s , 得到

$$\begin{aligned} & \text{minimize} && c^T x + d \\ & \text{subject to} && Gx + s = h \\ & && Ax = b \\ & && s \geq 0 \end{aligned}$$

第二步是将变量 x 表示为两个非负变量 x^+ 和 x^- 的差, 即 $x = x^+ - x^-$, $x^+, x^- \geq 0$, 则有

$$\begin{aligned} & \text{minimize} && c^T x^+ - c^T x^- + d \\ & \text{subject to} && G^T x^+ - G^T x^- + s = h \\ & && Ax^+ - Ax^- = b \\ & && x^+ \geq 0, x^- \geq 0, s \geq 0 \end{aligned}$$

这是标准形式的线性规划, 其优化变量是 x^+ , x^- 和 s 。

这些技巧可以将很多问题构造成标准型的线性规划。在非正式的情况下, 通常将一个可以转换为线性规划形式的问题称为线性规划, 即使他本身并不具备线性规划(1.33)的形式。

实例 线性规划出现在非常多的领域和应用中。这里我们给出一些典型的例子。

食谱问题

一份健康的饮食包含 m 种不同的营养, 每种至少需要 b_1, \dots, b_m 。我们可以从 n 种食物中选择非负的量 x_1, \dots, x_n 以构成一份食谱。单位第 j 种食品含有营养 i 的量为 a_{ij} , 而价格为 c_j 。我们希望设计出一份最便宜的满足营养需求的食谱。这一问题可以描述为线性规划

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && Ax \geq b \\ & && x \geq 0 \end{aligned}$$

运筹学外传

此问题的一些变化仍然可以构造为线性规划。例如我们可以要求营养的精确量(这将给出一个等式约束)。我们也可以如前给出下界那样, 规定每种营养的上界。

多面体的 Chebyshev 中心

考虑在多面体中寻找最大 *Euclid* 球的问题, 多面体由线性不等式表示为

$$P = \{x \in \mathbb{R}^n | a_i^T x \leq b_i, i = 1, \dots, m\}$$

(最优球的中心称为多面体的 Chebyshev 中心; 它是多面体内部最深的点, 即离边界最远的点) 将这个球重新表述为

$$B = \{x_c + u | \|u\|_2 \leq r\}$$

这个问题的变量是球的中心 $x_c \in \mathbb{R}^n$ 和半径 r ; 我们希望在 $B \subseteq P$ 的约束下极大化 r 。

我们从较为简单的约束开始考虑: B 在半空间 $a_i^T x \leq b_i$ 中, 即

$$\|u\|_2 \leq r \implies a_i^T(x_c + u) \leq b_i \quad (10.36)$$

因为

$$\sup\{a_i^T u | \|u\|_2 \leq r\} = r\|a_i\|_2$$

所以, 我们可以将式(1.36)写作

$$a_i^T x_c + r\|a_i\|_2 \leq b_i \quad (10.37)$$

这是 x_c 和 r 的线性不等式。换言之, 决定球在半空间 $a_i^T x \leq b_i$ 的约束可以写为一个线性不等式。

因此 $B \subseteq P$ 当且仅当对于所有 $i = 1, \dots, m$, 式(1.37)均成立。所以, Chebyshev 中心可以通过求解关于 x_c 和 r 的线性规划问题

$$\begin{aligned} &\text{maximize} && r \\ &\text{subject to} && a_i^T x_c + r\|a_i\|_2 \leq b_i, \quad i = 1, \dots, m \end{aligned}$$

得到。

二次优化问题

当凸优化问题(1.23)的目标函数是(凸)二次型, 并且约束函数为仿射函数时, 该问题称为二次规划(QP), 具体表示为

$$\begin{aligned} &\text{minimize} && \frac{1}{2}x^T Px + q^T x + r \\ &\text{subject to} && Gx \leq h \\ & && Ax = b \end{aligned} \quad (10.38)$$

其中, $P \in S_+^n, G \in \mathbb{R}^{m \times n}, A \in \mathbb{R}^{p \times n}$ 。在二次规划问题中, 我们再在多面体上极小化一个凸二次函数, 如图 哪张图, 没有标注出来所示:

如果在(1.23)中，除了目标函数，其不等式约束也是(凸)二次型，例如

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2}x^T P_0 x + q_0^T x + r_0 \\ & \text{subject to} \quad \frac{1}{2}x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \dots, m \\ & \quad Ax = b \end{aligned} \tag{10.39}$$

其中， $P_i \in S_+^n, i = 0, 1, \dots, m$ 。这一问题称为二次约束二次规划 (QCQP)。在 QCQP 中，当 $P_i > 0$ 时，在椭圆的交集构成的可行集上最小化凸二次函数。

线性规划是二次规划的特例，通过在(1.38)中取 $P = 0$ 可得。二次规划(因此也包括线性规划)是二次约束二次规划的特例，通过在(1.39)中令 $P_i = 0, i = 1, \dots, m$ 可得。

实例

最小二乘及回归

最小化凸二次函数

$$\|Ax - b\|_2^2 = x^T A^T A x - 2b^T A x + b^T b$$

的问题是一个(无约束的)二次规划。在很多领域中，都会遇到这个问题，并有很多的名字，例如回归分析或最小二乘逼近。这个问题很简单，有著名的解析解 $x = A^\dagger b$ ，其中， A^\dagger 是 A 的伪逆。

增加线性不等式约束后的问题称为约束回归或约束最小二乘。此问题不再有简单的解析解。作为一个例子，我们考虑具有变量上下界约束的回归问题，即

$$\begin{aligned} & \text{minimize} \quad \|Ax - b\|_2^2 \\ & \text{subject to} \quad l_i \leq x_i \leq u_i, \quad i = 1, \dots, n \end{aligned}$$

这是一个二次规划。

二阶锥规划

一个与二次规划紧密相关的问题是二阶锥规划(SOCP)：

$$\begin{aligned} & \text{minimize} \quad f^T x \\ & \text{subject to} \quad \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \dots, m \\ & \quad Fx = g \end{aligned} \tag{10.40}$$

其中， $x \in \mathbb{R}^n$ 为优化变量， $A_i \in \mathbb{R}^{n_i \times n}$ 。我们称这种形式的约束

$$\|Ax + b\|_2 \leq c^T x + d$$

为二阶锥约束，其中 $A \in \mathbb{R}^{k \times n}$ 。这是因为它等同于要求仿射函数 $(Ax + b, c^T x + d)$ 在 \mathbb{R}^{k+1} 的二阶锥中。

当 $c_i = 0, i = 1, \dots, m$ 时，SOCP 等同于 QCQO。类似地，如果 $A_i = 0, i = 1, \dots, m$ ，SOCP 退化为线性规划。

鲁棒线性规划

考虑不等式形式的线性规划

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && a_i^T x \leq b_i, \quad i = 1, \dots, m \end{aligned}$$

其中的参数 c, a_i, b_i 含有一些不确定性或变化。为简洁起见，假设 c 和 b_i 是固定的，并且已知 a_i 在给定的椭球中：

$$a_i \in \mathcal{E}_i = \{\bar{a}_i + P_i u \mid \|u\|_2 \leq 1\}$$

其中， $P_i \in \mathbb{R}^{n \times n}$ 。现在要求对于参数 a_i 的所有可能值，这些约束都必须满足，则可以得到相应的鲁棒线性规划，具体表示为

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && a_i^T x \leq b_i, \quad \forall a_i \in \mathcal{E}_i, \quad i = 1, \dots, m \end{aligned} \tag{10.41}$$

对于所有 $a_i \in \mathcal{E}_i$ ，都有 $a_i^T x \leq b_i$ 成立，这个约束条件等价于

$$\sup\{a_i^T x \mid a_i \in \mathcal{E}_i\} \leq b_i$$

其左端可以重新写作

$$\begin{aligned} \sup\{a_i^T x \mid a_i \in \mathcal{E}_i\} &= \bar{a}_i^T x + \sup\{u^T P_i^T x \mid \|u\|_2 \leq 1\} \\ &= \bar{a}_i^T x + \|P_i^T x\|_2 \end{aligned}$$

形式。因此，鲁棒线性约束可以简化为

$$\bar{a}_i^T x + \|P_i^T x\|_2 \leq b_i$$

这是一个二阶锥约束。因此，鲁棒线性规划(1.41)也可以看成是一个 SOCP

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && \bar{a}_i^T x + \|P_i^T x\|_2 \leq b_i, \quad i = 1, \dots, m \end{aligned}$$

注意，范数项在这里发挥了正则化的作用，它们可以避免 x 在参数 a_i 的值得考虑的不确定性方向上变得过大。

几何规划

本节将描述一类优化问题，它们的自然形式并不是凸的。但通过变量替换或目标函数、约束函数的变换，可以将它们转换为凸优化问题。

函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathbb{K} = \mathbb{R}_{++}^n$ 定义为

$$f(x) = cx_1^{a_1}x_2^{a_2} \cdots x_n^{a_n}$$

其中 $c > 0, a_i \in \mathbb{R}$ 。它被称为**单项式函数**或简称为**单项式**。单项式的指数 a_i 可以是任意实数，包括分数或负数，但系数 c 必须非负。（“单项式”与代数中的标准定义矛盾，在那里指数必须是非负整数，但这个矛盾不会有混淆。）单项式的和，即具有下列形式的函数称为**正项式函数**(具有 K 项)，或简称为**正项式**

$$f(x) = \sum_{k=1}^K c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}$$

其中 $c_k > 0$ 。

正项式对于加法，数乘和非负的伸缩变换是封闭的。单项式对于数乘和除是封闭的。如果将正项式乘以一个单项式，其结果是一个正项式；类似地，将正项式除以一个单项式，其结果仍为正项式。

具有下列形式的优化问题

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \leq 1, \quad i = 1, \dots, m \\ & && h_i(x) = 1, \quad i = 1, \dots, p \end{aligned}$$

被称为**几何规划 (GP)**，其中 f_0, \dots, f_m 为正项式， h_1, \dots, h_p 为单项式。这个问题的定义域 $D = \mathbb{R}_{++}^n$ ；约束 $x \succ 0$ 是隐式的。

凸形式的几何规划 几何规划（一般）不是凸优化问题，但是通过变量代换以及目标、约束函数的转换，它们可以被转换成凸问题。

我们用 $y_i = \log x_i$ 定义变量，因此 $x_i = e^{y_i}$ 。如果 f 是 x 的单项式 $f(x) = cx_1^{a_1}x_2^{a_2} \cdots x_n^{a_n}$ ，那么

$$\begin{aligned} f(x) &= f(e^{y_1}, \dots, e^{y_n}) \\ &= c(e^{y_1})^{a_1} \cdots (e^{y_n})^{a_n} \\ &= e^{a^T y + b} \end{aligned}$$

其中 $b = \log c$ 。由此可知，通过变量变换 $y_i = \log x_i$ 可以将一个单项式函数转换为以仿射函数为指数的函数。

类似地，如果 f 表示为

$$f(x) = \sum_{k=1}^K c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}$$

其中， $a_k = (a_{1k}, \dots, a_{nk})$, $b_k = \log c_k$ ，那么，令 $y_i = \log x_i$ ，则有

$$f(x) = \sum_{k=1}^K e^{a_k^T y + b_k}$$

换句话说，通过变量变换，可以把正项式转换为以仿射函数为指数的函数求和公式。因此，几何规划可以用新变量 y 的形式表示为

$$\begin{aligned} & \text{minimize} \quad \sum_{k=1}^{K_0} e^{a_{0k}^T y + b_{0k}} \\ & \text{subject to} \quad \sum_{k=1}^{K_i} e^{a_{ik}^T y + b_{ik}} \leq 1, \quad i = 1, \dots, m \\ & \quad e^{g_i^T y + h_i} = 1, \quad i = 1, \dots, p \end{aligned}$$

其中 $a_{ik} \in \mathbb{R}^n, i = 0, \dots, m$ ，包含了以正项式为指数的不等式约束， $g_i \in \mathbb{R}^n, i = 1, \dots, p$ ，包含了原几何规划中以单项式为指数的等式约束。

现在采用对数函数对目标函数和约束函数进行转换，从而得到问题

$$\begin{aligned} & \text{minimize} \quad \tilde{f}_0(y) = \log\left(\sum_{k=1}^{K_0} e^{a_{0k}^T y + b_{0k}}\right) \\ & \text{subject to} \quad \tilde{f}_i(y) = \log\left(\sum_{k=1}^{K_i} e^{a_{ik}^T y + b_{ik}}\right) \leq 0, i = 1, \dots, m \\ & \quad \tilde{h}_i(y) = g_i^T y + h_i = 0, i = 1, \dots, p \end{aligned}$$

因为函数 \tilde{f}_i 是凸的， \tilde{h}_i 是仿射的，所以该问题是一个凸优化问题。我们称其为凸形式的几何规划。为将其与原始的几何规划相区别，我们称原始形式的几何规划为正项式形式的几何规划。

需要注意的是正项式形式的几何规划与凸形式的几何规划之间的转换并不涉及任何运算；两个问题的数据是相同的。需要改变的仅仅是目标函数和的函数的形式。

如果正项式目标函数和所有约束函数都只含有一项，即都是单项式，那么凸形式的几何规划将退化为（一般的）线性规划。因此，我们将几何规划视为线性规划的一个推广或拓展。

半定规划

当 K 为 S_+^k ，即 $k \times k$ 半正定矩阵锥时，相应的锥形式问题称为半定规划（SDP），并具有如下形式

$$\begin{aligned} & \text{minimize} \quad c^T x \\ & \text{subject to} \quad x_1 F_1 + \dots + x_n F_n + G \preceq 0 \\ & \quad Ax = b \end{aligned}$$

其中 G, F_1, \dots, F_n 都是对角阵，那么上式中的 LMI (LMI 表示什么？) 等价于 n 个线性不等式，SDP 退化为线性规划。

标准和不等式形式的半定规划 仿照线性规划的分析, 标准形式的 SDP 具有对变量 $X \in S^n$ 的线性等式约束和 (矩阵) 非负约束:

$$\begin{aligned} & \text{minimize} && \text{Tr}(CX) \\ & \text{subject to} && \text{Tr}(A_i X) = b_i, i = 1, \dots, p \\ & && X \succeq 0 \end{aligned}$$

其中 $C, A_1, \dots, A_p \in S^n$, $\text{Tr}(CX) = \sum_{i,j=1}^n C_{ij} X_{ij}$ 是 S^n 上一般实值线性函数的形式。将这一形式与标准形式的线性规划进行比较, 在线性规划 (LP) 和 SDP 的标准形式中, 我们在变量的 p 个线性等式约束和变量非负约束下极小化变量的线性函数。

如同不等式形式的 LP, 不等式形式的 SDP 不含有等式的约束但是具有一个 LMI:

$$\begin{aligned} & \text{minimize} && C^T x \\ & \text{subject to} && x_1 A_1 + \dots + x_n A_n \preceq B \end{aligned}$$

其优化变量为 $x \in \mathbb{R}^n$, 参数为 $B, A_1, \dots, A_n \in S^k, c \in \mathbb{R}^n$ 。

10.5 阅读材料

Minkowski 被认为第一个对凸集进行了系统的分析并且引入了很多基础性的概念, 例如支撑超平面, 支撑超平面定理, Minkowski 距离函数 (参见习题 3.34), 凸集的极限点等。一些早期的综述可见 Bonnoscn 和 Fcnchel 的, Eggleston 的, Klee 的, 以及 Valentine 的。最近的一些书籍专注于凸集的几何特征, 例如 Lay 以及 Webster. Klee, Fenchel, Tikhomorov 以及 Berger 给出了非常有趣的综述, 讲述了凸性的历史及其在数学方面的应用。与线性规划问题相关联, 对线性不等式及多项式集合已经有了充分的讨论. 关于线性不等式和线性规划的一些里程碑式的文献有: Motzkin, vonNeumann 和 Morgenstern, Kantorovich, Koopmans, 以及 Dantzig. Dantzig 中讨论了线性不等式, 包括直至 1963 年前后的历史调研.

20 世纪 60 年代中在非线性规划的研究中提出了广义不等式 (参见 Luenberger 的以及 Isii 的), 并且被扩展到锥优化问题中。Bcllnmn 和 Fan 的是关于广义线性不等式集合 (关于半正定锥) 的一篇早期文献。对于超平面分离定理证明的推广, 我们推荐读者参看 Rockafdlar 的, 以及 Hiriart-Urruty 和 Lemarechal 的。Dantzig 的包含了 vonNeumann 和 Morgenstern 在给出的择一定理, 关于择一定理的更多文献, 参见第 5 章。一些术语 (包括 Pareto 最优性, 有效制造, 价格 A 的解释) 在 Luenberger 的中进行了详尽的讨论。

凸的几何性质在经典的力矩理论 (Krdn 和 Nudelman, Karlin 和 Studden 的) 中有显著的作用。一个著名的例子是非负多项式与力矩锥的对偶性, 参见习题 2.37。凸分析的标准参考文献是 Rockafellar。其他关于凸函数的书有 Stoer 和 Witzgall, Roberts 和 Varberg, VanTiel, Hiriart-Urruty 和 Lemarechal, Ekeland 和 Temam, Borwein 和 Lewis, Florenzano 和 LeVan, Barvinok, 以及 Bertsekas, Nedic 和 Ozdaglar。很多非线性规划的教材也有一些章节涉及凸函数 (例如, Mangasarian,

Bazaraa, Sherali 和 Shetty, Bertsekas, Polyak, 以及 Peressini, Sullivan 和 Uhl), 很多文献中提到了 Jensen 不等式。在不等式的一般性研究中, Jensen 不等式起到了重要的作用, Hardy, Littlewood 和 Polya 以及 Beckenbach 和 Bellman 中都涉及了不等式的研究。透视函数的概念来源于 Hiriart-Urniy 和 Lemarechal. 一些定义 (相对熵和 Kullback-Leibler 散度), 以及相应的习题, 可以参考 Cover 和 Thomas。早期的一些关于拟凸函数 (以及凸性的其他扩展) 的重要参考文献有 Nikaido, Mangasarian,

Arrow 和 Enthoven, Ponstein, 以及 Luenberger。更全面的此类参考文献可以参照 Bazaraa, Sherali 和 Shetty Prekopa 对对数-凹函数做了一个综述. 在 Barndorff-Nielsen 中提到了 Laplace 变换的对数-凸性. 关于对数-凹函数的积分的结论证明可以参看 Prekopa。广义不等式在最近的关于锥优化的参考文献中广泛使用, 如 Nesterov 和 Nemirovski; Ben-Tal 和 Nemirovski 以及第 4 章最后所列的参考文献。关于广义不等式的凸性在 Luenberger 和 Isii 中亦有涉及. 矩阵单调性和矩阵凸性由 Lowner 提出, Davis, Roberts 和 Varberg 以及 Marshall 和 Olkin 对其进行了详细讨论。例 3.48 中提到的函数 X^p 的凸性或者凹性的相关结论可以参看 Bondar. 另一个简单的例子, 函数 e^X 不是矩阵凸的, 可以参看 Marshall 和 Olkin。

自 20 世纪 40 年代以来, 线性规划已被广泛地研究, 并且是很多极好的书的主题, 包括 Dantzig 的, Luenberger 的, Schrijver 的, Papadimitriou 和 Steiglitz 的, Bortzsimas 和 Tsitsiklis 的, Vanderbei 的以及 Roos、Terlaky 和 Vial 的。Dantzig 和 Schrijver 也给出了线性规划的详细讨论。最近的综述参见 Todd 的。Schaible 给出了分式规划的概述, 其中包含了线性分式问题及其扩展, 例如凸-凹分式问题。例 4.7 中的经济增长模型出现在 von Neumann 的文献中。关于二次规划问题的研究开始于 20 世纪 50 年代 (例如, Frank 和 Wolfe 的, Markowitz 的, Hildreth 的)。其研究的动机是第 148 页讨论的投资组合优化问题 (Markowitz 的和第 147 页讨论的随机损失的线性规划问题 (参见 Freund)。对于二阶锥规划的兴趣要晚一些, 是从 Nesterov 和 Nemirovski 的才开始的。关于 SOCPs 理论和应用的综述由 Alizadeh 和 Goldfarb 的, Ben-Tal 和 Nemirovski 的 (在那里, 问题称为锥二次规划), 以及 Lobo、Vandenberghe、Boyd 和 Lebret 的给出。鲁棒线性规划和广义的鲁棒凸规划, 由 Ben-Tal 和 Nemirovski 的以及 ElGhaoui 和 Lebret 的提出。Goldfarb 和 Iyengar 的讨论了鲁棒 QCQPs 及其在投资优化中的应用。ElGhaoui、Oustry 和 Lebret 的则关注于鲁棒半定规划。几何规划问题自 20 世纪 60 年代起为人所知. 其在工程设计领域的应用首先由 Duffin、Peterson 和 Zener 的以及 Zener 的提出. Peterson 的以及 Ecker 的描述了七十年代取得的进展. 这些文章和书籍包括了应用在工程, 特别是在化学和土木工程中的例子。Fishburn 和 Dunlop 的, Sapatnekar、Rao、

Vaidya 和 Kang 的以及 Hershenson、Boyd 和 Lee 的将几何规划应用于集成电路的设计问题. 关于悬臂梁设计的例子 (第 156 页) 来源于 Vanderplaats. 关于 Perron-Frobenius 特征值的不同性质, Berman 和 Plemmons 在中给出了证明。Nesterov 和 Nemirovski 的引入了锥形式问题作为非线性凸优化的标准问题形式。随后 Ben-Tal 和 Nemirovski 的发展了锥规划方法, 并给出了许多应用。Alizadeh 以及

Nesterov 和 Nemirovski 的首次对半定规划进行了系统的研究，并且指出了其在凸优化领域的广泛应用。20 世纪 90 年代半定规划的持续研究受到多方面应用的激励，如组合优化（Goemana 和 Williamson 的），控制（Boyd、ElGhaoui、Feron 和 Balakrishnan 的）、Scherer、Gahinet 和 Chilali 的，Dullcrud 和 Paganini 的），通信与信号处理（Luo 的，Davidson、Luo、Wong 和 Ma 的）以及其他工程领域。由 Wolkowicz、Saigal 和 Vandenberghe 编著的书以及 Todd 的，Lewis 和 Overton 的，Vandenberghe 和 Boyd 的等文章提供了综述和扩展的文献，关于 SDP 和矩问题的联系，我们在第 163 页给出了一个简单的例子，

而 Bertsimas 和 Sethuraman 的，Nesterov 的及 Lasserre 的对其进行了细致的研究。最速混合 Markov 链问题来自于 Boyd、Diaconis 和 Xiao 的。多准则问题和 Pareto 最优性是经济学的基础工具，参见 Pareto 的，Debreu 的及 Luenberger 的。例 4.9 的结论被称为 Gauss-Markov 定理而为人所知（Kailath、Sayad 和 Hassibi 的）。

10.6 习题

习题 10.1. 下面的集合哪些是凸集？

(a) 平板，即形如 $\{x \in \mathbf{R}^n | \alpha \leq a^T x \leq \beta\}$ 的集合。

(b) 矩形，即形如 $\{x \in \mathbf{R}^n | \alpha_i \leq x_i \leq \beta_i, i = 1, \dots, n\}$ 的集合。当 $n > 2$ 时，矩形有时也称为超矩形。

(c) 楔形，即 $\{x \in \mathbf{R}^n | a_1^T x \leq b_1, a_2^T x \leq b_2\}$ 。

(d) 距离给定点比距离给定集合近的点构成的集合，即

$$\{x | \|x - x_0\|_2 \leq \|x - y\|_2, \forall y \in S\}$$

其中 $S \subseteq \mathbf{R}^n$ 。

习题 10.2. 令 $C \subseteq \mathbf{R}^n$ 为下列二次不等式的解集，

$$C = \{x \in \mathbf{R}^n | x^T A x + b^T x + c \leq 0\}$$

其中 $A \in \mathbf{S}^n, b \in \mathbf{R}^n, c \in \mathbf{R}$ 。

(a) 证明：如果 $A \succeq 0$ ，那么 C 是凸集。

(b) 证明：如果对某些 $\lambda \in \mathbf{R}$ 有 $A + \lambda gg^T \succeq 0$ ，那么 C 和由 $g^T x + h = 0$ （这里 $g \neq 0$ ）定义的超平面的交集是凸集。

以上命题的逆命题是否成立？

习题 10.3. 证明如果 S_1 和 S_2 是 $\mathbf{R}^{m \times n}$ 中的凸集，那么他们的部分和

$$S = \{(x, y_1 + y_2) | x \in \mathbf{R}^m, y_1, y_2 \in \mathbf{R}^n, (x, y_1) \in S_1, (x, y_2) \in S_2\}$$

也是凸的。

习题 10.4. 支撑超平面

- (a) 将闭凸集 $\{x \in \mathbf{R}_+^2 | x_1 x_2 \geq 1\}$ 表示为半空间的交集。
(b) 令 $C = \{x \in \mathbf{R}^n | \|x\|_\infty \leq 1\}$ 表示 \mathbb{R}^n 空间中的单位 l_∞ 范数球，并令 \hat{x} 为 C 的边界上的点。显示地写出集合 C 在 \hat{x} 处的支撑超平面。

习题 10.5. 设 $f : \mathbf{R} \rightarrow \mathbf{R}$ 递增，在其定义域 (a, b) 是凸函数，令 g 表示其反函数，即具有定义域 $(f(a), f(b))$ ，且对所有 $a < x < b$ 满足 $g(f(x)) = x$ 。函数 g 是凸函数还是反函数？为什么？

习题 10.6. 证明 $x^* = (1, 1/2, -1)$ 是如下优化问题的最优解

$$\begin{aligned} &\text{minimize} && (1/2)x^T Px + q^T x + r \\ &\text{subject to} && -1 \leq x_i \leq 1, \quad i = 1, 2, 3 \end{aligned}$$

其中

$$P = \begin{bmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{bmatrix}, \quad q = \begin{bmatrix} -22.0 \\ -14.5 \\ 13.0 \end{bmatrix}, \quad r = 1$$

习题 10.7. 考虑极小化二次函数

$$f_0(\mathbf{x}) = (1/2)\mathbf{x}^T \mathbf{P}\mathbf{x} + \mathbf{q}^T \mathbf{x} + r$$

其中， $\mathbf{P} \in \mathbb{S}_+^n$ (n 阶半正定矩阵)。给出 \mathbf{x} 为 f_0 最小解的重要条件，并说明 \mathbf{x} 何时无解，有唯一解，有多个解。

习题 10.8. 计算 $f(x)$ 的共轭函数，以及共轭函数的定义域。

- $f(x) = -\log x$
- $f(x) = e^x$

习题 10.9. 证明：Gauss 概率密度函数的累积分布函数

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-u^2/2} du$$

是对数-凹函数，即 $\log(\Phi(x))$ 是凹函数。

习题 10.10. 考虑优化问题

$$\begin{aligned} &\text{minimize} && f_0(x_1, x_2) \\ &\text{subject to} && 2x_1 + x_2 \geq 1 \\ & && x_1 + 3x_2 \geq 1 \\ & && x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

给出以下函数最优集和最优值

- (a) $f_0(x_1, x_2) = x_1 + x_2$
- (b) $f_0(x_1, x_2) = -x_1 - x_2$
- (c) $f_0(x_1, x_2) = x_1$
- (d) $f_0(x_1, x_2) = \max\{x_1, x_2\}$
- (e) $f_0(x_1, x_2) = x_1^2 + 9x_2^2$

10.7 参考文献

K.J.Arrow and A.C.Enthoven. Quasi-concave programming. *Econometrica*, 29(4):779-800, 1961.

F.Alizadeh and D.Goldfarb. Second-order cone programming. *Mathematical Programming Series B*, 95:3-51, 2003.

E.F.Beckenbach and R.Bellman. *Inequalities*.Springer, second edition, 1965.

S.Boyd, P.Diaconis, and L.Xiao.Fastest mixing Markov chain on a graph.*SIAM Review*, 46(4):667-689, 2004.

S.Boyd, L.KI Ghaoui, E.Feron, and V.Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.

M.Berger.Convexity.*The American Mathematical Monthly*, 97(8):650-678, 1990.

D.P.Bertsekas.*Nonlinear Programming*.Athena Scientific, second edition, 1999.

D.P.Bortsckas.*Convex Analysis and Optimization*.Athena Scientific, 2003. With A.Nedic and A.E.Ozdaglar.

T.Bonnesen and W.Fenchel.*Theorie der konvexen Korper*.Chelsea Publishing Company, 1948. First published in 1934.

R.Bellman and K.Fan.On systems of linear inequalities in Hermitian matrix variables.In V.L.Klee, editor, *Convexity*, volume VII of *Proceedings of the Symposia in Pure Mathematics* pages 1-11.American Mathematical Society, 1963.

A.Ben-Israel.Linear equations and inequalities on finite dimensional, real or complex vector spaces: A unified theory.*Journal of Mathematical Analysis and Applications* 27:367-389, 1969.

J.M.Borwein and A.S.Lewis.*Convex Analysis and Nonlinear Optimization*.Springer, 2000.

O.Barndorff-Niclsen.*Information and Exponential Families in Statistical Theory*.John Wiley & Sons, 1978.

J.V.Bondar.Comments on and complements to Inequalities: Theory of Majorization and Its Applications.*Linear Algebra and Its Applications* 199:115-129, 1994.

A.Berman and R.J.Plenunons.*Nonnegative Matrices in the Mathematical Sciences*.Society for Industrial and Applied Mathematics, 1994.First published in 1979 by Academic Press.

- M.S.Bazaraa, H.D.Sherali, and C.M.Shetty.Nonlinear Programming.Theory and Algorithms.John Wiley & Sons, second edition, 1993.
- D.Bertsimas and J.N. Tsitsiklis. Introduction to Linear Optimization, Athena Scientific, 1997.
- A.Ben-Tal and A.Nemirovski.Robust convex optimization.Mathematics of Operations Research, 23(4):769-805, 1998.
- A.Ben-Tal and A.Nemirovski.Robust solutions of uncertain linear programs.Operations Research Letters, 25(1):1-13, 1999.
- A.Ben-Tal and A.Nemirovski.Lectures on Modern Convex Optimization.Analysis, Algorithms, and Engineering Applications.Society for Industrial and Applied Mathematics, 2001.
- T.M.Cover and J.A.Thomas.Elements of Information Theory.John Wiley & Sons, 1991.
- G.B.Dantzig.Linear Programming and Extensions.Princeton University Press, 1963.
- C.Davis.Notions generalizing convexity for functions defined on spaces of matrices.In V.L.Klee, editor, Convexity, volume VII of Proceedings of the Symposia in Pure Mathematics. pages 187-201.American Mathematical Society, 1963.
- G.Debreu.Theory of Value: An Axiomatic Analysis of Economic Equilibrium. Yale University Press, 1959.
- T.N.Davidson, Z-Q.Luo, and K.M.Wong.Design of orthogonal pulse shapes for communications via semidefinite programming.IEEE Transactions on Signal Processing, 48(5):1433-1445, 2000.
- G.E.Dullcrud and F.Paganini.A Course in Robust Control Theory.A Convex Approach.Springer, 2000.
- R.J.Duffin, E.L.Peterson, and C.Zener.Geometric Programming.Theory and Applications.John Wiley & Sons, 1967.
- L.El Ghaoui and H.Lebret. Robust solutions to least-squares problems with uncertain data.SIAM Journal of Matrix Analysis and Applications, 18(4):1035- 1064, 1997.
- J.G.Ecker.Geometric programming: Methods, computations and applications.SIAM Review, 22(3):338-362, 1980.
- H.G.Eggleston.Convexity.Cambridge University Press, 1958.
- I.Ekeland and R.Temam.Convex Analysis and Variational Inequalities.Classics in Applied Mathematics.Society for Industrial and Applied Mathematics, 1999.Originally published in 1976.
- J.P.Fishburn and A.E.Dunlop.TILOS: A posynomial programming approach to transistor sizing.In IEEE International Conference on Computer-Aided Design: ICCAD-85.Digest of Technical Papers, pages 326 328.TEEE Computer Society Press, 1985.
- W.Fenchel.Convexity through the ages.In P.M.Gruber and J.M.Wills, editors, Convexity and Its Applications, pages 120-130.Birkhauser Verlag, 1983.

- M.Florenzano and C.Le Van.Finite Dimensional Convexity and Optimization.Number 13 in Studies in Economic Theory.Springer, 2001.
- M.Frank and P.Wolfe.An algorithm for quadratic programming.Naval Research Logistics Quarterly, 3:95-110, 1956.
- R.J.PYeund.The introduction of risk into a programming model.Econometrica,24(3):253-263, 1956.
- D.Goldfarb and G.Iyengar.Robust convex quadratically constrained programs.Mathematical Programming Series B, 97:495-515, 2003.
- D.Goldfarb and G.Iyengar.Robust portfolio selection problems.Mathematics of Operations Research, 28(1):1-38, 2003.
- M.X.Goemans and D.P.Williamson.Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming.Journal of the Association for Computing Machinery, 42(6):1115-1145, 1995.
- M.del Mar Hershenson, S.P.Boyd, and T.H.Lee. Optimal design of a CMOS opamp via geometric programming.IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 20(1):1-21, 2001.
- C.Hildreth.A quadratic programming procedure.Naval Research Logistics Quarterly, 4:79 85, 1957.
- J.-B.Hiriart-Urruty and C.Lemarechal. Convex Analysis and Minimization Algorithms.Springer, 1993.Two volumes.
- K.Isii.Inequalities of the types of ChebyKhev and Cramer-Rao and mathematical programming.Annals of The Institute of Statistical Mathematics, 16:277-293, 1964.
- J.L.W, V, Jensen.Sur les fonctions convexes et les inegalites entre les valeurs moyennes.Acta Mathematica, 30:175-193, 1906.
- L.V.Kantorovich. Mathematical methods of organizing and planning production. Management Science, 6(4):366-422,1960. Translated from Russian. First published in 1939.
- V.L.Klee, editor.Convexity, volume 7 of Proceedings of Symposia in Pure Mathematics.American Mathematical Society, 1963.
- V.Klee.What is a convex set? The American Mathematical Monthly, 78(6):616- 631, 1971.
- T.C.Koopmans, editor.Activity Analysis of Production and Allocation, volume 13 of Cowles Commission far Research in Economics Monographs.John Wiley & Sons, 1951.
- S-Kaxlin and W.J.Studden.Tchebycheff Systems: With Applications in Analysis and Statistics.John Wiley & Sons, 1966.
- T.Kailath, A.H.Sayed.and B.Hassibi.Linear Estimation.Prentice-Hall, 2000.
- J.B.Lasserre.Bounds on measures satisfying moment conditions.The Annals of Applied Probability, 12(3):1114-1137, 2002.
- S.R.Lay.Convex Sets and Their Applications.John Wiley & Sons, 1982.

- A.S.Lewis and M.L.Overton.Eigenvalue optimization.Acta Numerica, 5:149-190, 1996.
- K.Lowner.Uber monotone Matrixfunktionen.Mathematische Zeitschrift,38:177-216, 1934.
- D.G.Luenberger.Microeconomic Theory.McGraw-Hill,1995.
- D.G.Luenberger.Quasi-convex programming.SIAM Journal on Applied Mathematics,16(5),1968.
- D.G.Luenberger.Optimization by Vector Space Methods.John Wiley & Sons,1969.
- D.G.Luenberger.Linear and Nonlinear Programming.Addison-Wesley,second edition,1984.
- Z.-Q.Luo.Applications of convex optimization in signal processing and digital communication.Mathematical Programming Series B, 97:177-207f 2003.
- M.S.Lobo,L.Vandenberghe, S.Boyd, and H.Lebret.Applications of second-order cone progrsun-ming.Linear Algebra and Its Applications,284:193-228,1998.
- O.Mangasarian.Nonlinear Programming.Society for Industrial and Applied Mathematics, 1994.First published in 1969 by McGraw-Hill.
- H.Markowitz.Portfolio selection.The Journal of Finance,7(1):77-91,1952.
- H.Markowitz.The optimization of a quadratic function subject to linear constraints.Naval Research Logistics Quarterly, 3:111-133, 1956.
- W.-K.Ma, T.N.Davidson, K, M.Wong, Z.-Q.Luo, and P.-C.Ching.Quasimaximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous CDMA.IEEE Transactions on Signal Processing, 50:912- 922, 2002.
- A.W.Marshall and I.Olkin.Inequalities: Theory of Majorization and Its Applications.Academic Press, 1979.
- T.Motzkin.Beitrage zur Theorie der linearen Ungleichungen.PhD thesis, University of Basel, 1933.
- Y.Nesterov.Squared functional systems and optimization problems.In J.Prenk, C.Roos, T.Terlaky, and S.Zhang, editors, High Performance Optimization Techniques, pages 405-440.Kluwer, 2000.
- H.Nikaido.On von Neumann's minimax theorem.Pacific Journal of Mathematics, 1954.
- Y.Nesterov and A.Nemirovskii.Interior-Point Polynomial Methods in Convex Programming.Society for Industrial and Applied Mathematics, 1994.
- V.Pareto.Manual of Political Economy.A.M.Kelley Publishers,1971.Translated from the French cn-lition.First published in Italian in 1906.
- E.L.Peterson.Geometric programming.SIAM Remew,18(1):1-51,1976.
- B.T.Polyak.Introduction to Optimization.Optimization Software, 1987.TYanslated from Russian.
- J.Ponstcin.Seven kinds of convexity.SIAM Review,9(1):115-119,1967.
- A.Prekopa.Logarithmic concave measures with application to stochastic programming.Acta Scien-tiarum Mathematicarum, 32:301-315, 1971.
- A.Prekopa.On logarithmic concave measures and functions.Acta Scientiarum Mathematicarum,34:335-343, 1973.

- C.H.Papadimitriou and K.Steiglitz.Combinatorial Opttmtzation.Algorithms and Complexity.Dover Publications, 1998.First published in 1982 by Prentice-Hall.
- A.L.Peressini, F.E.Sullivan, and J.J.Uhl.The Mathematics of Nonlinear Programming.Undergraduate Texts in Mathematics.Springer, 1988.
- R.T.Rockafellar.Convex Analysis.Princeton University Press, 1970.
- C.Roos, T.Terlaky, and J-Ph.Vial.Theory and Algorithms for Linear Optimization.An Interior Point Approach.John Wiley & Sons, 1997.
- A.W.Roberts and D.E.Varberg.Convex Functions.Academic Press,1973.
- S.Schaible.Bibliography in fractional programming.Zeitschrift fur Operations Research, 26:211-241, 1982.
- S.Schaible.Fractional programming.Zeitschrift fur Operations Research, 27:39-54,1983.
- A.Schrijver.Theory of Linear and Integer Programming.John Wiley Sons,1986.
- C.Scherer, P.Gahinet, and M.Chilali.Multiobjective output-feedback control via LMI optimization.IEEE Transactions on Automatic Control, 42(7):896-906,1997.
- S.S.Sapatnekar, V.B.Rao, P.M.Vaidya, and S.-M.Kang.An exact solution to the transistor sizing problem for CMOS circuits using convex optimization.IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,12(11):1621-1634, 1993.
- J.Stoer and C.Witzgall.Convexity and Optimization in Finite Dimensions I.Springer-Vcrlag,1970.
- V.M.Tikhomorov.Convex analysis.In R.V.Gamkrelidze, editor, Analysis II:Convex Analysis and Approximation Theory,,volume 14,pages 1-92.Springer, 1990.
- M.J.Todd.The many facets of linear programming.Mathematical Programming Series B,91:417-436,2002.
- F.A.Valentine.Convex Sets.McGraw-Hill,1964.
- G.N.Vanderplaats.Numerical Optimization Techniques for Engineering Design.McGraw-Hill,1984.
- R.J.Vanderbei.Linear Programming: Foundations and Extensions.Khiwer,1996.
- J.von Neumann.A model of general economic equilibrium.Review of Economic Studies, 13(1):1-9, 1945-46.
- J.von Neumann.Discussion of a maximum problem.In A.H.Taub, editor, John von Neumann.Collected Works, volume VI, pages 89-95.Pergamon Press,1963.Unpublished working paper from 1947.
- J.von Neumann and O.Morgenstern.Theory of Games and Economic Behavior.Princeton University Press,third edition,1953.First published in 1944.
- J.van Tiel.Convex Analysis.An Introductory Text John Wiley & Sons,1984.
- H.Wolkowicz, R.Saigal, and L.Vandenberghe, editors.Handbook of Semidefinite Programming.Kluwer Academic Publishers, 2000.
- R.Webster.Convexity.Oxford University Press,1994.

C.Zener.Engineering Design by Geometric Programming.John Wiley & Sons, 1971.

草稿请勿外传

草稿清勿外传

第十一章 最优性条件和对偶理论

本章介绍拉格朗日对偶函数和拉格朗日对偶问题，把标准形式（可能是非凸）的优化问题转化为对偶问题进行求解；介绍凸优化的最优性条件；介绍数据科学中各种常见的优化问题的对偶性问题。

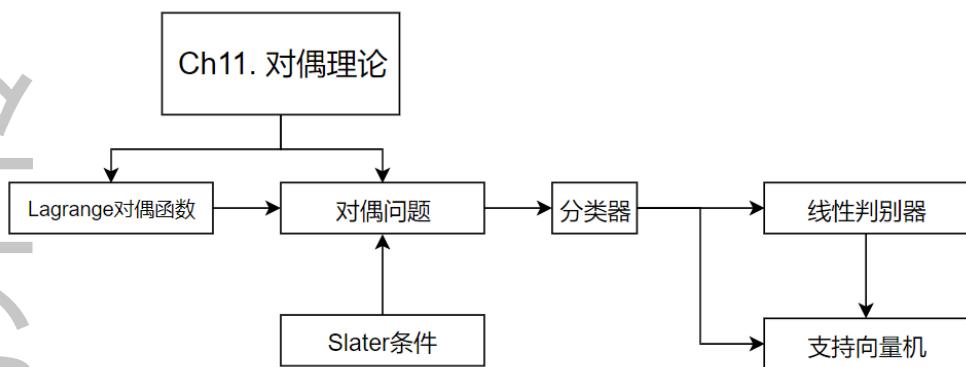


图 11.1: 本章导图

我觉得这张导图里面缺少 KKT 条件，在对偶问题向右箭头到 KKT 条件，然后再箭头指向分类器，这样会比较好

11.1 Lagrange 对偶函数

11.1.1 Lagrange 函数与对偶函数

Lagrange

考虑标准形式的优化问题(1.21):

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq 0, \quad i = 1, \dots, m \\ & && h_j(x) = 0, \quad j = 1, \dots, p \end{aligned} \tag{11.1}$$

其中, 自变量 $x \in \mathbb{R}^n$ 。设问题的定义域 $\mathcal{D} = \bigcap_{i=1}^m \text{dom } f_i \cap \bigcap_{j=1}^p \text{dom } h_j$ 等式约束的下标用 j 表示, 后面都要进行相同标注, 我就不再一一列举了是非空集合, 优化问题的最优值为 p^* 。注意, 这里并没有假设问题(11.1)是凸优化问题。

这部分描述使用 “Algorithms for Convex Optimization” 书的公式 (5.3) 前面一段文字

定义 11.1.1. 定义问题(11.1)的 **Lagrange 函数** $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \mapsto \mathbb{R}$ 为

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x)$$

其中, 定义域为 $\text{dom } L = \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p$ 。 λ_i 是第 i 个不等式约束 $f_i(x) \leq 0$ 的 **Lagrange 乘子**; 类似地, ν_i 是第 i 个等式约束 $h_i(x) = 0$ 对应的 **Lagrange 乘子**。向量 λ 和 ν 称为问题(11.1)的对偶变量或者 **Lagrange 乘子向量**。

Lagrange 对偶函数

定义 11.1.2. 定义 **Lagrange 对偶函数(或对偶函数)** $g : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ 为 **Lagrange 函数** 关于 x 的最小值: 即对 $\lambda \in \mathbb{R}^m, \nu \in \mathbb{R}^p$ 有

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x) \right)$$

如果 **Lagrange 函数** 关于 x 无下界, 则对偶函数取值为 $-\infty$ 。因为对偶函数是一族关于 (λ, ν) 的仿射函数的逐点下确界, 所以即使原问题(11.1)不是凸的, 对偶函数也是凹函数。

最优值的下界

定理 11.1.1. 对偶函数构成了原问题(11.1)最优值 p^* 的下界: 即对于任意 $\lambda \geq 0$ 和 ν 下式成立

$$g(\lambda, \nu) \leq p^* \tag{11.2}$$

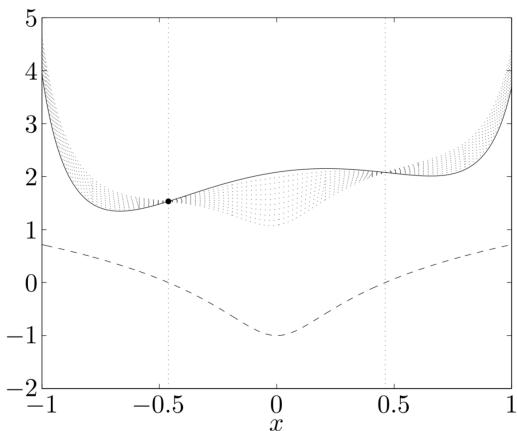


图 11.2: 对偶可行点给出的下界。实线表示目标函数 f_0 , 虚线表示约束函数 f_1 。可行集是区间 $[-0.46, 0.46]$, 如图中两条垂直点线所示。最优点和最优值分别为 $x^* = -0.46, p^* = 1.54$ (在图中用圆点表示)。点线表示一系列 Lagrange 函数 $L(x, \lambda)$, 点线是指的哪些线, 是竖直的点线, 还是水平的点线? 我没太看明白 其中, $\lambda = 0.1, 0.2, \dots, 1.0$ 。每个 Lagrange 函数都有一个极小值(建议把这些极小值点单独标注一下, 或者用不同的颜色, 我现在没看懂这些极小值分别对应的哪些点), 均小于原问题最优目标值 p^* , 这是因为在可行集上 (假设 $\lambda \geq 0$) 有 $L(x, \lambda) \leq f_0(x)$.

证明. 设 \bar{x} 是原问题(11.1)的一个可行点, 即 $f_i(\bar{x}) \leq 0$ 且 $h_i(\bar{x}) = 0$ 。根据假设, $\lambda \geq 0$, 我们有

$$\sum_{i=1}^m \lambda_i f_i(\bar{x}) + \sum_{i=1}^p \nu_i h_i(\bar{x}) \leq 0$$

这是因为左边的第一项非正而第二项为零。根据上述不等式, 有

$$L(\bar{x}, \lambda, \nu) = f_0(\bar{x}) + \sum_{i=1}^m \lambda_i f_i(\bar{x}) + \sum_{i=1}^p \nu_i h_i(\bar{x}) \leq f_0(\bar{x})$$

因此,

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \leq L(\bar{x}, \lambda, \nu) \leq f_0(\bar{x})$$

由于每一个可行点 \bar{x} 都满足 $g(\lambda, \nu) \leq f_0(\bar{x})$, 因此不等式(11.2)成立。 \square

针对某个只包含一个不等式约束的优化问题, 并且满足条件 $x \in R$, 图11.2描述了该问题的最优值 p^* 的下界。

虽然不等式(11.2)成立, 但是当 $g(\lambda, \nu) = -\infty$ 时, 其意义不大。只有当 $\lambda \geq 0$, 且 $(\lambda, \nu) \in \text{dom } g$, 即 $g(\lambda, \nu) > -\infty$ 时, 对偶函数才能给出 p^* 的一个非平凡下界。称满足条件 $\lambda \geq 0$ 和 $(\lambda, \nu) \in \text{dom } g$ 的 (λ, ν) 是对偶可行的。

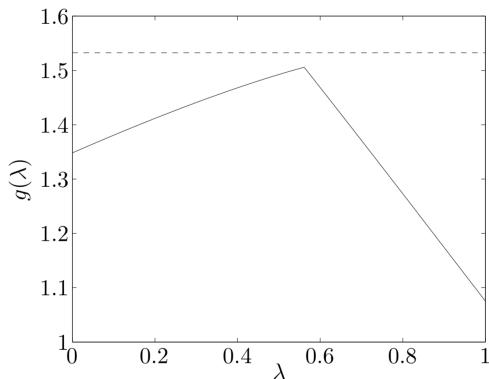


图 11.3: 问题的对偶函数 g 。函数 f_0 和 f_1 都不是凸函数，但是对偶函数是凹函数。水平虚线是原问题的最优函数值 p^* .

11.1.2 常见优化问题目标函数的对偶函数

线性方程组的最小二乘解

考虑问题

$$\begin{aligned} & \text{minimize} && x^T x \\ & \text{subject to} && Ax = b \end{aligned} \tag{11.3}$$

其中 $A \in \mathbb{R}^{p \times n}$ 。这个问题没有不等式约束，只有 p 个等式约束。它的 Lagrange 函数表示为 $L(x, \nu) = x^T x + \nu^T (Ax - b)$ ，其定义域为 $\mathbb{R}^n \times \mathbb{R}^p$ 。它的对偶函数是 $g(\nu) = \inf_x L(x, \nu)$ 。因为 $L(x, \nu)$ 是关于 x 的二次凸函数，可以通过求解如下最优化条件得到函数的最小值，

$$\nabla_x L(x, \nu) = 2x + A^T \nu = 0$$

在点 $x = -(1/2)A^T \nu$ 处，Lagrange 函数达到最小值。此时，对偶函数为

$$g(\nu) = L(-(1/2)A^T \nu, \nu) = -(1/4)\nu^T A A^T \nu - b^T \nu$$

它是一个二次凹函数，定义域为 \mathbb{R}^p 。根据对偶函数是原问题最优值的下界这一性质可知，对任意 $\nu \in \mathbb{R}^p$ ，都有

$$-(1/4)\nu^T A A^T \nu - b^T \nu \leq \inf\{x^T x \mid Ax = b\}$$

标准形式的线性规划

考虑标准形式的线性规划问题

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && Ax = b \\ & && x \geq 0 \end{aligned} \tag{11.4}$$

其中, 不等式约束函数为 $f_i(x) = -x_i \leq 0, i = 1, \dots, n$ 。为了推导 Lagrange 函数, 对 n 个不等式约束引入 Lagrange 乘子 λ_i , 对等式约束引入 Lagrange 乘子 ν_i , 则有

$$\begin{aligned} L(x, \lambda, \nu) &= c^T x - \sum_{i=1}^n \lambda_i x_i + \nu^T (Ax - b) \\ &= -b^T \nu + (c + A^T \nu - \lambda)^T x \end{aligned}$$

对偶函数为

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu) = -b^T \nu + \inf_x (c + A^T \nu - \lambda)^T x$$

可以很容易确定对偶函数的解析表达式, 因为线性函数只有恒为零时才有下界。因此 $c + A^T \nu - \lambda = 0$ 时, $g(\lambda, \nu) = -b^T \nu$, 其余情况下 $g(\lambda, \nu) = -\infty$, 即

$$g(\lambda, \nu) = \begin{cases} -b^T \nu & A^T \nu - \lambda + c = 0 \\ -\infty & \text{其他情况} \end{cases}$$

注意到对偶函数 g 只有在 $\mathbb{R}^m \times \mathbb{R}^p$ 上的一个正常仿射子集上才是有限值。后面我们将会看到这是一种常见的情况。

只有当 λ, ν 满足 $\lambda \geq 0$ 和 $c + A^T \nu - \lambda = 0$ 时, 下界性质(11.2)才是非平凡的, 在此情形下, $-b^T \nu$ 给出了线性规划问题(11.4)最优值的一个下界。

双向划分问题

$$\begin{aligned} &\text{minimize} \quad x^T W x \\ &\text{subject to} \quad x_i^2 = 1, \quad i = 1, \dots, n \end{aligned} \tag{11.5}$$

其中, $W \in S^n$ 。约束条件要求 x_i 的值为 1 或者 -1, 所以原问题等价于寻找这样的向量, 其分量 ± 1 , 并使 $x^T W x$ 最小。可行集是有限的, 包含 2^n 个离散点, 所以此问题本质上可以通过遍历所有可行点来求得最小值。然而, 可行点的数量是指数增长的, 所以, 只有当问题规模较小(比如 $n \leq 30$)时, 遍历法才是可行的。一般而言, 问题(11.5)很难求解。

可以将问题(11.5)看成 n 个元素的集合 ($\{1, \dots, n\}$) 上的双向划分问题, 对任意可行点 x , 其对应的划分为

$$\{1, \dots, n\} = \{i \mid x_i = -1\} \cup \{i \mid x_i = 1\}$$

矩阵系数 W_{ij} 是将 i, j 置于同一分区内的成本; $-W_{ij}$ 可以看成分量 i 和 j 在不同分区内的成本。问题(11.5)中的目标函数是考虑分量间所有配对的成本, 因此问题(11.5)也即寻找使得总成本最小的划分。

下面来推导此问题的对偶函数。Lagrange 函数为

$$\begin{aligned} L(x, \nu) &= x^T W x + \sum_{i=1}^n \nu_i (x_i^2 - 1) \\ &= x^T (W + \text{diag}(\nu)) x - \mathbf{1}^T \nu \end{aligned}$$

对 x 求极小得到 Lagrange 对偶函数

$$\begin{aligned} g(\nu) &= \inf_x x^T (W + \text{diag}(\nu))x - \mathbf{1}^T \nu \\ &= \begin{cases} -\mathbf{1}^T \nu & W + \text{diag}(\nu) \geq 0 \\ -\infty & \text{其他情况} \end{cases} \end{aligned}$$

事实上, 二次函数求下确界时或者是零(如果表达式是半正定的), 或者是 $-\infty$ (如果表达式不是半正定的), 因此对偶函数具有上述形式。

对偶函数构成了问题(11.5) 的最优值的一个下界。例如, 令对偶变量取值为

$$\nu = -\lambda_{\min}(W)\mathbf{1}$$

上述取值是对偶可行的, 这是因为

$$W + \text{diag}(\nu) = W - \lambda_{\min}(W)I \geq 0$$

由此得到了最优值 p^* 的一个下界

$$p^* \geq -\mathbf{1}^T \nu = n\lambda_{\min}(W) \quad (11.6)$$

11.1.3 共轭函数

共轭函数定义及相关性质

回忆函数 $f : R^n \rightarrow R$ 的共轭函数 f^* 为

$$f^*(y) = \sup_{x \in \text{dom } f} (y^T x - f(x))$$

改成定义的形式, 参考 Algorithms for Convex Optimization 的 5.2 节的定义 5.6, 后面把 5.2 节其余的内容都放进来

与 Lagrange 对偶函数的联系

事实上, 共轭函数和 Lagrange 对偶函数紧密相关。下面简单地说明一下它们之间的联系, 考虑问题

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x = 0 \end{aligned}$$

上述问题的 Lagrange 函数为 $L(x, \nu) = f(x) + \nu^T x$, 其对偶函数为

$$g(\nu) = \inf_x (f(x) + \nu^T x) = -\sup_x ((-\nu)^T x - f(x)) = -f^*(-\nu)$$

更一般地(也更有用地), 考虑一个优化问题, 其具有线性不等式以及等式约束,

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && Ax \leq b \\ &&& Cx = d \end{aligned} \quad (11.7)$$

利用函数 f_0 的共轭函数，我们可以将问题(11.7)的对偶函数表述为

$$\begin{aligned} g(\lambda, \nu) &= \inf_x (f_0(x) + \lambda^T(Ax - b) + \nu^T(Cx - d)) \\ &= -b^T\lambda - d^T\nu + \inf_x (f_0(x) + (A^T\lambda + C^T\nu)^T x) \\ &= -b^T\lambda - d^T\nu - f_0^*(-A^T\lambda - C^T\nu) \end{aligned} \quad (11.8)$$

函数 g 的定义域也可以由函数 f_0^* 的定义域得到，

$$\mathbf{dom} g = \{(\lambda, \nu) \mid -A^T\lambda - C^T\nu \in \mathbf{dom} f_0^*\}$$

因此，Lagrange 对偶函数可以用共轭函数来表示。

例 11.1.1. 考虑问题

$$\begin{aligned} &\text{minimize} \quad \|x\| \\ &\text{subject to} \quad Ax = b \end{aligned} \quad (11.9)$$

其中， $\|\cdot\|$ 是任意范数。函数 $f_0 = \|\cdot\|$ 的共轭函数为

$$f_0^*(y) = \begin{cases} 0 & \|y\|_* \leq 1 \\ \infty & \text{其他情况} \end{cases} \quad (11.10)$$

可以看出此函数是对偶范数单位球的示性函数。

利用上面提到的结论(11.8)，可以得到问题(11.9)的对偶函数

$$g(\nu) = -b^T\nu - f_0^*(-A^T\nu) = \begin{cases} -b^T\nu & \|A^T\nu\|_* \leq 1 \\ -\infty & \text{其他情况} \end{cases} \quad (11.11)$$

例 11.1.2. 考虑熵的最大化问题

$$\begin{aligned} &\text{minimize} \quad f_0(x) = \sum_{i=1}^n x_i \log x_i \\ &\text{subject to} \quad Ax \leq b \\ & \quad I^T x = 1 \end{aligned} \quad (11.12)$$

其中， $\mathbf{dom} f_0 = R_{++}^n$ ，关于变量 u 的负熵函数 $u \log u$ 的共轭函数是 e^{u-1} 。因为函数 f_0 是不同变量的负熵函数的和，所以，它的共轭函数可以表示为

$$f_0^*(y) = \sum_{i=1}^n e^{y_i - 1}$$

其定义域为 $\mathbf{dom} f_0^* = R^n$ 。根据结论(11.8)，问题(11.12)的对偶函数为

$$g(\lambda, \nu) = -b^T\lambda - \nu - \sum_{i=1}^n e^{-a_i^T\lambda - \nu - 1} = -b^T\lambda - \nu - e^{-\nu - 1} \sum_{i=1}^n e^{-a_i^T\lambda}$$

其中 a_i 是矩阵 A 的第 i 列向量。

11.2 Lagrange 对偶问题

11.2.1 Lagrange 对偶问题

对于任意一组 (λ, ν) , 其中 $\lambda \geq 0$, Lagrange 对偶函数给出了优化问题(11.1)的最优值 p^* 的一个下界。因此, 我们可以得到和参数 λ, ν 相关的一个下界。一个自然的问题是: 从 Lagrange 函数能够得到的最好下界是什么? 为了研究这个问题, 本节引入了如下优化问题:

定义 11.2.1. 定义问题(11.1)的 **Lagrange 对偶问题**:

$$\begin{aligned} & \text{maximize} \quad g(\lambda, \nu) \\ & \text{subject to} \quad \lambda \geq 0 \end{aligned} \tag{11.13}$$

在本书中, 原始问题(11.1)有时被称为原问题。前面提到的对偶可行的概念, 即描述满足 $\lambda \geq 0$ 和 $g(\lambda, \nu) > -\infty$ 的一组 (λ, ν) , 此时具有意义。它意味着, 这样的一组 (λ, ν) 是对偶问题(11.13)的一个可行解。称解 (λ^*, ν^*) 是对偶最优解或者是**最优 Lagrange 乘子**, 如果它是对偶问题(11.13)的最优解。

Lagrange 对偶问题(11.13)是一个凸优化问题, 这是因为目标函数是凹函数, 且约束集合是凸集, 因此, 对偶问题的凸性和原问题(11.1)是否是凸优化问题无关。

在将原问题转化为对偶问题时, 有时可通过显示表达对偶约束来进行。对偶函数的定义域

$$\mathbf{dom} g = \{(\lambda, \nu) \mid g(\lambda, \nu) > -\infty\}$$

的维数一般都小于 $m + p$ 。事实上, 很多情况下, 我们可以求出 $\mathbf{dom} g$ 的仿射包并将其表示为一系列线性等式约束, 也就是说, 我们可以识别出对偶问题(11.13)的目标函数 g 所“隐含”的等式约束。这样处理之后就可以得到一个等价问题, 在等价问题中, 这些等式约束都被显式地表达为优化问题的约束条件。接下来, 通过以下两个例子来具体说明如何用显示表达对偶约束。

例 11.2.1. 标准形式线性规划

$$\begin{aligned} & \text{minimize} \quad c^T x \\ & \text{subject to} \quad Ax = b \\ & \quad x \geq 0 \end{aligned} \tag{11.14}$$

的 Lagrange 对偶函数为

$$g(\lambda, \nu) = \begin{cases} -b^T \nu & A^T \nu - \lambda + c = 0 \\ -\infty & \text{其他情况} \end{cases}$$

它的对偶问题是在满足约束 $\lambda \geq 0$ 的条件下, 极大化对偶函数 g , 即

$$\begin{aligned} & \text{maximize} \quad g(\lambda, \nu) = \begin{cases} -b^T \nu & A^T \nu - \lambda + c = 0 \\ -\infty & \text{其他情况} \end{cases} \\ & \text{subject to} \quad \lambda \geq 0 \end{aligned} \tag{11.15}$$

当且仅当 $A^T \nu - \lambda + c = 0$ 时, 对偶函数 g 有界。因此, 可以通过将此“隐含”的等式约束“显式”化, 从而得到其等价问题

$$\begin{aligned} & \text{maximize} && -b^T \nu \\ & \text{subject to} && A^T \nu - \lambda + c = 0 \\ & && \lambda \geq 0 \end{aligned} \tag{11.16}$$

进一步地, 这个问题可以表述为

$$\begin{aligned} & \text{maximize} && -b^T \nu \\ & \text{subject to} && A^T \nu + c \geq 0 \end{aligned} \tag{11.17}$$

这是一个不等式形式的线性规划。

注意到这三个问题之间细微的差别。标准形式线性规划(11.14)的 Lagrange 对偶问题是优化问题(11.15), 而这个优化问题等价于问题(11.16)和(11.17)(但形式不同)。称问题(11.16)和(11.17)都是标准形式线性规划(11.14)的 Lagrange 对偶问题。

例 11.2.2. 不等式形式的线性规划问题

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && Ax \leq b \end{aligned} \tag{11.18}$$

的 Lagrange 函数为

$$L(x, \lambda) = c^T x + \lambda^T (Ax - b) = -b^T \lambda + (A^T \lambda + c)^T x$$

所以, 对偶函数为

$$g(\lambda) = \inf_x L(x, \lambda) = -b^T \lambda + \inf_x (A^T \lambda + c)^T x$$

若线性函数的系数不等于 0, 则线性函数的下确界是 $-\infty$ 。因此, 对偶函数可重新表示为

$$g(\lambda) = \begin{cases} -b^T \lambda & A^T \lambda + c = 0 \\ -\infty & \text{其他情况} \end{cases}$$

如果 $\lambda \geq 0$ 且 $A^T \lambda + c = 0$, 那么, 对偶变量 λ 是对偶可行的。

线性规划(11.18)的 Lagrange 对偶问题是对所有的 λ 极大化 g 。和前面一样, 我们可以显式表达对偶可行的条件并作为约束来重新描述对偶问题

$$\begin{aligned} & \text{maximize} && -b^T \lambda \\ & \text{subject to} && A^T \lambda + c = 0 \\ & && \lambda \geq 0 \end{aligned} \tag{11.19}$$

该对偶问题是一个标准形式的线性规划。

通过以上两个例子, 可以发现一个非常有趣的现象, 标准形式线性规划问题和不等式形式线性规划问题与它们的对偶问题之间都存在对称性: 标准形式线性规划的对偶问题是只含有不等式约束的线性规划问题, 反之亦然。此外, 问题(11.19)的 Lagrange 对偶问题就是(等价于)原问题(11.18)。

11.2.2 对偶性质

用 d^* 标记 Lagrange 对偶问题的最优值。 d^* 是通过 Lagrange 函数得到的原问题最优值 p^* 的最好下界。

弱对偶性

定理 11.2.1. 不等式

$$d^* \leq p^* \quad (11.20)$$

成立。即使原问题不是凸优化，上述不等式亦成立。这个性质称为弱对偶性。

根据定理 11.1.1，可直接推导出弱对偶性成立。

即使当 d^* 和 p^* 都趋于无穷时，弱对偶性不等式(11.20)也成立。例如，如果原问题无下界，即 $p^* = -\infty$ ，为了保证弱对偶性，必须有 $d^* = -\infty$ ，即 Lagrange 对偶问题不可行。反过来，若对偶问题无上界，即 $d^* = \infty$ ，为了保证弱对偶性成立，必须有 $p^* = \infty$ ，即原问题不可行。

定义 11.2.2. 差值 $p^* - d^*$ 是原问题的最优值与其通过 Lagrange 对偶函数得到的最好(最大)下界之间的差值。因此，称 $p^* - d^*$ 是原问题的最优对偶间隙。最优对偶间隙总是非负的。

当原问题很难求解时，弱对偶不等式(11.20)给出了原问题最优值的一个下界，这是因为对偶问题总是凸问题，而且在很多情况下都可以进行有效的求解，得到 d^* 。考虑双向划分问题(11.5)，其对偶问题是一个半定规划问题

$$\begin{aligned} & \text{maximize} && -\mathbf{1}^T \nu \\ & \text{subject to} && W + \text{diag}(\nu) \geq 0 \end{aligned} \quad (11.21)$$

其中，变量 $\nu \in R^n$ 。即使当 n 取相对较大的值（例如 $n = 100$ 时），该对偶问题都可以进行有效求解，其最优值给出了双向划分问题最优值的一个下界，而这个下界至少和由 $\lambda_{\min}(W)$ 推导出的下界(11.6)一样好。

强对偶性和 Slater 约束准则

定义 11.2.3. 如果原问题和对偶问题的最优值相等，即等式

$$p^* = d^* \quad (11.22)$$

成立，最优对偶间隙为零，那么，它们满足强对偶性。

对于一般情况，强对偶性不成立。但是，如果原问题(11.1)是凸问题，即表述为如下形式

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq 0, \quad i = 1, \dots, m, \\ & && Ax = b, \end{aligned} \quad (11.23)$$

其中, 函数 f_0, \dots, f_m 是凸函数, 强对偶性通常(但不总是)成立。有很多研究成果给出了强对偶性成立的条件(除了凸性条件以外), 例如, Slater 条件。这些条件称为约束准则。

定义 11.2.4. Slater 条件: 至少存在一点 $x \in \text{relint}\mathcal{D}$ (检查一下 $\text{relint}\mathcal{D}$ 是否在前文中定义过, 如果没有, 这里需要特别说明一下) 使得下式成立

$$f_i(x) < 0, \quad i = 1, \dots, m, \quad h_j = 0, j = 1, \dots, p. \quad (11.24)$$

因为不等式约束严格成立, 所以, 满足上述条件的点是严格可行的。

参考 Algorithms for Convex Optimization 的定理 5.5, 添加强对偶性定理, 并参考 5.3.2 节, 添加定理证明

强对偶性定理说明, 当 Slater 条件成立, 且原问题是凸问题时, 强对偶性成立。当不等式约束函数 f_i 中有一些是仿射函数时, Slater 条件可以进一步改进。

定义 11.2.5. 改进的 Slater 条件: 已知前面的 k 个约束函数 f_1, \dots, f_k 是仿射函数, 存在一点 $x \in \text{relint}\mathcal{D}$, 使得不等式

$$f_i(x) \leq 0, \quad i = 1, \dots, k, \quad f_i(x) < 0, \quad i = k + 1, \dots, m, \quad Ax = b. \quad (11.25)$$

成立。换言之, 仿射不等式不需要严格成立。

注意, 当所有约束条件都是线性等式或不等式且 $\text{dom } f_0$ 是开集时, 改进的 Slater 条件(11.25)就是该优化问题的可行性条件。若 Slater 条件(或是改进的 Slater 条件)满足, 则当 $d^* > -\infty$ 时, 对偶问题能够取得最优值, 即存在一组对偶可行解 (λ^*, ν^*) 使得 $g(\lambda^*, \nu^*) = d^* = p^*$ 。

几何解释

参考 Beyod 书的 5.3.1, 添加关于对偶性的几何解释

11.2.3 常见优化问题的对偶问题

线性方程组的最小二乘解 考虑问题(11.3)

$$\underset{x}{\text{minimize}} \quad x^T x$$

$$\text{subject to} \quad Ax = b$$

其对偶问题为

$$\underset{\nu}{\text{maximize}} \quad -(1/4)\nu^T AA^T \nu - b^T \nu$$

它是一个凹二次函数的无约束极大化问题。

此时, Slater 条件就是原问题的可行性条件。所以, 如果 $b \in \mathcal{R}(A)$, 即 $p^* < \infty$, 就有 $p^* = d^*$, 则强对偶性成立, 即使 $p^* = \infty$ 亦如此。并且当 $p^* = \infty$ 时, $b \notin \mathcal{R}(A)$, 故存在 z 使得 $A^T z = 0$, $b^T z \neq 0$ 。因此, 对偶函数在直线 $\{tz \mid t \in \mathbb{R}\}$ 上无界, 也就是说, 对偶问题最优值无界, $d^* = \infty$ 。

二次约束二次规划的 Lagrange 对偶

考虑约束和目标函数都是二次函数的优化问题 (QCQP)

$$\begin{aligned} & \text{minimize} \quad (1/2)x^T P_0 x + q_0^T x + r_0 \\ & \text{subject to} \quad (1/2)x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \dots, m \end{aligned} \quad (11.26)$$

其中, $P_0 \in S_{++}^n$, $P_i \in S_+^n$, $i = 1, \dots, m$ 。其 Lagrange 函数为

$$L(x, \lambda) = (1/2)x^T P(\lambda)x + q(\lambda)^T x + r(\lambda)$$

其中

$$P(\lambda) = P_0 + \sum_{i=1}^m \lambda_i P_i, \quad q(\lambda) = q_0 + \sum_{i=1}^m \lambda_i q_i, \quad r(\lambda) = r_0 + \sum_{i=1}^m \lambda_i r_i$$

若 $\lambda \geq 0$, 则有 $P(\lambda) > 0$ 以及

$$g(\lambda) = \inf_x L(x, \lambda) = -(1/2)q(\lambda)^T P(\lambda)^{-1}q(\lambda) + r(\lambda)$$

因此, 对偶问题可以表述为

$$\begin{aligned} & \text{maximize} \quad -1/2q(\lambda)^T P(\lambda)^{-1}q(\lambda) + r(\lambda) \\ & \text{subject to} \quad \lambda \geq 0 \end{aligned} \quad (11.27)$$

原问题满足 Slater 条件, 也就是二次不等式约束严格成立, 即存在一点 x , 使得

$$(1/2)x^T P_i x + q_i^T x + r_i < 0, \quad i = 1, \dots, m$$

根据强对偶定理可知, 优化问题(11.27)和(11.26)之间强对偶性成立。

熵的最大化

熵的最大化问题(11.12):

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^n x_i \log x_i \\ & \text{subject to} \quad Ax \leq b \\ & \quad \mathbf{1}^T x = 1 \end{aligned}$$

其定义域为 $\mathcal{D} = R_+^n$ 。前面曾在例 11.1.2 中推导过其 Lagrange 对偶函数, 因此, 对偶问题为

$$\begin{aligned} & \text{maximize} \quad -b^T \lambda - \nu - e^{-\nu-1} \sum_{i=1}^n e^{-a_i^T \lambda} \\ & \text{subject to} \quad \lambda \geq 0 \end{aligned} \quad (11.28)$$

其中, 对偶变量 $\lambda \in R^m$, $\nu \in R$ 。原问题满足 Slater 条件, 即存在一点 $x > 0$, 使得 $Ax \leq b$, 以及 $\mathbf{1}^T x = 1$ 成立, 根据强对偶定理可知, 最优对偶间隙为零。

关于对偶变量 ν 解析地求对偶问题的最大值, 这可以简化对偶问题(11.28)。对于任意固定 λ , 当目标函数对 ν 的偏导等于零时, 即

$$\nu = \log \sum_{i=1}^n e^{-a_i^T \lambda} - 1$$

目标函数取最大值。将 ν 的最优值代入对偶问题，可以得到

$$\begin{aligned} \text{maximize} \quad & -b^T \lambda - \log\left(\sum_{i=1}^n e^{-a_i^T \lambda}\right) \\ \text{subject to} \quad & \lambda \geq 0 \end{aligned}$$

这是一个非负约束的几何规划问题（凸优化问题）。

具有强对偶性的一个非凸二次规划问题

虽然不太常见，但是对于非凸问题，强对偶性有时也会成立。考虑在单位球内极小化非凸二次函数的优化问题

$$\begin{aligned} \text{minimize} \quad & x^T Ax + 2b^T x \\ \text{subject to} \quad & x^T x \leq 1 \end{aligned} \tag{11.29}$$

其中， $A \in S^n$, $A \not\succeq 0$, 且 $b \in R^n$ 。因为 $A \not\succeq 0$, 所以这不是一个凸优化问题。这个问题有时也称为信赖域问题，当在单位球内极小化一个函数的二阶逼近函数时会遇到此问题，此时的单位球即为假设二阶逼近近似有效的区域。

Lagrange 函数为

$$L(x, \lambda) = x^T Ax + 2b^T x + \lambda(x^T x - 1) = x^T(A + \lambda I)x + 2b^T x - \lambda$$

则对偶函数为

$$g(\lambda) = \begin{cases} -b^T(A + \lambda I)^\dagger b - \lambda & A + \lambda I \geq 0, \quad b \in \mathcal{R}(A + \lambda I) \\ -\infty & \text{其他情况} \end{cases}$$

其中，矩阵 $(A + \lambda I)^\dagger$ 是矩阵 $A + \lambda I$ 的伪逆。因此，Lagrange 对偶问题为

$$\begin{aligned} \text{maximize} \quad & -b^T(A + \lambda I)^\dagger b - \lambda \\ \text{subject to} \quad & A + \lambda I \geq 0, \quad b \in \mathcal{R}(A + \lambda I) \end{aligned} \tag{11.30}$$

其中，对偶变量 $\lambda \in R$ 。虽然表达式看起来不明显，这是一个凸优化问题。事实上，对偶问题可以很容易地求解，可以将其写成

$$\begin{aligned} \text{maximize} \quad & -\sum_{i=1}^n (q_i^T b)^2 / (\lambda_i + \lambda) - \lambda \\ \text{subject to} \quad & \lambda \geq -\lambda_{\min}(A) \end{aligned}$$

其中， λ_i 和 q_i 分别是矩阵 A 的特征值和相应的（标准正交）特征向量，我们规定若 $q_i^T b = 0$ ，比值 $(q_i^T b)^2 / 0$ 为零，其他情况该比值为 ∞ 。

尽管原问题(11.29)不是凸问题，此问题的最优对偶间隙始终是零：问题(11.29)和问题(11.30)的最优解总是相同。事实上，存在一个更为一般的结论：如果 Slater 条件成立，对于具有二次目标函数和一个二次不等式约束的优化问题，强对偶性总是成立。

11.3 最优性条件

11.3.1 次优解认证和终止准则

如果能够找到一个对偶可行解 (λ, ν) , 就对原问题的最优值建立了一个下界: $p^* \geq g(\lambda, \nu)$ 。因此, 对偶可行点 (λ, ν) 为表达式 $p^* \geq g(\lambda, \nu)$ 的成立提供了一个证明或认证。强对偶性意味着存在任意好的认证。

对偶可行点可以让我们在不知道 p^* 的确切值的情况下界定给定可行点的次优程度。

定义 11.3.1. 如果 x 是原问题可行解且 (λ, ν) 对偶可行, 并且满足

$$f_0(x) - p^* \leq f_0(x) - g(\lambda, \nu)$$

那么, x 是原问题的 ϵ -次优解, 其中, $\epsilon = f_0(x) - g(\lambda, \nu)$ 。此时, (λ, ν) 是对偶问题的 ϵ -次优解。)

定义原问题和对偶问题目标函数的差值

$$f_0(x) - g(\lambda, \nu)$$

为原问题可行解 x 和对偶可行解 (λ, ν) 之间的对偶间隙。一对可行点 $(x, (\lambda, \nu))$ 将原问题(对偶问题)的最优值限制在一个区间上:

$$p^* \in [g(\lambda, \nu), f_0(x)], \quad d^* \in [g(\lambda, \nu), f_0(x)]$$

区间的长度即为上面定义的对偶间隙。

定义 11.3.2. 如果可行点对 $(x, (\lambda, \nu))$ 的对偶间隙为零, 即 $f_0(x) = g(\lambda, \nu)$, 那么, x 是原问题的最优解, (λ, ν) 是对偶问题的最优解。

此时, 我们可以认为 (λ, ν) 是证明 x 是最优解的一个认证(类似地, 也可以认为 x 是证明 (λ, ν) 对偶最优的一个认证)。

上述现象可以用在优化算法中给出非启发式停止准则。

定义 11.3.3. 设某个算法给出一系列原问题可行解 $x^{(k)}$ 以及对偶问题可行解 $(\lambda^{(k)}, \nu^{(k)})$, $k = 1, 2, \dots$, 给定要求的绝对精度 $\epsilon_{abs} > 0$, 那么停止准则(即终止算法的条件)

$$f_0(x^{(k)}) - g(\lambda^{(k)}, \nu^{(k)}) \leq \epsilon_{abs}$$

保证当算法终止的时候, x^k 是 ϵ_{abs} -次优。

事实上, $(\lambda^{(k)}, \nu^{(k)})$ 为此提供了一个认证。(当然, 只有在强对偶性成立的条件下, 此方法对任意小的 ϵ_{abs} 才都可行。)

给定相对精度 $\epsilon_{rel} > 0$, 可以推导类似的条件保证 ϵ -次优。如果

$$g(\lambda^{(k)}, \nu^{(k)}) > 0, \quad \frac{f_0(x^{(k)}) - g(\lambda^{(k)}, \nu^{(k)})}{g(\lambda^{(k)}, \nu^{(k)})} \leq \epsilon_{rel}$$

成立, 或者

$$f_0(x^{(k)}) < 0, \quad \frac{f_0(x^{(k)}) - g(\lambda^{(k)}, \nu^{(k)})}{-f_0(x^{(k)})} \leq \epsilon_{rel}$$

成立, 那么 $p^* \neq 0$, 且可以保证相对误差

$$\frac{|f_0(x^{(k)}) - p^*|}{|p^*|}$$

小于等于 ϵ_{rel} 。

11.3.2 互补松弛条件

假设原问题和对偶问题的最优值都可以达到且相等 (即强对偶性成立)。令 x^* 是原问题的最优解, (λ^*, ν^*) 是对偶问题的最优解, 这表明

$$\begin{aligned} f_0(x^*) &= g(\lambda^*, \nu^*) \\ &= \inf_x (f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x)) \\ &\leq f_0(x^*) + \sum_{i=1}^m \lambda_i^* f_i(x^*) + \sum_{i=1}^p \nu_i^* h_i(x^*) \\ &\leq f_0(x^*) \end{aligned}$$

第一个等式说明最优对偶间隙为零, 第二个等式是对偶函数的定义, 第三个不等式成立是因为 Lagrange 函数关于 x 的下确界小于等于其在 $x = x^*$ 处的值, 最后一个不等式成立则是因为 $\lambda_i^* \geq 0, f_i(x^*) \leq 0, i = 1, \dots, m$, 以及 $h_i(x^*) = 0, i = 1, \dots, p$ 。因此, 在上面的式子链中, 两个不等式取等号。

可以由此得出一些有意义的结论。例如, 由于第三个不等式变为等式, 因此, $L(x, \lambda^*, \nu^*)$ 关于 x 求极小值是在 x^* 处取得的。(Lagrange 函数 $L(x, \lambda^*, \nu^*)$ 也可以有其他最优点; x^* 只是其中一个最优点)。

另一个重要的结论是

$$\sum_{i=1}^m \lambda_i^* f_i(x^*) = 0$$

事实上, 求和项的每一项都非正, 因此有

$$\lambda_i^* f_i(x^*) = 0, \quad i = 1, \dots, m. \quad (11.31)$$

上述条件称为互补松弛性; 它对任意原问题最优解 x^* 以及对偶问题最优解 (λ^*, ν^*) 都成立 (当强对偶性成立时)。我们可以将互补松弛条件写成

$$\lambda_i^* > 0 \implies f_i(x^*) = 0$$

或者

$$f_i(x^*) < 0 \implies \lambda_i^* = 0$$

这表明, 在最优点处, 除非第 i 个约束起作用, 否则第 i 个最优 Lagrange 乘子取值为零。

11.3.3 KKT 最优性条件

假设函数 $f_0, \dots, f_m, h_1, \dots, h_p$ 可微 (因此定义域是开集), 此时并没有假设这些函数是凸函数。

优化问题的 KKT 条件

定义 11.3.4. 令 x^* 和 (λ^*, ν^*) 分别是原问题和对偶问题的最优解, 其对偶间隙为零。已知 $L(x, \lambda^*, \nu^*)$ 关于 x 求极小值是在点 x^* 处取得的, 故函数在 x^* 处的导数为零, 即

$$\nabla f_0(x^*) + \sum_{i=1}^m \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^p \nu_i^* \nabla h_i(x^*) = 0$$

因此, 就有

$$\begin{aligned} f_i(x^*) &\leq 0, \quad i = 1, \dots, m \\ h_i(x^*) &= 0, \quad i = 1, \dots, p \\ \lambda_i^* &\geq 0, \quad i = 1, \dots, m \\ \lambda_i^* f_i(x^*) &= 0, \quad i = 1, \dots, m \end{aligned} \tag{11.32}$$

$$\nabla f_0(x^*) + \sum_{i=1}^m \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^p \nu_i^* \nabla h_i(x^*) = 0,$$

这些公式被称为 **Karush-Kuhn-Tucker(KKT)** 条件。*(参考 Algorithms for Convex Optimization 书的 5.3 节, 可以将每个条件代表什么意思给说清楚)*

总之, 对于目标函数和约束函数可微的任意优化问题, 如果强对偶性成立, 那么, 原问题和对偶问题的任意一对最优解都必须满足 KKT 条件(11.32)。

(插入 Algorithms for Convex Optimization 书的定理 5.1.2, 结合定理 5.1.2 的部分证明以及本书接下来一段文字的描述 (标绿), 写出这个定理的完整证明) 为了说明这一点, 注意到前面两个条件说明了 \tilde{x} 是原问题的可行解。因为 $\tilde{\lambda}_i \geq 0$, $L(x, \tilde{\lambda}, \tilde{\nu})$ 是 x 的凸函数; 最后一个 KKT 条件说明在 $x = \tilde{x}$ 处, Lagrange 函数的导数为零。因此, $L(x, \tilde{\lambda}, \tilde{\nu})$ 关于 x 求极小在 \tilde{x} 处取得最小值。我们得出结论

$$\begin{aligned} g(\tilde{\lambda}, \tilde{\nu}) &= L(\tilde{x}, \tilde{\lambda}, \tilde{\nu}) \\ &= f_0(\tilde{x}) + \sum_{i=1}^m \tilde{\lambda}_i f_i(\tilde{x}) + \sum_{i=1}^p \tilde{\nu}_i h_i(\tilde{x}) \\ &= f_0(\tilde{x}) \end{aligned}$$

最后一行成立是因为 $h_i(\tilde{x}) = 0$ 以及 $\tilde{\lambda}_i f_i(\tilde{x}) = 0$. 这说明原问题的解 \tilde{x} 和对偶问题的解 $(\tilde{\lambda}, \tilde{\nu})$ 之间的对偶间隙为零, 因此分别是原、对偶问题最优解。总之, 对目标函数和约束函数可微的任意凸优化问题, 任意满足 KKT 条件的点分别是原、对偶最优解, 对偶间隙为零。

KKT 条件在优化领域有着重要的作用。在一些特殊情形下，是可以解析求解 KKT 条件的（因此也可以求解优化问题）。更一般地，很多求解凸优化问题的方法可以认为或理解为求解 KKT 条件的方法。

例 11.3.1. 考虑问题

$$\begin{aligned} & \text{minimize} \quad (1/2)x^T Px + q^T x + r \\ & \text{subject to} \quad Ax = b \end{aligned} \tag{11.33}$$

其中, $P \in S_+^n$ 。此问题的 KKT 条件为

$$Ax^* = b, \quad Px^* + q + A^T v^* = 0$$

我们可以将其写成

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ v^* \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}$$

求解变量 x^* , v^* 的 $m+n$ 个方程，其中变量的维数为 $m+n$ ，可以得到优化问题 (11.33) 的最优原变量和对偶变量。

11.3.4 通过解对偶问题求解原问题

之前我们提到，如果强对偶性成立，且存在一个对偶最优解 (λ^*, ν^*) ，那么任意原问题最优点也是 $L(x, \lambda^*, \nu^*)$ 的最优解。这个性质可以让我们从对偶最优方程去求解原问题最优解。

更精确地，假设强对偶性成立，对偶最优解 (λ^*, ν^*) 已知。假设 $L(x, \lambda^*, \nu^*)$ 的最小值点唯一，即下列问题的解

$$\text{minimize} \quad f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x) \tag{11.34}$$

唯一。对于优化凸问题而言，这是必然会发生（比如说， $L(x, \lambda^*, \nu^*)$ 是关于 x 的严格凸函数）。如果问题(11.34)的解是原问题的可行解，那么，它就是原问题的最优解；反之，如果它不是原问题的可行解，那么，原问题不存在最优点，即原问题的最优解无法达到。当对偶问题比原问题更容易求解时，比如说对偶问题可以解析求解或者有某些特殊的结构更易分析，上述方法很有意义。

例 11.3.2. 考虑熵的最大化问题

$$\text{minimize} \quad f_0(x) = \sum_{i=1}^n x_i \log x_i$$

$$\text{subject to} \quad Ax \leq b$$

$$I^T x = 1$$

其中定义域为 R_{++}^n ，其对偶问题为

$$\text{maximize} \quad -b^T \lambda - \nu - e^{-\nu-1} \sum_{i=1}^n e^{-a_i^T \lambda}$$

$$\text{subject to} \quad \lambda \geq 0$$

假设改进的 *Slater* 条件成立，即存在 $x > 0$ 使得 $Ax \leq b$ 以及 $\mathbf{I}^T x = 1$ ，因此，强对偶性成立，则存在一个对偶最优解 (λ^*, ν^*) 。

设对偶问题已经解出。 (λ^*, ν^*) 处的 *Lagrange* 函数为

$$L(x, \lambda^*, \nu^*) = \sum_{i=1}^n x_i \log x_i + \lambda^{*T} (Ax - b) + \nu^* (\mathbf{I}^T x - 1)$$

它在 D 上严格凸且有下界，因此，有唯一解 x^* ，

$$x_i^* = 1 / \exp(a_i^T \lambda^* + \nu^* + 1), \quad i = 1, \dots, n$$

其中 a_i 是矩阵 A 的列向量。如果 x^* 是原问题的可行解，那么，它必然是原问题(11.12)的最优解；反之，如果 x^* 不是原问题的可行解，那么，就说原问题的最优解不能达到。

例 11.3.3. 在等式约束下极小化可分函数

$$\begin{aligned} &\text{minimize} \quad f_0(x) = \sum_{i=1}^n f_i(x_i) \\ &\text{subject to} \quad a^T x = b \end{aligned}$$

其中 $a \in R^n$, $b \in R$, 函数 $f_i : R \rightarrow R$ 是可微函数，也是严格凸函数。目标函数是可分的，因为它可以表示为关于一系列单变量 x_1, \dots, x_n 的函数求和的形式。假设函数 f_0 的定义域与约束集有交集，即存在一点 $x_0 \in \text{dom } f_0$ ，使得 $a^T x_0 = b$ 。由此可知，该问题存在唯一最优解 x^* 。

该问题的 *Lagrange* 函数为

$$L(x, \nu) = \sum_{i=1}^n f_i(x_i) + \nu(a^T x - b) = -b\nu + \sum_{i=1}^n (f_i(x_i) + \nu a_i x_i)$$

同样是可分函数，因此，对偶函数为

$$\begin{aligned} g(\nu) &= -b\nu + \inf_x \left(\sum_{i=1}^n (f_i(x_i) + \nu a_i x_i) \right) \\ &= -b\nu + \sum_{i=1}^n \inf_{x_i} (f_i(x_i) + \nu a_i x_i) \\ &= -b\nu - \sum_{i=1}^n f_i^*(-\nu a_i) \end{aligned}$$

故对偶问题可表示为

$$\max_{\nu} -b\nu - \sum_{i=1}^n f_i^*(-\nu a_i)$$

其中， $\nu \in R$ 是实变量。

现在假设找到了一个对偶最优解 ν^* 。事实上，有很多简单的方法来求解一个实变量的凸问题，比如说二分法。因为每个函数 f_i 都是严格凸的，所以，函数 $L(x, \nu^*)$ 关于 x 是严格凸的，故具有唯一的最小点 \tilde{x} 。然而，已知 x^* 是 $L(x, \nu^*)$ 的最小点，因此，就有 $\tilde{x} = x^*$ 。综上所述，可以通过求解 $\nabla_x L(x, \nu^*) = 0$ 得到 x^* ，即求解方程组 $f_i'(x_i^*) = -\nu^* a_i$ 。

11.4 数据科学中常见模型的对偶问题

在模式识别问题和分类问题中，给定 \mathbb{R}^n 中的两个点集 $\{x_1, \dots, x_N\}$ 和 $\{y_1, \dots, y_M\}$ ，我们希望（从给定的函数族中）找到一个函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 在第一个集合中为正而在第二个中为负，即

$$f(x_i) > 0, \quad i = 1, \dots, N, \quad f(y_i) < 0, \quad i = 1, \dots, M$$

如果这些不等式成立，我们称 f ，或其 0-水平集 $\{x | f(x) = 0\}$ 分离、分类或判别了两个点集。我们有时也考虑弱分离，在这种情况下，只需要弱不等式成立。用 y_i 来表示负类样本点，不符合平时使用习惯，建议改掉

11.4.1 分类模型：感知机

感知机 1957 年由 Rosenblatt 提出，是神经网络与支持向量机的基础。它是一个二分类模型，其输入为实例的特征向量，输出为实例的类别，取值为 $\{+1, -1\}$ 。感知机本质上是在输入空间（特征空间）中，通过一个分离超平面将实例划分为正类和负类，属于判别模型。感知机学习旨在求出将训练数据进行线性划分的分离超平面，为此，导入基于误分类的损失函数，利用梯度下降法对损失函数进行极小化，求得感知机模型。感知机学习算法简单并且易于实现，分为原始形式和对偶形式。因此，本节重点叙述感知机学习的具体算法，包括原始形式和对偶形式。

感知机模型

定义 11.4.1. 假设输入空间（特征空间）是 $\mathcal{X} \subseteq \mathbb{R}^n$ ，输出空间是 $\mathcal{Y} = \{+1, -1\}$ 。输入 $x \in \mathcal{X}$ 表示实例的特征向量，对应于输入空间（特征空间）的点，输出 $y \in \mathcal{Y}$ 表示实例的类别。输入空间到输出空间的映射函数

$$f(x) = sign(w \cdot x + b) \tag{11.35}$$

被称为感知机，其中， w 和 b 为感知机模型参数， $w \in \mathbb{R}^n$ 叫做权值或权值向量， $b \in \mathbb{R}$ 叫作偏置， $w \cdot x$ 的内积。 $sign$ 是符号函数，即

$$sign(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x \leq 0 \end{cases} \tag{11.36}$$

感知机模型的假设空间是定义在特征空间中的所有线性分类模型或线性分类器，即函数集合 $\{f : f(x) = w \cdot x + b\}$ 。

感知机有如下几种解释：线性方程

$$w \cdot x + b = 0 \tag{11.37}$$

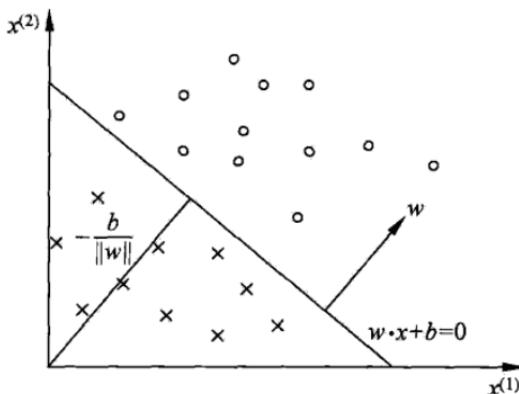


图 11.4: 感知机模型

对应于特征空间 R^n 中的一个超平面 S , 其中 w 是超平面的法向量, b 是超平面的截距。这个超平面将特征空间划分为两个部分。位于两部分的点(特征向量)分别被分为正、负两类。因此, 超平面 S 称为分离超平面, 如图11.4所示。

感知机学习, 由训练数据集(实例的特征向量及类别)

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

其中, $x_i \in \mathcal{X} = R^n$, $y_i \in \mathcal{Y} = \{+1, -1\}$, $i = 1, 2, \dots, N$, 求得感知机模型(11.35), 即求得模型参数 w, b 。感知机预测, 是通过学习得到的感知机模型, 对于新的输入实例给出其对应的输出类别。

感知机学习算法的原始形式

给定一个训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

其中, $x_i \in \mathcal{X} = R^n$, $y_i \in \mathcal{Y} = \{-1, 1\}$, $i = 1, 2, \dots, N$, 求参数 w, b , 使其损失函数最小

$$\min_{w,b} L(w, b) = - \sum_{x_i \in M} y_i (w \cdot x_i + b) \quad (11.38)$$

其中, M 为误分类点的集合。

感知机学习算法是误分类驱动的, 通常采用随机梯度下降法进行求解。首先, 任意选取一个超平面 w_0, b_0 , 然后用梯度下降法不断地极小化目标函数(11.38)。在极小化过程中, 不是一次性使 M 中所有误分类点的梯度下降, 而是一次随机选取一个误分类点使其梯度下降。

假设误分类点集合 M 是固定的，那么损失函数 $L(w, b)$ 的梯度由

$$\nabla_w L(w, b) = - \sum_{x_i \in M} y_i x_i$$

$$\nabla_b L(w, b) = - \sum_{x_i \in M} y_i$$

给出。

随机选取一个误分类点 (x_i, y_i) ，对 w, b 进行更新：

$$\begin{aligned} w &\leftarrow w + \eta y_i x_i \\ b &\leftarrow b + \eta y_i \end{aligned} \tag{11.39}$$

式中， $\eta(0 < \eta \leq 1)$ 是步长，在统计学习中又称为学习率。这样，通过迭代使得期望损失函数 $L(w, b)$ 不断减小，直到为 0。综上所述，得到如下算法：

算法 1：感知机学习算法的原始形式

输入：训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ ，其中 $x_i \in \mathcal{X} = R^n$, $y_i \in \mathcal{Y} = \{-1, +1\}$, $i = 1, 2, \dots, N$ ；学习率 $\eta(0 < \eta \leq 1)$ ；

输出： w, b ；感知机模型 $f(x) = \text{sign}(w \cdot x + b)$ 。

(1) 选取初值 w_0, b_0

(2) 在训练集中选取数据 x_i, y_i

(3) 如果 $y_i(w \cdot x_i + b) \leq 0$

$$w \leftarrow w + \eta y_i x_i$$

$$b \leftarrow b + \eta y_i$$

(4) 转至 (2)，直至训练集中没有误分类点。

这种学习算法直观上有如下解释：当一个实例点被误分类，即位于分离超平面的错误一侧时，则调整 w, b 的值，使分离超平面向该误分类点的一侧移动，以减少该误分类点与超平面间的距离，直至超平面越过该误分类点使其被正确分类。

算法 1 是感知机学习的基本算法，称为原始形式。该算法简单且易于实现。

感知机学习算法的对偶形式

现在考虑感知机学习算法的对偶形式。感知机学习算法的原始形式和对偶形式与下一节支持向量机学习算法的原始形式和对偶形式相对应。

对偶形式的基本想法是，将 w 和 b 表示为实例 x_i 和标记 y_i 的线性组合的形式，通过求解其系数而求得 w 和 b 。为了不失一般性，在算法 1 中假设初始值 w_0, b_0 均为 0。对误分类点 (x_i, y_i) 通过

$$w \leftarrow w + \eta y_i x_i$$

$$b \leftarrow b + \eta y_i$$

逐步修改 w, b , 假设修改 n 次, w, b 关于 (x_i, y_i) 每次修改的增量分别是 $\alpha_i y_i x_i$ 和 $\alpha_i y_i$, 这里 $\alpha_i = \eta$ 。这样, 从学习过程不难看出, 最后学习到的 w, b 可以分别表示为

$$\begin{aligned} w &= \sum_{i=1}^N \alpha_i y_i x_i \\ b &= \sum_{i=1}^N \alpha_i y_i \end{aligned} \quad (11.40)$$

这里, $\alpha_i \geq 0, i = 1, 2, \dots, N$, 当 $\eta = 1$ 时, 表示第 i 个实例点由于误分而进行更新的次数。**(哪个公式表示第 i 个实例点由于误分而进行更新的次数, 这里没有写清楚)** 实例点更新次数越多, 意味着它距离分离超平面越近, 也就越难正确分类。换句话说, 这样的实例对学习结果影响最大。

下面对照原始形式来描述感知机学习算法的对偶形式。

算法 2 (感知机学习算法的对偶形式)

输入: 线性可分的数据集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, 其中 $x_i \in \mathcal{X} = R^n$, $y_i \in \mathcal{Y} = \{-1, +1\}$, $i = 1, 2, \dots, N$; 学习率 $\eta (0 < \eta \leq 1)$;

输出: α, b ; 感知机模型 $f(x) = \text{sign}(\sum_{j=1}^N \alpha_j y_j x_j \cdot x + b)$ 。

其中 $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$ 。

(1) $\alpha \leftarrow 0, b \leftarrow 0$

(2) 在训练集中选取数据 (x_i, y_i)

(3) 如果 $y_i(\sum_{j=1}^N \alpha_j y_j x_j \cdot x_i + b) \leq 0$

$$\alpha_i \leftarrow \alpha_i + \eta$$

$$b \leftarrow b + \eta y_i$$

(4) 转至 (2) 直到没有误分类数据。

对偶形式中训练示例仅以内积的形式出现。为了方便, 可以预先将训练集中实例间的内积计算出来并以矩阵的形式存储, 这个矩阵就是所谓的 Gram 矩阵

$$G = [x_i \cdot x_j]_{N \times N}$$

11.4.2 分类模型：支持向量机

线性判别

在线性判别中, 我们寻找仿射函数 $f(x) = a^T x - b$ 用以区分这些点, 即

$$a^T x_i - b > 0, \quad i = 1, \dots, N, \quad a^T y_i - b < 0, \quad i = 1, \dots, M \quad (\text{用 } y_i \text{ 来表示负类样本点, 不符合平时使用习惯}) \quad (11.41)$$

在几何意义上，我们是在寻找分离两个点集的超平面。因为严格不等式(11.41)对于 a 和 b 是齐次的，所以它们是可行的，当且仅当(关于变量 a 和 b)不严格不等式组

$$a^T x_i - b \geq 1, \quad i = 1, \dots, N, \quad a^T y_i - b \leq -1, \quad i = 1, \dots, M \quad (11.42)$$

是可行的。下图显式了两个点集及线性判别函数的例子。

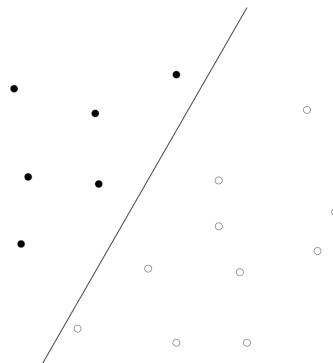


图 11.5: 点 x_1, \dots, x_N 由空心圆圈所示, y_1, \dots, y_M 由实心圆圈所示。两个集合由仿射函数 f 所分类, 其 0-水平集(一条直线) 分离了它们。

鲁棒线性判别

仿射分类函数 $f(x) = a^T x - b$ 是否存在等价于以 a 和 b 为变量的一组线性不等式 f 是否有解。如果两个集合可以被线性判别, 那么, 存在一个可以分离它们的仿射函数的多面体, 于是, 我们可以从中选择某些稳健度量下最优的一个。例如, 可以寻找给出在 x_i 上的(正)值和 y_i 上的(负)值之间最大可能“间距”的函数。为此, 需要对 a 和 b 进行归一化, 因为不这样做的话, 就可以用正常数对 a 和 b 进行伸缩变换而使得数值上的间距任意地大。这样就得到了关于变量 a, b 和 t 的问题

$$\begin{aligned} & \text{maximize} && t \\ & \text{subject to} && a^T x_i - b \geq t, \quad i = 1, \dots, N \\ & && a^T y_i - b \leq -t, \quad i = 1, \dots, M \\ & && \|a\|_2 \leq 1, \end{aligned} \quad (11.43)$$

这个图问题(线性目标、线性和二次不等式)的最优值 t^* 为正, 当且仅当两个点集可以线性分离。在这种情况下, 不等式 $\|a\|_2 \leq 1$ 在最优解处总是紧的, $\|a\|_2 = 1$ 。

我们可以对鲁棒线性判别问题(11.43)给出一个简单的几何解释。如果 $\|a\|_2 = 1$ (在任意最优解处的情况), 那么 $a^T x_i - b$ 是点 x_i 到分离超平面 $H = \{z | a^T z = b\}$ 的 Euclid 距离。类似地,

$b - a^T y_i$ 是点 y_i 到这个超平面的距离，因此，问题(11.43)找到了一个分离两个点集的超平面，并且具有到集合的最大距离。换言之，它找到了分离两个集合的最宽的带。

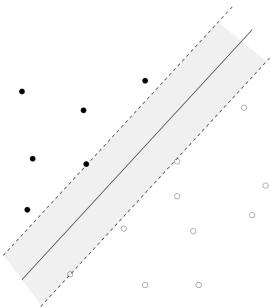


图 11.6: 通过求解鲁棒线性判别问题(11.43)，我们找到了给出两个集合间函数值最大间隙的仿射函数(函数的线性部分有归一化的边界)。几何上，我们寻找分离两个点集的最宽的带。

如上所示例子，最优值 t^* (即带宽的一半)是两个点集的凸包间距离的一半。从鲁棒线性判别问题(11.43)的对偶形式可以更清晰地看到这点。(极小化 $-t$ 问题) 的 Lagrange 为

$$-t + \sum_{i=1}^N \mu_i(t + b - a^T x_i) + \sum_{i=1}^M \nu_i(t - b + a^T y_i) + \lambda(\|a\|_2 - 1)$$

在 b 和 t 上极小化得到条件 $1^T \mu = 1/2, 1^T \nu = 1/2$ 。当它们成立时，我们有

$$\begin{aligned} g(\mu, \nu, \lambda) &= \inf_a (a^T (\sum_{i=1}^M \nu_i y_i - \sum_{i=1}^N \mu_i x_i) + \lambda \|a\|_2 - \lambda) \\ &= \begin{cases} -\lambda & \left\| \sum_{i=1}^M \nu_i y_i - \sum_{i=1}^N \mu_i x_i \right\|_2 \leq \lambda \\ -\infty & \text{其他情况} \end{cases} \end{aligned}$$

于是，这个对偶问题可以写为

$$\begin{aligned} &\text{maximize} && - \left\| \sum_{i=1}^M \nu_i y_i - \sum_{i=1}^N \mu_i x_i \right\|_2 \\ &\text{subject to} && \mu \geq 0, \quad 1^T \mu = 1/2 \\ & && \nu \geq 0, \quad 1^T \nu = 1/2 \end{aligned}$$

我们可以把 $2 \sum_{i=1}^N \mu_i x_i$ 解释为 $\{x_1, \dots, x_N\}$ 的凸包上的一点，而 $2 \sum_{i=1}^M \nu_i y_i$ 是 $\{y_1, \dots, y_M\}$ 的凸包上的一点。对偶目标是极小化这两个点之间的(半)距离，即寻找两个集合的凸包之间的(半)距离。

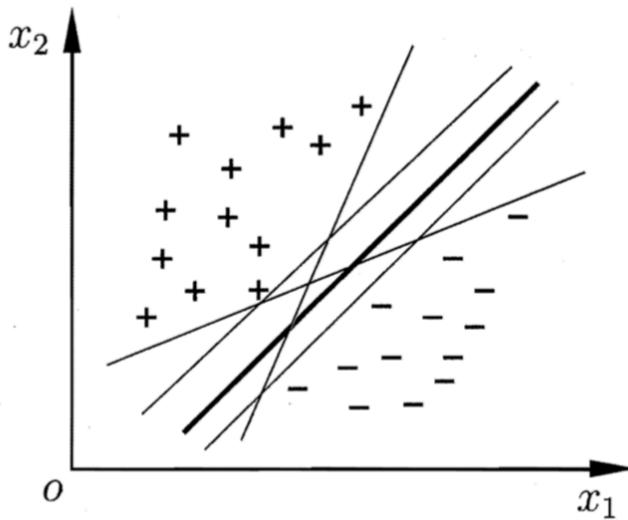


图 11.7: 存在多个划分超平面将两类训练样本分开

给定训练样本集 $D = (x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$, $y_i \in \{-1, +1\}$, 分类学习最基本的想法就是基于训练集 D 在样本空间中找到一个划分超平面, 将不同类别的样本分开。但能将训练样本分开的划分超平面可能有很多, 如图 11.7 所示, 我们应该努力去找到哪一个呢?

直观上看, 应该去找位于两类训练样本“正中间”的划分超平面, 即图 11.7 中的那个, 因为该划分超面对训练样本局部扰动的“容忍”性最好。例如, 由于训练集的局限性或噪声等因素, 训练集外的样本可能比图 11.7 中的训练样本更接近两个类的分隔界, 这将使许多划分超平面出现错误, 而红色的超平面受影响最小。**(图 11.7 里面没有标注红色的超平面)** 换言之, 这个划分超平面所产生的分类结果是最鲁棒的, 对未见示例的泛化能力最强。

在样本空间中, 划分超平面可通过如下线性方程来描述:

$$w^T x + b = 0 \quad (11.44)$$

其中, $w = (w_1, w_2, \dots, w_d)$ 为法向量, 决定了超平面的方向; b 为位移项, 决定了超平面与原点之间的距离。显然, 划分超平面可被法向量 w 和位移 b 确定, 下面我们将其记为 (w, b) 。样本空间中任一点 x 到超平面 (w, b) 的距离可写为

$$r = \frac{|w^T x + b|}{\|w\|} \quad (11.45)$$

假设超平面 (w, b) 能将训练样本正确分类, 即对于 $(x_i, y_i) \in D$, 若 $y_i = +1$, 则有 $w^T x_i + b > 0$; 若 $y_i = -1$, 则有 $w^T x_i + b < 0$ 。令

$$\begin{cases} w^T x_i + b \geq +1, & y_i = +1; \\ w^T x_i + b \leq -1, & y_i = -1. \end{cases} \quad (11.46)$$

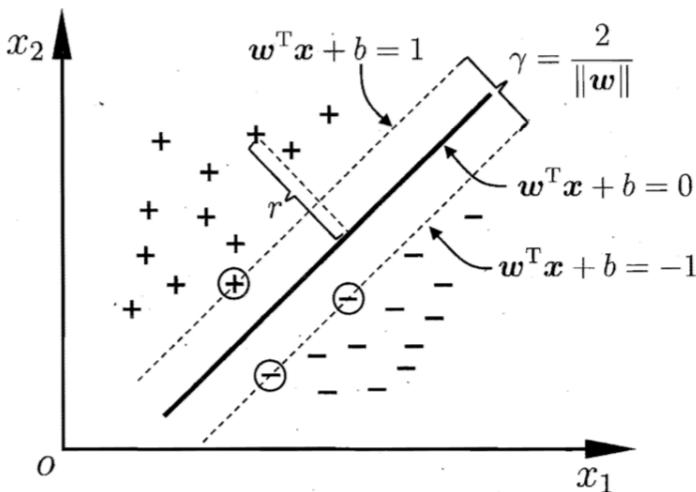


图 11.8: 支持向量与间隔

如图 11.8 所示, 距离超平面最近的几个训练样本点使式(11.46)的等号成立, 它们被称为“支持向量” (support vector), 两个异类支持向量到超平面的距离之和为

$$\gamma = \frac{2}{\|w\|} \quad (11.47)$$

它被称为“间隔” (margin)。想要找到具有“最大间隔” (maximum margin) 的划分超平面, 也就是要找到能满足式中约束的参数 w 和 b , 使得 γ 最大, 即

$$\max_{w,b} \frac{2}{\|w\|} \quad (11.48)$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, \dots, m$$

显然, 为了最大化间隔, 仅需最大化 $\|w\|^{-1}$, 这等价于最小化 $\|w\|^2$ 。于是, 式(11.48)可重写为

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad (11.49)$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, \dots, m$$

这就是支持向量机 (Support Vector Machine, 简称 SVM) 的基本型。

对偶问题

我们希望求解式(11.49)来得到大间隔划分超平面所对应的模型

$$f(x) = w^T x + b \quad (11.50)$$

其中, w 和 b 是模型参数。注意到式(11.49)本身是一个凸二次规划问题, 能直接用现成的优化计算包求解, 但我们可以有更高效的办法。

对式(11.49)使用拉格朗日乘子法可得到其“对偶问题”。具体来说, 对式(11.49)的每条约束添加拉格朗日乘子 $\alpha \geq 0$, 则该问题的拉格朗日函数可写为

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i (w^T x_i + b)) \quad (11.51)$$

其中 $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$ 。令 $L(w, b, \alpha)$ 对 w 和 b 的偏导为零可得

$$w = \sum_{i=1}^m \alpha_i y_i x_i, \quad (11.52)$$

$$0 = \sum_{i=1}^m \alpha_i y_i \quad (11.53)$$

将式(11.52)代入(11.51), 即可将 $L(w, b, \alpha)$ 中的 w 和 b 消去, 再考虑式(11.53)的约束, 就得到式(11.49)的对偶问题

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i = 0, \\ & \alpha_i \geq 0, \quad i = 1, 2, \dots, m \end{aligned} \quad (11.54)$$

解出 α 后, 求出 w 与 b 即可得到模型

$$\begin{aligned} f(x) &= w^T x + b \\ &= \sum_{i=1}^m \alpha_i y_i x_i^T x + b \end{aligned} \quad (11.55)$$

从对偶问题(11.54)得到的解是公式(11.51)中的拉格朗日乘子, 它恰对应着训练样本 $(x_i \square y_i)$ 。注意到式(11.49)中有不等式约束, 因此上述过程需满足 KKT (Karush-Kuhn-Tucker) 条件, 即要求

$$\begin{cases} \alpha_i \geq 0 \\ y_i f(x_i) - 1 \geq 0 \\ \alpha_i (y_i f(x_i) - 1) = 0 \end{cases} \quad (11.56)$$

于是, 对任意训练样本 (x_i, y_i) , 总有 $\alpha_i = 0$ 或 $y_i f(x_i) = 1$ 。若 $\alpha_i = 0$, 则该样本不会在式(11.55)的求和式中出现, 也就不会对 $f(x)$ 产生影响; 若 $\alpha_i > 0$, 则必有 $y_i f(x_i) = 1$, 所对应的样本点位于最大间隔边界上, 是一个支持向量。这也表明了支持向量机的一个重要性质: 训练完成后, 大部分的训练样本都不需保留, 最终模型仅与支持向量有关。

11.5 阅读材料

详细介绍 Lagrange 对偶理论的文献很多，如 Luenberger, Rockafellar, Whittle , Hiriart-Urruty 和 Lemarechal 以及 Bertsckas,Nedic 和 Ozdaglar. Lagrange 对偶这个名字来源于利用 Lagrange 乘子法求解具有等式约束的优化问题，参见 Courant 和 Hilbert。

5.2.5 中矩阵对策的极大极小结论的提出事实上是早于线性规划对偶理论的，von Neuman 和 Morgenstern 通过一个择一定理证明了这个结论。第 219 页提到的关于线性规划的强对偶性的结论是基于 von Neumann 以及 Gale, Kuhn 和 Tucker 的。非凸二次规划问题 (5.32) 的强对偶性是采用信赖域方法求解非线性优化的文献中的一个基本结论 (Nocedal 和 Wright)。这和控制理论中的 S 过程也有关联，见附录 §B.1 中的讨论，将 §5.3.2 中强对偶性的证明扩展至改进的 Slater 条件可以参看文献 Rockafellar。

鞍点性质成立的条件 (5.47) 可以参看文献 Rockafellar 以及 Bertsekas,Nedic 和 Ozdaglar；

KKT 条件得名于 Karush(他在 1939 年未发表的硕士论文中提到了这个结论，文献 Kuhn 对其进行了整理) 以及 Kuhn 和 Tucker.John 也推导了类似的最优性条件。例 5.2 中的注水算法在信息理论以及通信领域得到了应用 (Cover 和 Thomas)。

Farkas 引理由 Farkas 提出。这个引理也是关于线性不等式和等式系统的择一性理论的最为知名的定理，事实上，关于这个定理还有很多不同的变化形式；参见 Mangasarian。Farkas 引理在资产定价 (例 5.10) 中的应用在文献 Bertsimas 和 Tsitsiklis 以及 Ross 中都有涉及。

参考文献 Isii,Luenberger,Berman, 以及 Rockafellar 中都提到了 Lagrange 对偶理论在广义不等式问题中的扩展。在文献 Nesterov 和 Nemirovski 以及 Ben-Tal 和 Nemirovski 中，这种扩展在锥规划问题中予以讨论。广义不等式的强择一定理在参考文献 Ben-Israel, Berman 和 Ben-Israel 以及 Craven 和 Kohila 中被提及。文献 Bellman 和 Fan,Wolkowicz, 以及 Lasserre 给出了 Farkas 引理在线性矩阵不等式中的扩展。

11.6 习题

习题 11.1. 推导共轭函数： $f(x) = \max_{i=1,\dots,n} x_i$ ，定义在 \mathbf{R}^n 上。

习题 11.2. 考虑优化问题

$$\begin{aligned} & \text{minimize } e^{-x} \\ & \text{subject to } x^2/y \leq 0 \end{aligned}$$

优化变量为 x 和 y ，定义域为 $\mathcal{D} = \{(x, y) | y > 0\}$ 。

(a) 证明这是一个凸优化问题，求解最优值。

(b) 给出 Lagrange 对偶问题，求解对偶问题的最优解 λ^* 和最优值 d^* 。给最优对偶间隙。

(c) Slater 条件对此问题是否成立？

习题 11.3. 给定函数 $f(\mathbf{X}) = \text{tr}(\mathbf{X}^{-1})$, 定义域 $\text{dom } f = \mathbf{S}_{++}^n$ 。证明 $f(\mathbf{X})$ 的共轭函数为:

$$f^*(\mathbf{Y}) = -2 \text{tr}(-\mathbf{Y})^{1/2}, \quad \text{dom } f^* = -\mathbf{S}_+^n$$

习题 11.4. 考虑问题

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && f(x) \leq 0 \end{aligned}$$

其中 $c \neq 0$ 。利用共轭 f^* 表述对偶问题。我们不假设函数 f 是凸的, 证明对偶问题是凸的。

习题 11.5. 求解线性规划

$$\begin{aligned} & \text{minimize} && e^T x \\ & \text{subject to} && Gx \leq h \\ & && Ax = b \end{aligned}$$

的对偶函数, 给出对偶问题。

习题 11.6. 证明弱极大极小不等式

$$\sup_{z \in Z} \inf_{w \in W} f(w, z) \leq \inf_{w \in W} \sup_{z \in Z} f(w, z)$$

总是成立。函数 $f : \mathbf{R}^n \times \mathbf{R}^m \rightarrow \mathbf{R}$. $W \subseteq \mathbf{R}^n$, $Z \subseteq \mathbf{R}^m$ 任意。

习题 11.7. 写出下述非线性规划的 KKT 条件并求解

$$(1) \quad \text{maximize} \quad f(x) = (x - 3)^2$$

$$\text{subject to} \quad 1 \leq x \leq 5$$

$$(2) \quad \text{minimize} \quad f(x) = (x - 3)^2$$

$$\text{subject to} \quad 1 \leq x \leq 5$$

习题 11.8. 考虑等式约束的最小二乘问题

$$\text{minimize} \quad \|\mathbf{Ax} - \mathbf{b}\|_2^2$$

$$\text{subject to} \quad \mathbf{Gx} = \mathbf{h}$$

其中 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\text{rank}(\mathbf{A}) = n$, $\mathbf{G} \in \mathbb{R}^{p \times n}$, $\text{rank}(\mathbf{G}) = p$. 给出 KKT 条件, 推导原问题最优解 x^* 以及对偶问题最优解 v^* 的表达式。

习题 11.9. 用 Lagrange 乘子法证明: 矩阵 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 的 2 范数

$$\|\mathbf{A}\|_2 = \max_{\|\mathbf{x}\|_2=1, \mathbf{x} \in \mathbb{R}^n} \|\mathbf{Ax}\|_2$$

的平方是 $\mathbf{A}^\top \mathbf{A}$ 的最大特征值。

习题 11.10. 用 Lagrange 乘子法求欠定方程 $\mathbf{Ax} = \mathbf{b}$ 的最小二范数解, 其中 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m \leq n$, $\text{rank}(\mathbf{A}) = m$

11.7 参考文献

A. Berman and A. Ben-Israel. More on linear inequalities with applications to matrix theory. *Journal of Mathematical Analysis and Applications*, 33:482-496, 1971.

R.Bellman and K.Fan.On systems of linear inequalities in Hermitian matrix variables.In V.L.Klee, editor, *Convexity*, volume VII of *Proceedings of the Symposia in Pure Mathematics* pages 1-11.American Mathematical Society, 1963.

A.Ben-Israel.Linear equations and inequalities on finite dimensional, real or complex vector spaces: A unified theory.*Journal of Mathematical Analysis and Applications* 27:367-389, 1969.

D.Bertsimas and J.N. Tsitsiklis. *Introduction to Linear Optimization*, Athena Scientific, 1997.

A.Ben-Tal and A.Nemirovski.*Lectures on Modern Convex Optimization*.Analysis, Algorithms, and Engineering Applications.Society for Industrial and Applied Mathematics, 2001.

D.P.Bortsckas.*Convex Analysis and Optimization*.Athena Scientific, 2003.With A.Nedic and A.E.Ozdaglar.

A.Berman.*Cones, Matrices and Mathematical Programming*. Springer, 1973.

R.Courant and D.Hilbert.*Method of Mathematical Physics*.Volume 1.Interscience Publishers, 1953.Translated and revised from the 1937 German original.

B.D.Craven and J.J.Koliha.Generalizations of Farkas' theorem.*SIAM Journal on Numerical Analysis*, 8(6), 1977.

T.M.Cover and J.A.Thomas.*Elements of Information Theory*.John Wiley & Sons, 1991.

J.Farkas.Theorie der einfachen Ungleichungen.*Journal fur die Reine und Angewandte Mathematik*, 124:1-27, 1902.

D.Gale, H.W.Kuhn, and A.W.Tucker.Linear programming and the theory of games.In T.C.Koopmans, editor, *Activity Analysis of Production and Allocation*, volume 13 of *Cowles Commission for Research in Economics Monographs*, pages 317-335.John Wiley & Sons, 1951.

J.-B.Hiriart-Urruty and C.Lemarechal. *Convex Analysis and Minimization Algorithms*.Springer, 1993.Two volumes.

K.Iisii.Inequalities of the types of ChebyKhev and Cramer-Rao and mathematical programming.*Annals of The Institute of Statistical Mathematics*, 16:277-293, 1964.

F.John.Extremum problems with inequalities as subsidiary conditions.In J.Moser, editor, *Pritz John, Collected Papers*, pages 543-560.Birkhauser Verlag, 1985.First published in 1948.

H.W.Kuhn and A.W.Tucker.Nonlinear programming.In J.Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability* pages 481-492.University of California Press, 1951.

H.W.Kuhn.Nonlinear programming, A historical view.In R.W.Cottle and C.E.Lemke, editors.*Nonlinear Programming*, volume 9 of *SIAM-AMS Proceedings*, pages 1-26.American Mathematical Society, 1976.

- J.B.Lasserre.A new Farkas lemma for positive semidefinite matrices.IEEE Transactions on Automatic Control 40(6):1131-1133.1995.
- D.G.Luenberger.Optimization by Vector Space Methods.John Wiley & Sons,1969.
- O.Mangasarian.Nonlinear Programming.Society for Industrial and Applied Mathematics, 1994.First published in 1969 by McGraw-Hill.
- Y.Nesterov and A.Nemirovskii.Interior-Point Polynomial Methods in Convex Programming.Society for Industrial and Applied Mathematics, 1994.
- J.Nocedal and S.J.Wright.Numerical Optimization Springer, 1999.
- R.T.Rockafellar.Convex Analysis.Princeton University Press, 1970.
- R.T.Rockafellar.Conjugate Duality and Optimization.Society for Industrial and Applied Mathematics, 1989.First published in 1974.
- S.M.Ross.An Introduction to Mathematical Finance: Options and Other Topics.Cambridge University Press, 1999.
- P.Whittle.Optimization under Constraints.John Wiley & Sons, 1971.
- H.Wolkowicz.Some applications of optimization in matrix theory.Linear Algebra and Its Applications,40:101-118, 1981.
- J.von Neumann.Discussion of a maximum problem.In A.H.Taub, editor, John von Neumann.Collected Works, volume VI, pages 89-95.Pergamon Press,1963.Unpublished working paper from 1947.
- J.von Neumann and O.Morgenstern.Theory of Games and Economic Behavior.Princeton University Press,third edition,1953.First published in 1944.

草稿请勿外传

第十二章 优化算法

对于特定的优化问题，可以找到问题的解析解。比如最小二乘问题。然而，很多问题并没有解析解，或者虽然有解析解，但是利用解析式求解最优值的方式需要极高的运算量。采用迭代的方法逐渐逼近一个最优解是一种可行的方式。优化算法可分为无约束优化算法和约束优化算法，其中无约束优化算法可分为零阶方法（一维搜索），一阶方法和二阶方法；约束优化算法可分为可行方向法和制约函数法。

本章主要介绍无约束优化和约束优化算法的性质和求解方法，除此之外，本章还介绍了深度学习中常用的优化算法，以便读者在实践中使用。

12.1 无约束优化

本节讨论下述无约束优化问题的求解方法

$$\min f(\mathbf{x}) \quad (12.1)$$

其中 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ 的可微函数，现说明该优化问题的极值点（局部极小点）存在的必要条件和充分条件。

定理 12.1.1. [必要条件]

设 $f(\mathbf{x})$ 有一阶连续偏导数，且在点 $\mathbf{x}^* \in \mathbb{R}^n$ 取得局部极值，则必有

$$\nabla f(\mathbf{x}^*) = \mathbf{0} \quad (12.2)$$

其中， \mathbf{x}^* 称为平稳点或驻点。需要指出，极值点必为平稳点，但平稳点不一定是极值点。

定理 12.1.2. [充分条件]

设 $f(\mathbf{x})$ 有二阶连续偏导数， $\mathbf{x}^* \in \mathbb{R}^n$ ，若 $\nabla f(\mathbf{x}^*) = \mathbf{0}$ ，且对于任何非零向量 $\mathbf{z} \in \mathbb{R}^n$ 有

$$\mathbf{z}^T \mathbf{H}(\mathbf{x}^*) \mathbf{z} > 0 \quad (12.3)$$

则 \mathbf{x}^* 为 $f(\mathbf{x})$ 的严格局部极小点。

此处 $\mathbf{H}(\mathbf{x}^*)$ 为 $f(\mathbf{x})$ 在点 \mathbf{x}^* 处的 *Hessian* 矩阵。

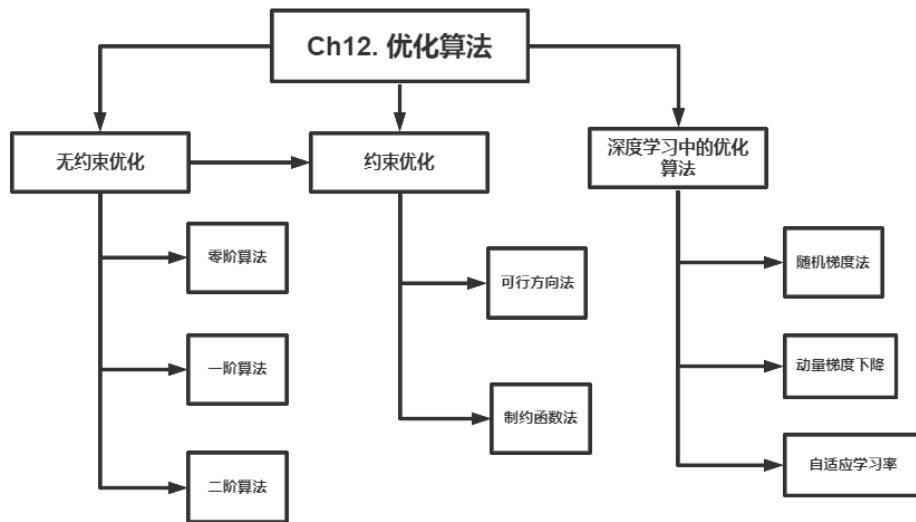


图 12.1: 本章导图

以上两个定理证明从略。需要指出，定理12.1.2中的充分条件式(12.3)并不是必要的。可以举出这样的例子： \mathbf{x}^* 是 $f(\mathbf{x})$ 的极小点，却不满足条件式(12.3)。例如， $f(\mathbf{x}) = \mathbf{x}^4$ ，它的极小点是 $\mathbf{x}^* = 0$ ，但是 $f''(\mathbf{x}^*) = 0$ ，这不满足式(12.3)。

为了求某可微函数的最优解，根据必要条件和充分条件，可如下进行操作：令该函数的梯度等于零，由此求得平稳点；然后用充分条件进行判别，求出所要的解。对某些较为简单的函数，直接求解(12.2)是可行的，但对于一般的 n 元函数 $f(\mathbf{x})$ 来说，方程(12.2)通常是一个非线性方程组，解它相当困难。对于不可微函数，当然谈不上使用这样的方法。为此，常使用迭代法进行求解。

迭代法的基本思想是：为了求函数 $f(\mathbf{x})$ 的最优解，首先给定一个初始估计 $\mathbf{x}^{(0)}$ ，然后按某种规则（即算法）找出比 $\mathbf{x}^{(0)}$ 更好的解 $\mathbf{x}^{(1)}$ （对极小化问题， $f(\mathbf{x}^{(1)}) < f(\mathbf{x}^{(0)})$ ；对极大化问题， $f(\mathbf{x}^{(1)}) > f(\mathbf{x}^{(0)})$ ），再按此种规则找出比 $\mathbf{x}^{(1)}$ 更好的解 $\mathbf{x}^{(2)}, \dots$ 。如此即可得到一个解的序列 $\{\mathbf{x}^{(k)}\}$ 。若这个解序列有极限 \mathbf{x}^* ，即

$$\lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\| = 0$$

则称它收敛于 \mathbf{x}^* 。

若这算法是有效的，那么它所产生的解的序列将收敛于该问题的最优解。不过，由于计算机只能进行有限次迭代，一般说很难得到准确解，而只能得到近似解。当满足所要求的精度时，即可停止迭代。

若由某算法所产生的解的序列 $\{\mathbf{x}^{(k)}\}$ 使目标函数值 $f(\mathbf{x}^{(k)})$ 逐步减少，就称这算法为下降算法。“下降”的要求比较容易实现，它包含了很多种具体算法。显然，求解极小化问题应采用下

降算法。

现假定已迭代到点 $\mathbf{x}^{(k)}$ (见图12.2)，若从 $\mathbf{x}^{(k)}$ 出发沿任何方向移动都不能使目标函数值下降，则 $\mathbf{x}^{(k)}$ 是一局部极小点，迭代停止。若从 $\mathbf{x}^{(k)}$ 出发至少存在一个方向可使目标函数值有所下降，则可选定能使目标函数值下降的某方向 $\mathbf{p}^{(k)}$ ，沿这个方向迈进适当的一步，得到下一个迭代点 $\mathbf{x}^{(k+1)}$ ，并使 $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$ 。这相当于在射线 $\mathbf{x} = \mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)}$ 上选定新点 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)}$ ，其中， $\mathbf{p}^{(k)}$ 称为搜索方向； λ_k 称为步长或步长因子。

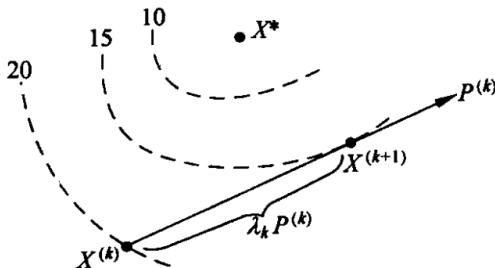


图 12.2

下降迭代算法的步骤可总结如下：

1. 选定某一初始点 $\mathbf{x}^{(0)}$ ，并令 $k := 0$ ；
2. 确定搜索方向 $\mathbf{p}^{(k)}$ ；
3. 从 $\mathbf{x}^{(k)}$ 出发，沿方向 $\mathbf{p}^{(k)}$ 求步长 λ_k ，以产生下一个迭代点 $\mathbf{x}^{(k+1)}$ ；
4. 检查得到的新点 $\mathbf{x}^{(k+1)}$ 是否为极小点或近似极小点。若是，则停止迭代。否则，令 $k := k+1$ ，转回第二步继续进行迭代。

在以上步骤中，选取方向 $\mathbf{p}^{(k)}$ 是最关键的一步，有关各种算法的区分，主要在于确定搜索方向的方法不同。

确定步长 λ_k 可选定不同方法。例如：

- 等于某一常数 (例 $\lambda_k = 1$)；
- 可接受点算法，只要能使目标函数值下降，可任取步长 λ_k ；
- 沿搜索方向使目标函数值下降最多，即沿射线 $\mathbf{x} = \mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)}$ 求目标函数 $f(\mathbf{x})$ 的极小，换言之

$$\lambda_k : f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)})$$

由于改方法是求以 λ 为变量的一元函数 $f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)})$ 的极小点 λ_k ，故称这一过程为（最优）一维搜索或线搜索，这样确定的步长为最佳步长。

一维搜索有个十分重要的性质：在搜索方向上所得最优点处目标函数的梯度和该搜索方向正交。

定理 12.1.3. 设目标函数 $f(\mathbf{x})$ 具有一阶连续偏导数, $\mathbf{x}^{(k+1)}$ 按照下述规则产生

$$\begin{cases} \lambda_k : \min_{\lambda} f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)}) \\ \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)} \end{cases}$$

则有

$$\nabla f(\mathbf{x}^{(k+1)})^T \mathbf{p}^{(k)} = 0 \quad (12.4)$$

证明. 构造函数 $\varphi(\lambda) = f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)})$, 则得

$$\begin{cases} \varphi(\lambda_k) = \min_{\lambda} \varphi(\lambda) \\ \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)} \end{cases}$$

即 λ_k 为 $\varphi(\lambda)$ 的极小点。此外 $\varphi'(\lambda) = \nabla f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)})^T \mathbf{p}^{(k)}$ 。

由 $\varphi'(\lambda)|_{\lambda=\lambda_k} = 0$, 可得

$$\nabla f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)})^T \mathbf{p}^{(k)} = \nabla f(\mathbf{x}^{(k+1)})^T \mathbf{p}^{(k)} = 0$$

□

式(12.4)的几何意义见图12.3。

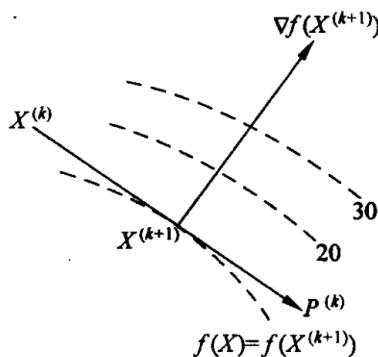


图 12.3

对一个好的算法, 不仅要求它产生的点列能收敛到问题的最优解, 还要求具有较快的收敛速度。设序列 $\{\mathbf{x}^{(k)}\}$ 收敛于 \mathbf{x}^* , 若存在与迭代次数 k 无关的数 $0 < \beta < \infty$ 和 $\alpha \geq 1$, 使 k 从某个 $k_0 > 0$ 开始都有

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| \leq \beta \|\mathbf{x}^{(k)} - \mathbf{x}^*\|^{\alpha} \quad (12.5)$$

成立, 就称 $\{\mathbf{x}^{(k)}\}$ 收敛的阶为 α , 或 $\{\mathbf{x}^{(k)}\}_{\alpha}$ 阶收敛。

草稿勿外传

- 当 $\alpha = 2$ 时，称为二阶收敛；
- 当 $1 < \alpha < 2$ 时，称为超线性收敛；
- 当 $\alpha = 1$ ，且 $0 < \beta < 1$ 时，称为线性收敛或一阶收敛。

一般来讲，线性收敛的收敛速度是比较慢的，二阶收敛是很快的，超线性收敛介于以上两者之间。若一个算法具有超线性或更高的收敛速度，就认为它是一个很好的算法。

因为真正的最优解事先并不知道，为决定什么时候停止计算，只能根据相继两次迭代的结果。常用的终止计算准则有以下几种。

- 根据相继两次迭代的绝对误差

$$\begin{aligned}\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\| &< \varepsilon_1 \\ |f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})| &< \varepsilon_2\end{aligned}$$

- 根据相继两次迭代的相对误差

$$\begin{aligned}\frac{\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}^{(k)}\|} &< \varepsilon_3 \\ \frac{|f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)})|}{|f(\mathbf{x}^{(k)})|} &< \varepsilon_4\end{aligned}$$

- 根据目标函数梯度的模足够小

$$\|\nabla f(\mathbf{x}^{(k)})\| < \varepsilon_5$$

其中， $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5$ 为事先给定的足够小的正数。

根据利用目标函数的信息不同，优化算法可以分为以下几类：

- 零阶方法：又可称为一维搜索，该方法仅需要目标函数的数值，而不需要其梯度信息，常用于梯度和 Hessian 矩阵很难得到的优化问题，例如：没有显式形式、不可微的目标函数；
- 一阶方法：该方法利用目标函数的梯度信息进行优化，适用于不需要很高精度的大数据优化问题，例如：机器学习、深度学习；
- 二阶方法：该方法利用目标函数的 Hessian 矩阵进行优化，适用于需要高精度的优化问题，例如：科学计算。

12.1.1 零阶方法

之前提及，当用迭代法求目标函数的极小点，常常要用到一维搜索，即沿某一已知方向求目标函数的极小点。一维搜索的方法很多，常用的有：

- 试探法（“成功-失败”法，斐波那契法，0.618 法等）；
- 插值法（抛物线插值法，三次插值法等）；
- 微积分中的求根法（切线法，二分法等）。

限于篇幅，以下仅介绍斐波那契法和 0.618 法。

斐波那契法

设 $y = f(t)$ 是区间 $[a, b]$ 上的下单峰函数（图12.4），在此区间内它有唯一极小点 t^* 。若在此区间内任取两点 a_1 和 b_1 , $a_1 < b_1$, 并计算函数值 $f(a_1)$ 和 $f(b_1)$, 可能出现以下两种情况：

- $f(a_1) < f(b_1)$ (图12.4(a)), 这时极小点 t^* 必在区间 $[a, b_1]$ 内;
- $f(a_1) \geq f(b_1)$ (图12.4(b)), 这时极小点 t^* 必在区间 $[a_1, b]$ 内。

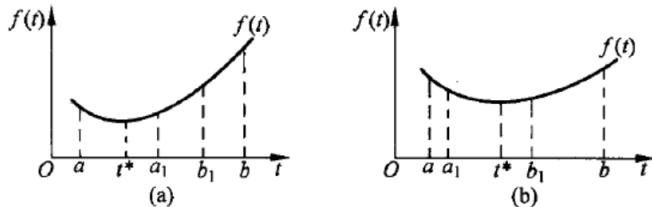


图 12.4

这说明，只要在区间 $[a, b]$ 内取两个不同点，并算出它们的函数值加以比较，就可以把搜索区间 $[a, b]$ 缩小成 $[a, b_1]$ 或 $[a_1, b]$ (缩小后的区间仍需包含极小点)。现在，如果要继续缩小搜索区间 $[a, b_1]$ (或 $[a_1, b]$)，就只需在上述区间内再取一点算出其函数值，并与 $f(a_1)$ 或 $f(b_1)$ 加以比较即可。只要缩小后的区间包含极小点 t^* ，则区间缩小得越小，就越接近于函数的极小点，但计算函数值的次数也就越多。这就说明区间的缩短率和函数值的计算次数有关。现在要问，计算函数值 n 次，能把含有极小点的区间缩小到什么程度呢？或者换一种说法，计算函数值 n 次能把原来多大的区间缩小成长为一个单位的区间呢？

如果用 F_n 表示计算 n 个函数值能缩短为单位区间的最大原区间长度，显然

$$F_0 = F_1 = 1 \quad (12.6)$$

其原因是，只有当原区间长度本来就是一个单位长度时才不必计算函数值；此外，只计算一次函数值无法将区间缩短，故只有区间长度本来就是单位区间才行。

现考虑计算函数值两次的情形，今后我们把计算函数值的点称作试算点或试点。

在区间 $[a, b]$ 内取两个不同点 a_1 和 b_1 (图12.5(a))，计算其函数值以缩短区间，缩短后的区间为 $[a, b_1]$ 或 $[a_1, b]$ 。显然，这两个区间长度之和必大于 $[a, b]$ 的长度，也就是说，计算两次函数值一般无法把长度大于两个单位的区间缩成单位区间。但是，对于长度为两个单位的区间，可以如图12.5(b) 那样选取试点 a_1 和 b_1 ，图中 ε 为任意小的正数，缩短后的区间长度为 $1 + \varepsilon$ 。由于 ε 可任意选取，故缩短后的区间长度接近于一个单位长度。由此可得 $F_2 = 2$ 。

根据同样的分析（见图12.6）可得

$$F_3 = 3, F_4 = 5, F_5 = 8, \dots$$

序列 $\{F_n\}$ 可写成一个递推公式

$$F_n = F_{n-1} + F_{n-2}, \quad n \geq 2 \quad (12.7)$$

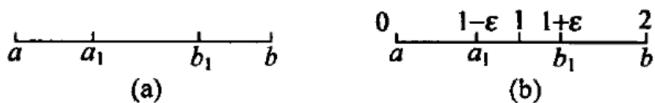


图 12.5

利用式(12.7), 可依次算出各 F_n 的值, 这些 F_n 就是通常所说的斐波那契数。

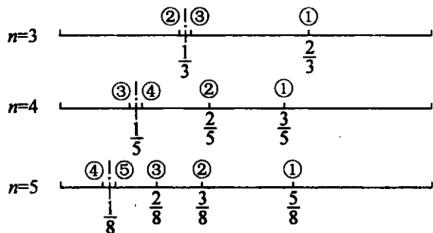


图 12.6

由以上讨论可知, 计算 n 次函数值所能获得的最大缩短率 (缩短后的区间长度与原区间长度之比) 为 $1/F_n$ 。例如 $F_{20} = 10946$, 所以计算 20 个函数值即可把原长度为 L 的区间缩短为

$$\frac{L}{10946} = 0.00009L$$

的区间。现在, 要想计算 n 个函数值, 而把区间 $[a_0, b_0]$ 的长度缩短为原来长度的 δ 倍, 即缩短后的区间长度为

$$b_{n-1} - a_{n-1} \leq (b_0 - a_0)\delta$$

则只要 n 足够大, 能使下式成立即可

$$F_n \geq \frac{1}{\delta} \quad (12.8)$$

其中, δ 为一个正小数, 称为区间缩短的相对精度。有时给出区间缩短的绝对精度 η , 即要求

$$b_{n-1} - a_{n-1} \leq \eta$$

显然, 上述相对精度和绝对精度之间有如下关系

$$\eta = (b_0 - a_0)\delta$$

用这个方法缩短区间的步骤如下:

1. 确定试点的个数 n 。根据相对精度 δ , 即可用式(12.8)算出 F_n , 并确定最小的 n ;

2. 选取前两个试点的位置：由式(12.7)可知第一次缩短时的两个试点位置分别是（见图12.7）：

$$\left\{ \begin{array}{l} t_1 = a_0 + \frac{F_{n-2}}{F_n}(b_0 - a_0) \\ \quad = b_0 + \frac{F_{n-1}}{F_n}(a_0 - b_0) \\ t_1' = a_0 + \frac{F_{n-1}}{F_n}(b_0 - a_0) \end{array} \right. \quad (12.9)$$

它们在区间内的位置是对称的；

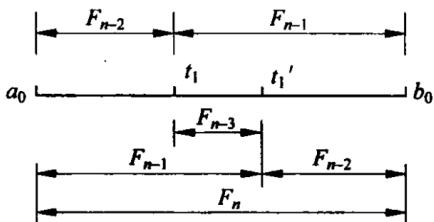


图 12.7

3. 计算函数值 $f(t_1)$ 和 $f(t_1')$ ，并比较它们的大小；若 $f(t_1) < f(t_1')$ ，则取

$$a_1 = a_0 \quad b_1 = t_1' \quad t_2 = t_1$$

并令

$$t_2 = b_1 + \frac{F_{n-2}}{F_{n-1}}(a_1 - b_1)$$

否则，取

$$a_1 = t_1 \quad b_1 = b_0 \quad t_2 = t_1'$$

并令

$$t_2' = a_1 + \frac{F_{n-2}}{F_{n-1}}(b_1 - a_1)$$

4. 计算 $f(t_2)$ 或 $f(t_2')$ （其中的一个已经算出），如第 3 步那样一步步迭代。计算试点的一般公式为

$$\left\{ \begin{array}{l} t_k = b_{k-1} + \frac{F_{n-k}}{F_{n-k+1}}(a_{k-1} - b_{k-1}) \\ t_k' = a_{k-1} + \frac{F_{n-k}}{F_{n-k+1}}(b_{k-1} - a_{k-1}) \end{array} \right. \quad (12.10)$$

其中， $k = 1, 2, \dots, n-1$ ；

5. 当进行至 $k = n-1$ 时

$$t_{n-1} = t_{n-1}' = \frac{1}{2}(a_{n-2} + b_{n-2})$$

这就无法借比较函数值 $f(t_{n-1})$ 和 $f(t'_{n-1})$ 的大小以确定最终区间，为此，取

$$\begin{cases} t_{n-1} = \frac{1}{2}(a_{n-2} + b_{n-2}) \\ t'_{n-1} = a_{n-2} + \left(\frac{1}{2} + \varepsilon\right)(b_{n-2} - a_{n-2}) \end{cases} \quad (12.11)$$

其中 ε 为任意小的数。在 t_{n-1} 和 t'_{n-1} 这两点中，以函数值较小者为近似极小点，相应的函数值为近似极小值，并得最终区间 $[a_{n-2}, t'_{n-1}]$ 或 $[t_{n-1}, b_{n-2}]$ 。

由上述分析可知，斐波那契法使用对称搜索的方法，逐步缩短所考察的区间，它能以尽量少的函数求值次数，达到预定的某一缩短率。

例 12.1.1. 试用斐波那契法求函数 $f(t) = t^2 - t + 2$ 的近似极小点和极小值，要求缩短后的区间长度不大于区间 $[-1, 3]$ 的 0.08 倍。

解. 容易验证，在此区间上函数 $f(t) = t^2 - t + 2$ 为严格凸函数。为了进行比较，我们给出其精确解是： $t^* = 0.5$, $f(t^*) = 1.75$ 。

已知 $\delta = 0.08$, $F_n \geq 1/\delta = 1/0.08 = 12.5$

由斐波那契序列得， $n = 6, a_0 = -1, b_0 = 3$

$$t_1 = b_0 + \frac{F_5}{F_6}(a_0 - b_0) = 3 + \frac{8}{13}(-1 - 3) = 0.538$$

$$t'_1 = a_0 + \frac{F_5}{F_6}(b_0 - a_0) = -1 + \frac{8}{13}(3 - (-1)) = 1.462$$

$$f(t_1) = 0.538^2 - 0.538 + 2 = 1.751$$

$$f(t'_1) = 1.462^2 - 1.462 + 2 = 2.675$$

由于 $f(t_1) < f(t'_1)$, 故取 $a_1 = -1, b_1 = 1.462, t'_2 = 0.538$

$$t_2 = b_1 + \frac{F_4}{F_5}(a_1 - b_1) = 1.462 + \frac{5}{8}(-1 - 1.462) = -0.077$$

$$f(t_2) = (-0.077)^2 - (-0.077) + 2 = 2.083$$

由于 $f(t_2) > f(t'_2) = 1.751$, 故取 $a_2 = -0.077, b_2 = 1.462, t_3 = 0.538$

$$t'_3 = a_2 + \frac{F_3}{F_4}(b_2 - a_2) = -0.077 + \frac{3}{5}(1.462 + 0.077) = 0.846$$

$$f(t'_3) = 0.846^2 - 0.846 + 2 = 1.870$$

由于 $f(t'_3) > f(t_3) = 1.751$, 故取 $a_3 = -0.077, b_3 = 0.846, t'_4 = 0.538$

$$t_4 = b_3 + \frac{F_2}{F_3}(a_3 - b_3) = 0.846 + \frac{2}{3}(-0.077 - 0.846) = 0.231$$

$$f(t_4) = 0.231^2 - 0.231 + 2 = 1.822$$

传外物情高卓

由于 $f(t_4) > f(t'_4) = 1.751$, 故取 $a_4 = 0.231, b_4 = 0.846, t_5 = 0.538$ 。现令 $\varepsilon = 0.01$, 则

$$\begin{aligned} t'_5 &= a_4 + \left(\frac{1}{2} + \varepsilon \right) (b_4 - a_4) \\ &= 0.231 + (0.5 + 0.01)(0.846 - 0.231) = 0.545 \\ f(t'_5) &= 0.545^2 - 0.545 + 2 = 1.752 > f(t_5) = 1.751 \end{aligned}$$

故取 $a_5 = 0.231, b_5 = 0.545$ 。由于 $f(t_5) = 1.751 < f(t'_5) = 1.752$, 所以 t_5 为近似极小点, 近似极小值为 1.751。

缩短后的区间长度为 $0.545 - 0.231 = 0.314, 0.314/4 = 0.0785 < 0.08$ 。

0.618 法

由上节可知, 当用斐波那契法以 n 个试点来缩短某一区间时, 区间长度的第一次缩短率为 F_{n-1}/F_n , 其后各次分别为

$$\frac{F_{n-2}}{F_{n-1}}, \quad \frac{F_{n-3}}{F_{n-2}}, \quad \dots, \quad \frac{F_1}{F_2}$$

现将以上数列分为奇数项 F_{2k-1}/F_{2k} 和偶数项 F_{2k}/F_{2k+1} , 可以证明, 这两个数列收敛于同一个极限。

设当 $k \rightarrow \infty$ 时

$$\frac{F_{2k-1}}{F_{2k}} \rightarrow \lambda \quad \frac{F_{2k}}{F_{2k+1}} \rightarrow \mu$$

由于

$$\frac{F_{2k-1}}{F_{2k}} = \frac{F_{2k-1}}{F_{2k-1} + F_{2k-2}} = \frac{1}{1 + \frac{F_{2k-2}}{F_{2k-1}}}$$

故当 $k \rightarrow \infty$ 时

$$\lim_{k \rightarrow \infty} \frac{F_{2k-1}}{F_{2k}} = \frac{1}{1 + \mu} = \lambda \tag{12.12}$$

同理可证

$$\mu = \frac{1}{1 + \lambda} \tag{12.13}$$

将式(12.12)带入式(12.13)得

$$\mu = \frac{1 + \mu}{2 + \mu}$$

即

$$\mu^2 + \mu - 1 = 0$$

$$\mu = \frac{\sqrt{5} - 1}{2}$$

从而可得

若把式(12.13)带入式(12.12), 则得

$$\lambda^2 + \lambda - 1 = 0$$

故有

$$\lambda = \mu = \frac{\sqrt{5} - 1}{2} = 0.6180339887418948 \quad (12.14)$$

现用不变的区间缩短率 0.618, 代替斐波那契法每次不同的缩短率, 就得到了 0.618 法。这个方法可以看成是斐波那契法的近似, 实现起来比较容易, 效果也相当好, 因而易于为人们所接受。

当用 0.618 法时, 计算 n 个试点的函数值可以把原区间 $[a_0, b_0]$ 连续缩短 $n - 1$ 次, 因为每次的缩短率均为 μ , 故最后的区间长度为

$$(b_0 - a_0)\mu^{n-1}$$

这就是说, 当已知缩短的相对精度为 δ 时, 可用下式计算试点个数 n

$$\mu^{n-1} \leq \delta \quad (12.15)$$

0.618 法是一种等速对称进行试探的方法, 每次的试点均取在区间长度的 0.618 倍和 0.382 倍处。

12.1.2 一阶方法

最速下降法

在求解无约束优化问题中, 梯度法是最为古老但又十分基本的一种数字方法。它的迭代过程简单, 使用方便, 而且又是理解某些其它最优化方法的基础, 所以我们先来说明这一方法。

假定无约束优化问题中的目标函数 $f(\mathbf{x})$ 有一阶连续偏导数, 具有极小点 \mathbf{x}^* 。以 $\mathbf{x}^{(k)}$ 表示极小点的第 k 次近似, 为了求其第 $k + 1$ 次近似点 $\mathbf{x}^{(k+1)}$, 我们在 $\mathbf{x}^{(k)}$ 点沿方向 \mathbf{p}^k 作射线

$$\mathbf{x} = \mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)} \quad (\lambda \geq 0)$$

现将 $f(\mathbf{x})$ 在 $\mathbf{x}^{(k)}$ 点处展成泰勒级数

$$\begin{aligned} f(\mathbf{x}) &= f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)}) \\ &= f(\mathbf{x}^{(k)}) + \lambda \nabla f(\mathbf{x}^{(k)})^\top \mathbf{p}^{(k)} + o(\lambda) \end{aligned}$$

其中

$$\lim_{\lambda \rightarrow 0} \frac{o(\lambda)}{\lambda} = 0$$

对于充分小的 λ , 只要

$$\nabla f(\mathbf{x}^{(k)})^\top \mathbf{p}^{(k)} < 0 \quad (12.16)$$

即可保证 $f(\mathbf{x}^k + \lambda \mathbf{p}^{(k)}) < f(\mathbf{x}^{(k)})$ 。这时若取

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)}$$

就能使目标函数值得到改善。

现考察不同的方向 $\mathbf{p}^{(k)}$ 。假定 $\mathbf{p}^{(k)}$ 的模一定（且不为零），并设 $\nabla f(\mathbf{x}^{(k)}) \neq 0$ （否则， $\mathbf{x}^{(k)}$ 是平稳点），使式(12.15)成立的 $\mathbf{p}^{(k)}$ 有无限多个。为了使目标函数值能得到尽量大的改善，必须寻求使 $\nabla f(\mathbf{x}^{(k)})^T \mathbf{p}^{(k)}$ 取最小值的 $\mathbf{p}^{(k)}$ 。由线性代数学知道

$$\nabla f(\mathbf{x}^{(k)})^T \mathbf{p}^{(k)} = \|\nabla f(\mathbf{x}^{(k)})\| \cdot \|\mathbf{p}^{(k)}\| \cos \theta \quad (12.17)$$

式中 θ 为向量 $\nabla f(\mathbf{x}^{(k)})$ 与 $\mathbf{p}^{(k)}$ 的夹角。当 $\mathbf{p}^{(k)}$ 与 $\nabla f(\mathbf{x}^{(k)})$ 反向时， $\theta = 180^\circ, \cos \theta = -1$ 。这时式(12.16)成立，而且其左端取最小值。我们称方向

$$\mathbf{p}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$$

为负梯度方向，它是使函数值下降最快的方向（在 $\mathbf{x}^{(k)}$ 的某一小范围内）。

为了得到下一个近似极小点，在选定了搜索方向之后，还要确定步长 λ 。当采用可接受点算法时，就是取某一 λ 进行试算，看是否满足不等式

$$f(\mathbf{x}^{(k)} - \lambda \nabla f(\mathbf{x}^{(k)})) < f(\mathbf{x}^{(k)}) \quad (12.18)$$

若上述不等式成立，就可以迭代下去。否则，缩小 λ 使满足不等式式(12.18)。由于采用负梯度方向，满足式(12.18)的 λ 总是存在的。

另一种方法是通过在负梯度方向的一维搜索，来确定使 $f(\mathbf{x})$ 最小的 λ_k ，这种梯度法就是所谓的最速下降法。

现将用梯度法解无约束优化问题的步骤简要总结如下：

1. 给定初始近似点 $\mathbf{x}^{(0)}$ 及精度 $\varepsilon > 0$ ，若 $\|\nabla f(\mathbf{x}^{(0)})\|^2 \leq \varepsilon$ ，则 $f(\mathbf{x}^{(0)})$ 即为近似极小点；
2. 若 $\|\nabla f(\mathbf{x}^{(0)})\|^2 > \varepsilon$ ，求步长 λ_0 ，并计算

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \lambda_0 \nabla f(\mathbf{x}^{(0)})$$

求步长可用一维搜索法、微分法或试算法。若求最佳步长，则应使用前两种方法；

3. 一般地，设已迭代到点 $\mathbf{x}^{(k)}$ ，若 $\|\nabla f(\mathbf{x}^{(k)})\|^2 \leq \varepsilon$ ，则 $\mathbf{x}^{(k)}$ 即为所求的近似解；若 $\|\nabla f(\mathbf{x}^{(k)})\|^2 > \varepsilon$ ，则 $\mathbf{x}^{(k)}$ ，则求步长 λ_k ，并确定下一个近似点

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \lambda_k \nabla f(\mathbf{x}^{(k)}) \quad (12.19)$$

如此继续，直至达到要求的精度为止。

若 $f(\mathbf{x})$ 具有二阶连续偏导数，在 $\mathbf{x}^{(k)}$ 作 $f(\mathbf{x}^{(k)} - \lambda \nabla f(\mathbf{x}^{(k)}))$ 的泰勒展开

$$\begin{aligned} f(\mathbf{x}^{(k)} - \lambda \nabla f(\mathbf{x}^{(k)})) \approx & f(\mathbf{x}^{(k)}) - \nabla f(\mathbf{x}^{(k)})^T \lambda \nabla f(\mathbf{x}^{(k)}) + \\ & \frac{1}{2} \lambda \nabla f(\mathbf{x}^{(k)})^T \mathbf{H}(\mathbf{x}^{(k)}) \lambda \nabla f(\mathbf{x}^{(k)}) \end{aligned}$$

对 λ 求导并令其等于零，则得近似最佳步长

$$\lambda_k = \frac{\nabla f(\mathbf{x}^{(k)})^T \nabla f(\mathbf{x}^{(k)})}{\nabla f(\mathbf{x}^{(k)})^T \mathbf{H}(\mathbf{x}^{(k)}) \nabla f(\mathbf{x}^{(k)})} \quad (12.20)$$

可见近似最佳步长不只与梯度有关，而且与 Hessian 矩阵 \mathbf{H} 也有关系，计算起来比较麻烦。确定步长 λ_k 也可不用式(12.20)，而采用任一种一维搜索法（例如 0.618 法等）。

有时，将搜索方向 $\mathbf{p}^{(k)}$ 的模规格化为 1，在这种情况下

$$\mathbf{p}^{(k)} = \frac{-\nabla f(\mathbf{x}^{(k)})}{\|\nabla f(\mathbf{x}^{(k)})\|} \quad (12.21)$$

同时，式(12.20)变为

$$\lambda_k = \frac{\nabla f(\mathbf{x}^{(k)})^T \nabla f(\mathbf{x}^{(k)}) \|\nabla f(\mathbf{x}^{(k)})\|}{\nabla f(\mathbf{x}^{(k)})^T \mathbf{H}(\mathbf{x}^{(k)}) \nabla f(\mathbf{x}^{(k)})} \quad (12.22)$$

例 12.1.2. 试用梯度法求

$$f(\mathbf{x}) = (x_1 - 1)^2 + (x_2 - 1)^2$$

的极小点，已知 $\varepsilon = 0.1$ 。

解. 取初始点 $\mathbf{x}^{(0)} = (0, 0)^T$

$$\nabla f(\mathbf{x}) = [2(x_1 - 1), 2(x_2 - 1)]^T$$

$$\nabla f(\mathbf{x}^{(0)}) = (-2, -2)^T$$

$$\|\nabla f(\mathbf{x}^{(0)})\|^2 = (-2)^2 + (-2)^2 = 8 > \varepsilon$$

$$\mathbf{H}(\mathbf{x}) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

由式(12.20)

$$\begin{aligned} \lambda_0 &= \frac{\nabla f(\mathbf{x}^{(0)})^T \nabla f(\mathbf{x}^{(0)})}{\nabla f(\mathbf{x}^{(0)})^T \mathbf{H}(\mathbf{x}^{(0)}) \nabla f(\mathbf{x}^{(0)})} \\ &= \frac{(-2, -2) \begin{pmatrix} -2 \\ -2 \end{pmatrix}}{(-2, -2) \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} -2 \\ -2 \end{pmatrix}} = \frac{8}{16} = \frac{1}{2} \end{aligned}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \lambda_0 \nabla f(\mathbf{x}^{(0)}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} -2 \\ -2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\nabla f(\mathbf{x}^{(1)}) = [2(1 - 1), 2(1 - 1)]^T = (0, 0)^T$$

故 $\mathbf{x}^{(1)}$ 即为极小点。

注意，计算步长 λ_0 时也可不用 Hessian 矩阵。由于

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \lambda \nabla f(\mathbf{x}^{(0)}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \lambda \begin{pmatrix} -2 \\ -2 \end{pmatrix} = \begin{pmatrix} 2\lambda \\ 2\lambda \end{pmatrix}$$

代入目标函数可得

$$f(\mathbf{x}^{(1)}) = (2\lambda - 1)^2 + (2\lambda - 1)^2 = 2(2\lambda - 1)^2$$

令

$$df(\mathbf{x}^{(1)})/d\lambda = 0$$

即得所求步长 $\lambda_0 = 1/2$ 。

由这个例子可知，对于目标函数的等值线为圆的问题来说，不管初始点位置取在哪里，负梯度方向总是直指圆心，而圆心即为极值点。这样，只要一次迭代即可达到最优解。

例 12.1.3. 试求 $f(\mathbf{x}) = x_1^2 + 25x_2^2$ 的极小点。

解. 取初始点 $\mathbf{x}^{(0)} = (2, 2)^T$, $f(\mathbf{x}^{(0)}) = 104$ 。本例使用规格化搜索方向法。现先取用固定步长 $\lambda = 1$ ，其迭代过程如表 12.1 所示。

步骤	点	x_1	x_2	$\frac{\partial f(\mathbf{x}^{(k)})}{\partial x_1}$	$\frac{\partial f(\mathbf{x}^{(k)})}{\partial x_2}$	$\ \nabla f(\mathbf{x}^{(k)})\ $
0	$\mathbf{x}^{(0)}$	2	2	4	100	~ 100
1	$\mathbf{x}^{(1)}$	1.96	1.00	3.92	50	50.1
2	$\mathbf{x}^{(2)}$	1.88	0	3.76	0	3.76
3	$\mathbf{x}^{(3)}$	0.88	0	1.76	0	1.76
4	$\mathbf{x}^{(4)}$	-0.12	0	-0.24	0	0.24
5	$\mathbf{x}^{(5)}$	0.88	0			

表 12.1

继续计算下去可以看出， x_1 将来回振荡，难以收敛到极小点 $(0, 0)$ 。为使迭代过程收敛，必须不断减小步长 λ 的值。

采用最佳步长时收敛较快，而且相邻两步的搜索方向互相垂直。下面用最佳步长进行搜索。其迭代过程列于表 12.2 中。

为直观起见，将上述两种迭代过程分别画于图 12.8 和图 12.9 中。

可以证明，当 $f(\mathbf{x})$ 是具有一阶连续偏导数的凸函数时，如果由最速下降法所得的点列 $\{\mathbf{x}^{(k)}\}$ 有界，则必有：(1) 数列 $\{f(\mathbf{x}^{(k)})\}$ 单调下降；(2) 序列 $\{\mathbf{x}^{(k)}\}$ 的极限 \mathbf{x}^* 满足 $\nabla f(\mathbf{x}^*) = \mathbf{0}$ ；(3) \mathbf{x}^* 为全局极小点。

由于负梯度方向的最速下降性，很容易使人们认为负梯度方向是理想的搜索方向，最速下降法是一种理想的极小化方法。必须指出， \mathbf{x} 点处的负梯度方向 $-\nabla f(\mathbf{x})$ ，仅在 \mathbf{x} 点附近才具有这种“最速下降”的性质，而对于整个极小化过程来说，那就是另外一回事了。由例 12.1.2 可知，若目标函数的等值线为一族同心圆（或同心球面），则从任意初始点出发，沿最速下降方向一步

步骤	点	λ_k	x_1	x_2	$\frac{\partial f(x^{(k)})}{\partial x_1}$	$\frac{\partial f(x^{(k)})}{\partial x_2}$	$\ \nabla f(\mathbf{x}^{(k)})\ $
0	$\mathbf{x}^{(0)}$	2.003	2	2	4	100	104
1	$\mathbf{x}^{(1)}$	1.850	1.92	-0.003	3.84	-0.15	3.69
2	$\mathbf{x}^{(2)}$	0.070	0.070	0.070	0.14	3.50	0.13
3	$\mathbf{x}^{(3)}$		0.070	-0.000			

表 12.2

草稿请勿外传

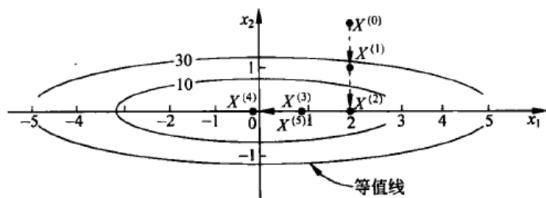


图 12.8

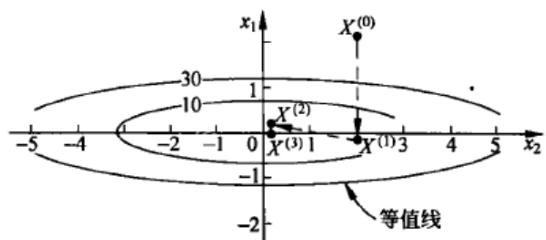


图 12.9

即可达到极小点。但通常的情况并不是这样，例如，一般二元二次凸函数的等值线为一族共心椭圆，当用最速下降法趋近极小点时，其搜索路径呈直角锯齿状（图）。在开头几步，目标函数值下降较快，但接近极小点 x^* 时，收敛速度就不理想了。特别是当目标函数的等值线椭圆比较扁平时，收敛速度就更慢了。因此，在实用中，常将梯度法和其它方法联合起来应用，在前期使用梯度法，而在接近极小点时，则使用收敛较快的其它方法。

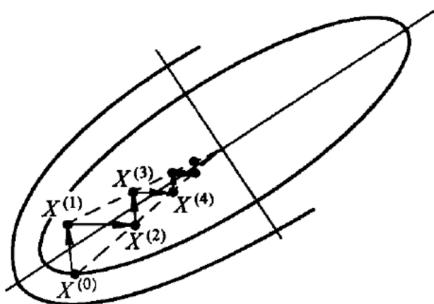


图 12.10

共轭梯度法

(1) 共轭方向

设 \mathbf{x} 和 \mathbf{y} 是 n 维向量，若有

$$\mathbf{x}^T \mathbf{y} = 0$$

就称 \mathbf{x} 和 \mathbf{y} 正交。再设 A 为 $n \times n$ 对称正定阵，如果 \mathbf{x} 和 $A\mathbf{y}$ 正交，即有

$$\mathbf{x}^T A \mathbf{y} = 0 \quad (12.23)$$

则称 \mathbf{x} 和 \mathbf{y} 关于 A 共轭，或 \mathbf{x} 和 \mathbf{y} 为 A 共轭 (A 正交)。

一般地，设 A 为 $n \times n$ 对称正定阵，若非零向量组 $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(n)}$ 满足条件

$$(\mathbf{p}^{(i)})^T A \mathbf{p}^{(j)} = 0 \quad (i \neq j; \quad i, j = 1, 2, \dots, n) \quad (12.24)$$

则称该向量组为 A 共轭。如果 $A = I$ (单位阵)，则上述条件即为通常的正交条件。因此， A 共轭概念实际上是通常正交概念的推广。

定理 12.1.4. 设 A 为 $n \times n$ 对称正定阵， $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(n)}$ 为 A 共轭的非零向量，则这一组向量线性无关。

证明：设向量 $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(n)}$ 之间存在如下线性关系

$$\alpha_1 \mathbf{p}^{(1)} + \alpha_2 \mathbf{p}^{(2)} + \dots + \alpha_n \mathbf{p}^{(n)} = 0$$

对 $i = 1, 2, \dots, n$, 用 $(\mathbf{p}^{(i)})^T \mathbf{A}$ 左乘上式得

$$\alpha_i (\mathbf{p}^{(i)})^T \mathbf{A} \mathbf{p}^{(i)} = 0$$

但 $\mathbf{p}^{(i)} \neq 0, \mathbf{A}$ 为正定, 即

$$(\mathbf{p}^{(i)})^T \mathbf{A} \mathbf{p}^{(i)} > 0$$

故必有

$$\alpha_i = 0, \quad i = 1, 2, \dots, n$$

从而 $\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(n)}$ 线性无关。 \square

无约束优化问题的一个特殊情形是

$$\min f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{B}^T \mathbf{x} + c \quad (12.25)$$

式中 \mathbf{A} 为 $n \times n$ 对称正定阵; \mathbf{x}, \mathbf{B} 为 n 维向量; c 为常数。问题式(12.25)称为正定二次函数极小问题, 它在最优化问题中起到极其重要的作用。

定理 12.1.5. 设向量 $\mathbf{p}^{(i)}, i = 0, 1, 2, \dots, n - 1$, 为 \mathbf{A} 共轭, 则从任一点 $\mathbf{x}^{(0)}$ 出发, 相继以 $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n-1)}$ 为搜索方向的下述算法

$$\begin{cases} \min_{\lambda} f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)}) = f(\mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)}) \\ \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)} \end{cases}$$

经 n 次一维搜索收敛于问题式(12.25)的极小点 \mathbf{x}^* 。

证明. 由式(12.25)

$$\nabla f(\mathbf{x}) = \mathbf{A} \mathbf{x} + \mathbf{B}$$

设相继各次搜索得到的近似解分别为 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}$, 则

$$\begin{aligned} \nabla f(\mathbf{x}^{(k)}) &= \mathbf{A} \mathbf{x}^{(k)} + \mathbf{B} \\ \nabla f(\mathbf{x}^{(k+1)}) &= \mathbf{A} \mathbf{x}^{(k+1)} + \mathbf{B} = \mathbf{A}(\mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)}) + \mathbf{B} \\ &= \nabla f(\mathbf{x}^{(k)}) + \lambda_k \mathbf{A} \mathbf{p}^{(k)} \end{aligned}$$

假定 $\nabla f(\mathbf{x}^{(k)}) \neq 0, k = 0, 1, 2, \dots, n - 1$, 则有

$$\begin{aligned} \nabla f(\mathbf{x}^{(n)}) &= \nabla f(\mathbf{x}^{(n-1)}) + \lambda_{n-1} \mathbf{A} \mathbf{p}^{n-1} = \dots \\ &= \nabla f(\mathbf{x}^{k+1}) + \lambda_{k+1} \mathbf{A} \mathbf{p}^{k+1} + \lambda_{k+2} \mathbf{A} \mathbf{p}^{k+2} + \dots + \\ &\quad \lambda_{n-1} \mathbf{A} \mathbf{p}^{(n-1)} \end{aligned}$$

由于在进行一维搜索时, 为确定最佳步长 λ_k , 令

$$\frac{df(\mathbf{x}^{(k+1)})}{d\lambda} = \frac{df[\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)}]}{d\lambda} = \nabla f(\mathbf{x}^{(k+1)})^T \mathbf{p}^{(k)} = 0 \quad (12.26)$$

故对 $k = 0, 1, 2, \dots, n - 1$ 有

$$(\mathbf{p}^{(k)})^T \nabla f(\mathbf{x}^{(n)}) = (\mathbf{p}^{(k)})^T \nabla f(\mathbf{x}^{(k+1)}) + \lambda_{k+1} (\mathbf{p}^{(k)})^T \mathbf{A} \mathbf{p}^{(k+1)} + \dots + \lambda_{n-1} (\mathbf{p}^{(k)})^T \mathbf{A} \mathbf{p}^{n-1} = 0$$

这就是 $\nabla f(\mathbf{x}^{(n)})$ 和 n 个线性无关的向量 $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n-1)}$ (它们为 \mathbf{A} 共轭) 正交, 从而必有

$$\nabla f(\mathbf{x}^{(n)}) = 0$$

即 \mathbf{x}^n 为 $f(\mathbf{x})$ 的极小点 \mathbf{x}^* 。

□

下面我们就二维正定二次函数的情况加以说明, 以便对上述定理有个直观认识。

二维正定二次函数的等值线, 在极小点附近可用一族共心椭圆来代表 (图)。大家知道, 过椭圆族中心 \mathbf{x}^* 引任意直线, 必与诸椭圆相交, 各交点处的切线互相平行。如果在两个互相平行的方向上进行最优一维搜索, 则可得 $f(\mathbf{x})$ 在此方向上的极小点 $\mathbf{x}^{(1)}$ 和 $\bar{\mathbf{x}}^{(1)}$, 此两点必为椭圆族中某椭圆与该平行直线的切点, 而且联结 $\mathbf{x}^{(1)}$ 和 $\bar{\mathbf{x}}^{(1)}$ 的直线必通过椭圆族的中心 \mathbf{x}^* 。

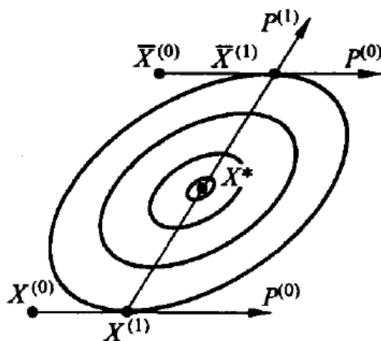


图 12.11

现从任一点 $\mathbf{x}^{(0)}$ 出发, 沿射线 $\mathbf{p}^{(0)}$ 作一维搜索, 则可得问题式(12.25)的目标函数 $f(\mathbf{x})$ 在射线 $\mathbf{x}^{(0)} + \lambda \mathbf{p}^{(0)}$ 上的极小点

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda_0 \mathbf{p}^{(0)}$$

其中, λ_0 满足

$$\nabla f(\mathbf{x}^{(1)})^T \mathbf{p}^{(0)} = 0$$

同样, 从另一点 $\bar{\mathbf{x}}^{(0)}$ 出发也沿 $\mathbf{p}^{(0)}$ 方向作一维搜索, 可得式(12.25)中 $f(\mathbf{x})$ 在射线 $\bar{\mathbf{x}}^{(0)} + \lambda \mathbf{p}^{(0)}$ 上的极小点 $\bar{\mathbf{x}}^{(1)} = \bar{\mathbf{x}}^{(0)} + \lambda_0 \mathbf{p}^{(0)}$ 其中, λ_0 满足

$$\nabla f(\bar{\mathbf{x}}^{(1)})^T \mathbf{p}^{(0)} = 0$$

从而

$$[\nabla f(\bar{\mathbf{x}}^{(1)}) - \nabla f(\mathbf{x}^{(1)})]^T \mathbf{p}^{(0)} = 0$$

但由式(12.25)

$$\nabla f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{B}$$

若令

$$\mathbf{p}^{(1)} = \bar{\mathbf{x}}^{(1)} - \mathbf{x}^{(1)}$$

则有

$$(\mathbf{p}^{(1)})^T \mathbf{A} \mathbf{p}^{(0)} = 0$$

即 $\mathbf{p}^{(1)}$ 和 $\mathbf{p}^{(0)}$ 为 \mathbf{A} 共轭。

上述分析说明, 对于二维正定二次函数来说, 从任一点 $\mathbf{x}^{(0)}$ 出发, 沿相互共轭的方向 $\mathbf{p}^{(0)}$ 和 $\mathbf{p}^{(1)}$ 进行两次一维搜索, 即可收敛到函数的极小点。

(2) 正定二次函数的共轭梯度法

对于问题式(12.25)来说, 由于 \mathbf{A} 为对称正定阵, 故存在唯一极小点 \mathbf{x}^* , 它满足方程

$$\nabla f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{B} = 0 \quad (12.27)$$

且具有形式

$$\mathbf{x}^* = -\mathbf{A}^{-1}\mathbf{B} \quad (12.28)$$

如果已知某共轭向量组 $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n-1)}$, 由定理12.1.5可知, 问题式(12.25)的极小点 \mathbf{x}^* 可通过下列算法得到

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)}, & k = 0, 1, 2, \dots, n-1 \\ \lambda_k : \min_{\lambda} f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)}) \\ \mathbf{x}^{(n)} = \mathbf{x}^* \end{cases} \quad (12.29)$$

算法式(12.28)称为共轭方向法。它要求: 搜索方向 $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n-1)}$ 必须共轭; 确定各近似极小点时必须按最优以为搜索进行。

共轭梯度法是共轭方向法的一种, 它的搜索方向是利用一维搜索所得极小点处函数的梯度生成的, 我们现在就来构造正定二次函数的共轭梯度法。

由于 $\nabla f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{B}$, 故有

$$\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}) = \mathbf{A}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$$

但

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)}$$

故

$$\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}) = \lambda_k \mathbf{A} \mathbf{p}^{(k)}, \quad k = 0, 1, 2, \dots, n-1 \quad (12.30)$$

任取初始近似点 $\mathbf{x}^{(0)}$, 并取初始搜索方向为此点的负梯度方向, 即

$$\mathbf{p}^{(0)} = -\nabla f(\mathbf{x}^{(0)})$$

沿射线 $\mathbf{x}^{(0)} + \lambda \mathbf{p}^{(0)}$ 进行一维搜索, 得

$$\begin{cases} \mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda_0 \mathbf{p}^{(0)} \\ \lambda_0 : \min_{\lambda} f(\mathbf{x}^{(0)} + \lambda \mathbf{p}^{(0)}) \end{cases}$$

算出 $\nabla f(\mathbf{x}^{(1)})$, 由式(12.26)知

$$\nabla f(\mathbf{x}^{(1)})^T \mathbf{p}^{(0)} = -\nabla f(\mathbf{x}^{(1)})^T \nabla f(\mathbf{x}^{(0)}) = 0$$

从而可知 $\nabla f(\mathbf{x}^{(1)})$ 和 $\nabla f(\mathbf{x}^{(0)})$ 正交 (这里假设 $\nabla f(\mathbf{x}^{(1)})$ 和 $\nabla f(\mathbf{x}^{(0)})$ 均不等于零)。 $\nabla f(\mathbf{x}^{(0)})$ 和 $\nabla f(\mathbf{x}^{(1)})$ 构成一正交系, 我们可以在由它们生成的二维子空间中寻求 $\mathbf{p}^{(1)}$ 。为此, 可令

$$\mathbf{p}^{(1)} = -\nabla f(\mathbf{x}^{(1)}) + \alpha_0 \nabla f(\mathbf{x}^{(0)})$$

式中, α_0 为待定系数。欲使 $\mathbf{p}^{(1)}$ 与 $\mathbf{p}^{(0)}$ 为 A 共轭, 由式(12.30), 必须

$$[-\nabla f(\mathbf{x}^{(1)}) + \alpha_0 \nabla f(\mathbf{x}^{(0)})]^T [\nabla f(\mathbf{x}^{(1)}) - \nabla f(\mathbf{x}^{(0)})] = 0$$

故

$$-\alpha_0 = \frac{\nabla f(\mathbf{x}^{(1)})^T \nabla f(\mathbf{x}^{(1)})}{\nabla f(\mathbf{x}^{(0)})^T \nabla f(\mathbf{x}^{(0)})}$$

令

$$\beta_0 = -\alpha_0 = \frac{\nabla f(\mathbf{x}^{(1)})^T \nabla f(\mathbf{x}^{(1)})}{\nabla f(\mathbf{x}^{(0)})^T \nabla f(\mathbf{x}^{(0)})} \quad (12.31)$$

由此可得

$$\mathbf{p}^{(1)} = -\nabla f(\mathbf{x}^{(1)}) + \beta_0 \mathbf{p}^{(0)} \quad (12.32)$$

以 $\mathbf{p}^{(1)}$ 为搜索方向进行最优一维搜索, 可得

$$\begin{cases} \mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \lambda_1 \mathbf{p}^{(1)} \\ \lambda_1 : \min_{\lambda} f(\mathbf{x}^{(1)} + \lambda \mathbf{p}^{(1)}) \end{cases}$$

算出 $\nabla f(\mathbf{x}^{(2)})$, 假定 $\nabla f(\mathbf{x}^{(2)}) \neq 0$, 因 $\mathbf{p}^{(0)}$ 和 $\mathbf{p}^{(1)}$ 为 A 共轭, 故

$$\nabla f(\mathbf{x}^{(0)})^T [\nabla f(\mathbf{x}^{(2)}) - \nabla f(\mathbf{x}^{(1)})] = 0$$

但

$$\nabla f(\mathbf{x}^{(0)})^T \nabla f(\mathbf{x}^{(1)}) = 0$$

故

$$\nabla f(\mathbf{x}^{(0)})^T \nabla f(\mathbf{x}^{(2)}) = 0$$

由于

$$\nabla f(\mathbf{x}^{(2)})^T \mathbf{p}^{(1)} = \nabla f(\mathbf{x}^{(2)})^T [-\nabla f(\mathbf{x}^{(1)}) + \alpha_0 \nabla f(\mathbf{x}^{(0)})] = 0$$

所以

$$\nabla f(\mathbf{x}^{(2)})^T \nabla f(\mathbf{x}^{(1)}) = 0$$

即 $\nabla f(\mathbf{x}^{(2)})$ 、 $\nabla f(\mathbf{x}^{(1)})$ 和 $\nabla f(\mathbf{x}^{(0)})$ 构成一正交系。现由它们生成的三维子空间中，寻求与 $\mathbf{p}^{(0)}$ 和 $\mathbf{p}^{(1)}$ 为 \mathbf{A} 共轭的搜索方向 $\mathbf{p}^{(2)}$ 。令

$$\mathbf{p}^{(2)} = -\nabla f(\mathbf{x}^{(2)}) + \alpha_1 \nabla f(\mathbf{x}^{(1)}) + \alpha_0 \nabla f(\mathbf{x}^{(0)})$$

式中， α_0 和 α_1 均为待定系数。由于 $\mathbf{p}^{(2)}$ 应与 $\mathbf{p}^{(0)}$ 和 $\mathbf{p}^{(1)}$ 为 \mathbf{A} 共轭，故须

$$[-\nabla f(\mathbf{x}^{(2)}) + \alpha_1 \nabla f(\mathbf{x}^{(1)}) + \alpha_0 \nabla f(\mathbf{x}^{(0)})]^T [\nabla f(\mathbf{x}^{(1)}) - \nabla f(\mathbf{x}^{(0)})] = 0$$

$$[-\nabla f(\mathbf{x}^{(2)}) + \alpha_1 \nabla f(\mathbf{x}^{(1)}) + \alpha_0 \nabla f(\mathbf{x}^{(0)})]^T [\nabla f(\mathbf{x}^{(2)}) - \nabla f(\mathbf{x}^{(1)})] = 0$$

从而

$$\alpha_1 \nabla f(\mathbf{x}^{(1)})^T \nabla f(\mathbf{x}^{(1)}) - \alpha_0 \nabla f(\mathbf{x}^{(0)})^T \nabla f(\mathbf{x}^{(0)}) = 0$$

$$-\nabla f(\mathbf{x}^{(2)})^T \nabla f(\mathbf{x}^{(2)}) - \alpha_1 \nabla f(\mathbf{x}^{(1)})^T \nabla f(\mathbf{x}^{(1)}) = 0$$

解之得

$$-\alpha_1 = \frac{\nabla f(\mathbf{x}^{(2)})^T \nabla f(\mathbf{x}^{(2)})}{\nabla f(\mathbf{x}^{(1)})^T \nabla f(\mathbf{x}^{(1)})}$$

$$\alpha_0 = \alpha_1 \frac{\nabla f(\mathbf{x}^{(1)})^T \nabla f(\mathbf{x}^{(1)})}{\nabla f(\mathbf{x}^{(0)})^T \nabla f(\mathbf{x}^{(0)})}$$

令 $\beta_1 = -\alpha_1$ ，则 $\alpha_0 = -\beta_1 \beta_0$ ，于是

$$\begin{aligned} \mathbf{p}^{(2)} &= -\nabla f(\mathbf{x}^{(2)}) - \beta_1 \nabla f(\mathbf{x}^{(1)}) - \beta_0 \beta_1 \nabla f(\mathbf{x}^{(0)}) \\ &= -\nabla f(\mathbf{x}^{(2)}) + \beta_1 [-\nabla f(\mathbf{x}^{(1)}) - \beta_0 \nabla f(\mathbf{x}^{(0)})] \\ &= -\nabla f(\mathbf{x}^{(2)}) + \beta_1 [-\nabla f(\mathbf{x}^{(1)}) + \beta_0 \mathbf{p}^{(0)}] \\ &= -\nabla f(\mathbf{x}^{(2)}) + \beta_1 \mathbf{p}^{(1)} \end{aligned} \tag{12.33}$$

继续上述步骤，可得一般公式如下

$$\begin{cases} \mathbf{p}^{(k+1)} = -\nabla f(\mathbf{x}^{(k+1)}) + \beta_k \mathbf{p}^{(k)} \\ \beta_k = \frac{\nabla f(\mathbf{x}^{(k+1)})^T \nabla f(\mathbf{x}^{(k+1)})}{\nabla f(\mathbf{x}^{(k)})^T \nabla f(\mathbf{x}^{(k)})} \end{cases}$$

对于正定二次函数来说， $\nabla f(\mathbf{x}) = \mathbf{Ax} + \mathbf{B}$ ，由式(12.30)

$$\nabla f(\mathbf{x}^{(k+1)}) = \nabla f(\mathbf{x}^{(k)}) + \lambda_k \mathbf{Ap}^{(k)}$$

由于进行的是最优一维搜索，故有

$$\nabla f(\mathbf{x}^{(k+1)})^T \mathbf{p}^{(k)} = 0$$

从而

$$\lambda_k = -\frac{\nabla f(\mathbf{x}^{(k)})^T \mathbf{p}^{(k)}}{(\mathbf{p}^{(k)})^T \mathbf{Ap}^{(k)}}$$

如此，即可得共轭梯度法的一组计算公式如下

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)} \tag{12.34}$$

$$\lambda_k = -\frac{\nabla f(\mathbf{x}^{(k)})^T \mathbf{p}^{(k)}}{(\mathbf{p}^{(k)})^T \mathbf{A} \mathbf{p}^{(k)}} \quad (12.35)$$

$$\mathbf{p}^{(k+1)} = -\nabla f(\mathbf{x}^{(k+1)}) + \beta_k \mathbf{p}^{(k)} \quad (12.36)$$

$$\beta_k = \frac{\nabla f(\mathbf{x}^{(k+1)})^T \nabla f(\mathbf{x}^{(k+1)})}{\nabla f(\mathbf{x}^{(k)})^T \nabla f(\mathbf{x}^{(k)})} \quad (12.37)$$

$$k = 0, 1, 2, \dots, n-1$$

其中, $\mathbf{x}^{(0)}$ 为初始近似, $\mathbf{p}^{(0)} = -\nabla f(\mathbf{x}^{(0)})$ 。

由于 $\mathbf{p}^{(k)} = -\nabla f(\mathbf{x}^{(k)}) + \beta_{k-1} \mathbf{p}^{(k-1)}$ 以及 $\nabla f(\mathbf{x}^{(k)})^T \mathbf{p}^{(k-1)} = 0$, 故式(12.35)也可写成

$$\lambda_k = \frac{\nabla f(\mathbf{x}^{(k)})^T \nabla f(\mathbf{x}^{(k)})}{(\mathbf{p}^{(k)})^T \mathbf{A} \mathbf{p}^{(k)}} \quad (12.38)$$

式(12.37)最先由弗莱彻 (Fletcher) 和瑞夫斯 (Reeves) 提出, 故此法亦称为 FR 共轭梯度法。上述公司还有其它等价形式。例如, 借助于式(12.30), 可将它变为

$$\beta_k = \frac{\nabla f(\mathbf{x}^{(k+1)})^T \mathbf{A} \mathbf{p}^{(k)}}{(\mathbf{p}^{(k)})^T \mathbf{A} \mathbf{p}^{(k)}} \quad (12.39)$$

现将共轭梯度法的计算步骤总结如下:

1. 选择初始近似 $\mathbf{x}^{(0)}$, 给出允许误差 $\varepsilon > 0$;

2. 计算

$$\mathbf{p}^{(0)} = -\nabla f(\mathbf{x}^{(0)})$$

并用式(12.34)和式(12.35)算出 $\mathbf{x}^{(1)}$ 。计算步长也可使用以前介绍过的一维搜索法;

3. 一般地, 假定已得出 $\mathbf{x}^{(k)}$ 和 $\mathbf{p}^{(k)}$, 则可计算其第 $k+1$ 次近似 $\mathbf{x}^{(k+1)}$

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)} \\ \lambda_k : \min_{\lambda} f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)}) \end{cases}$$

4. 若 $\|\nabla f(\mathbf{x}^{(k+1)})\|^2 \leq \varepsilon$, 停止计算, $\mathbf{x}^{(k+1)}$ 即为要求的近似解。否则, 若 $k < n-1$, 则用式(12.37)和式(12.36)计算 β_k 和 $\mathbf{p}^{(k+1)}$, 并转向第 3 步。

应当指出, 对于二次函数的情形, 从理论上说, 进行 n 次迭代即可达到极小点。但是, 在实际计算中, 由于数据的四舍五入以及计算误差的积累, 往往做不到这一点。此外, 由于 n 维问题的共轭方向最多只有 n 个, 在 n 步以后继续如上进行是没有意义的。因此, 在实际应用时, 如迭代到 n 步还不收敛, 就将 $\mathbf{x}^{(n)}$ 作为新的初始近似, 重新开始迭代。根据实际经验, 采用这种再开始的办法, 一般都可得到较好的效果。

例 12.1.4. 试用共轭梯度法求下述二次函数的极小点

$$f(\mathbf{x}) = \frac{3}{2}x_1^2 + \frac{1}{2}x_2^2 - x_1x_2 - 2x_1$$

解. 将 $f(\mathbf{x})$ 化成式(12.25)的形式, 得

$$\mathbf{A} = \begin{pmatrix} 3 & -1 \\ -1 & 1 \end{pmatrix}$$

现从 $\mathbf{x}^{(0)} = (-2, 4)^T$ 开始, 由于

$$\nabla f(\mathbf{x}) = [(3x_1 - x_2 - 2), (x_2 - x_1)]^T$$

故

$$\nabla f(\mathbf{x}^{(0)}) = (-12, 6)^T$$

$$\mathbf{p}^{(0)} = -\nabla f(\mathbf{x}^{(0)}) = (12, -6)^T$$

$$\lambda_0 = -\frac{\nabla f(\mathbf{x}^{(0)})^T \mathbf{p}^{(0)}}{(\mathbf{p}^{(0)})^T \mathbf{A} \mathbf{p}^{(0)}} = -\frac{(-12, 6) \begin{pmatrix} 12 \\ -6 \end{pmatrix}}{(12, -6) \begin{pmatrix} 3 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 12 \\ -6 \end{pmatrix}} = \frac{180}{612} = \frac{5}{17}$$

于是

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda_0 \mathbf{p}^{(0)} = \begin{pmatrix} -2 \\ 4 \end{pmatrix} + \frac{5}{17} \begin{pmatrix} 12 \\ -6 \end{pmatrix} = \left(\frac{26}{17}, \frac{38}{17} \right)^T$$

$$\nabla f(\mathbf{x}^{(1)}) = \left(\frac{6}{17}, \frac{12}{17} \right)^T$$

$$\beta_0 = \frac{\nabla f(\mathbf{x}^{(1)})^T \nabla f(\mathbf{x}^{(1)})}{\nabla f(\mathbf{x}^{(0)})^T \nabla f(\mathbf{x}^{(0)})} = \frac{\left(\frac{6}{17}, \frac{12}{17} \right) \begin{pmatrix} \frac{6}{17} \\ \frac{12}{17} \end{pmatrix}}{(-12, 6) \begin{pmatrix} 12 \\ -6 \end{pmatrix}} = \frac{1}{289}$$

$$\mathbf{p}^{(1)} = -\nabla f(\mathbf{x}^{(1)}) + \beta_0 \mathbf{p}^{(0)} = -\begin{pmatrix} \frac{6}{17} \\ \frac{12}{17} \end{pmatrix} + \frac{1}{289} \begin{pmatrix} 12 \\ -6 \end{pmatrix} = \left(-\frac{90}{289}, -\frac{210}{289} \right)^T$$

$$\begin{aligned} \lambda_1 &= -\frac{\nabla f(\mathbf{x}^{(1)})^T \mathbf{p}^{(1)}}{(\mathbf{p}^{(1)})^T \mathbf{A} \mathbf{p}^{(1)}} \\ &= -\frac{\left(\frac{6}{17}, \frac{12}{17} \right) \left(-\frac{90}{289}, -\frac{210}{289} \right)^T}{\left(-\frac{90}{289}, -\frac{210}{289} \right) \begin{pmatrix} 3 & -1 \\ -1 & 1 \end{pmatrix} \left(-\frac{90}{289}, -\frac{210}{289} \right)^T} \end{aligned}$$

$$= \frac{6 \times 17 \times 90 + 12 \times 17 \times 210}{(-60, -120)(-90, -210)^T} = \frac{17(6 \times 90 + 12 \times 210)}{60 \times 90 + 120 \times 210} = \frac{17}{10}$$

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \lambda_1 \mathbf{p}^{(1)} = \begin{pmatrix} \frac{26}{17} \\ \frac{38}{17} \end{pmatrix} + \frac{17}{10} \begin{pmatrix} -\frac{90}{289} \\ -\frac{210}{289} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

这就是 $f(\mathbf{x})$ 的极小点。图 12.12 表明了本例的搜索方向的步骤。

草稿勿外传

故

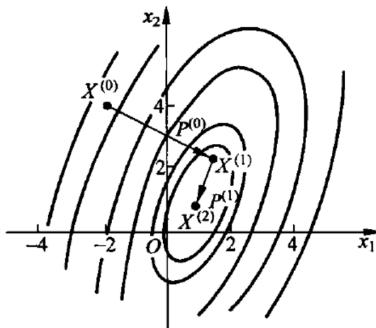


图 12.12

(3) 非二次函数的共轭梯度法

可将共轭梯度法推广到求解一般无约束优化问题式(12.1)。

设 $f(\mathbf{x})$ 为某一严格凸函数, 它具有二阶连续偏导数, 其唯一极小点为 \mathbf{x}^* 。现任取初始近似 $\mathbf{x}^{(0)}$, 计算 $\nabla f(\mathbf{x}^{(0)})$, 选取 $\mathbf{p}^{(0)} = -\nabla f(\mathbf{x}^{(0)})$ 为初始搜索方向, 作射线 $\mathbf{x}^{(0)} + \lambda \mathbf{p}^{(0)} (\lambda \geq 0)$, 并将 $f(\mathbf{x}) = f(\mathbf{x}^{(0)} + \lambda \mathbf{p}^{(0)})$ 附近做泰勒展开

$$f(\mathbf{x}^{(0)} + \lambda \mathbf{p}^{(0)}) \approx f(\mathbf{x}^{(0)}) + \lambda \nabla f(\mathbf{x}^{(0)})^\top \mathbf{p}^{(0)} + \frac{1}{2} \lambda^2 (\mathbf{p}^{(0)})^\top \mathbf{H}(\mathbf{x}^{(0)}) \mathbf{p}^{(0)}$$

上式为 λ 的二次函数, 因 $(\mathbf{p}^{(0)})^\top \mathbf{H}(\mathbf{x}^{(0)}) \mathbf{p}^{(0)} > 0$, 故使该二次函数沿 $\mathbf{p}^{(0)}$ 方向取极小值的 λ 为

$$\lambda_0 = -\frac{\nabla f(\mathbf{x}^{(0)})^\top \mathbf{p}^{(0)}}{(\mathbf{p}^{(0)})^\top \mathbf{H}(\mathbf{x}^{(0)}) \mathbf{p}^{(0)}}$$

显然, 它近似满足 $\min_{\lambda} f(\mathbf{x}^{(0)} + \lambda \mathbf{p}^{(0)})$ 。令

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda_0 \mathbf{p}^{(0)}$$

则 $\mathbf{x}^{(1)}$ 近似满足

$$\nabla f(\mathbf{x}^{(1)})^\top \mathbf{p}^{(0)} = 0$$

现构造向量

$$\mathbf{p}^{(1)} = -\nabla f(\mathbf{x}^{(1)}) + \beta_0 \mathbf{p}^{(0)}$$

使满足 $(\mathbf{p}^{(1)})^\top \mathbf{H}(\mathbf{x}^{(0)}) \mathbf{p}^{(0)} = 0$, 则得

$$\beta_0 = \frac{\nabla f(\mathbf{x}^{(1)})^\top \mathbf{H}(\mathbf{x}^{(0)}) \mathbf{p}^{(0)}}{(\mathbf{p}^{(0)})^\top \mathbf{H}(\mathbf{x}^{(0)}) \mathbf{p}^{(0)}}$$

这就确定了 $\mathbf{p}^{(1)}$ 。

按此手续可构造各次迭代的搜索方向及近似点。一般地, 我们有

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)} \\ \lambda_k &= -\frac{\nabla f(\mathbf{x}^{(k)})^\top \mathbf{p}^{(k)}}{(\mathbf{p}^{(k)})^\top \mathbf{H}(\mathbf{x}^{(k)}) \mathbf{p}^{(k)}} \end{aligned}$$

$$\mathbf{p}^{(k+1)} = -\nabla f(\mathbf{x}^{(k+1)}) + \beta_k \mathbf{p}^{(k)}$$

$$\beta_k = \frac{\nabla f(\mathbf{x}^{(k+1)})^T \mathbf{H}(\mathbf{x}^{(k)}) \mathbf{p}^{(k)}}{(\mathbf{p}^{(k)})^T \mathbf{H}(\mathbf{x}^{(k)}) \mathbf{p}^{(k)}}$$

这就是推广到非二次函数的共轭梯度法的计算公式。

由于在导出上述公式的过程中利用了一些近似关系，以及 $\mathbf{H}(\mathbf{x}^{(k)})$ 的逐次变化，使 $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n-1)}$ 的共轭性遭受破坏，因而对于一般非二次函数来说，要以 n 步迭代取得收敛常常是不可能的。所以在实际应用时，如迭代步数 $k \leq n$ 已达到要求的精度，则以 $\mathbf{x}^{(k)}$ 作为要求的近似解。否则可将前 n 步作为一个循环，同时以所得到的 $\mathbf{x}^{(n)}$ 作为新的初始近似重新开始，进行第二个循环。重复进行，直至满足要求的精度为止。

12.1.3 二阶方法

变尺度法

变尺度法是近 40 多年来发展起来的，它是求解无约束优化问题的一种有效方法。由于它既避免了计算二阶导数矩阵及其求逆过程，又比梯度法的收敛速度快，特别是对高维问题具有显著的优越性，因而使变尺度法获得了很高的声誉，至今仍被公认为求解无约束优化问题最有效的算法之一。下面我们就来简要地介绍变尺度法的基本原理及其计算过程。

(1) 基本原理

假定无约束优化问题的目标函数 $f(\mathbf{x})$ 具有二阶连续偏导数， $\mathbf{x}^{(k)}$ 为某极小点的某一近似。在这个点附近取 $f(\mathbf{x})$ 的二阶泰勒多项式逼近

$$f(\mathbf{x}) \approx f(\mathbf{x}^{(k)}) + \nabla f(\mathbf{x}^{(k)})^T \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T \mathbf{H}(\mathbf{x}^{(k)}) \Delta \mathbf{x} \quad (12.40)$$

则其梯度为

$$\nabla f(\mathbf{x}) \approx \nabla f(\mathbf{x}^{(k)}) + \mathbf{H}(\mathbf{x}^{(k)}) \Delta \mathbf{x} \quad (12.41)$$

这个近似函数的极小点满足

$$\nabla f(\mathbf{x}^{(k)}) + \mathbf{H}(\mathbf{x}^{(k)}) \Delta \mathbf{x} = 0$$

从而

$$\mathbf{x} = \mathbf{x}^{(k)} - \mathbf{H}(\mathbf{x}^{(k)})^{-1} \nabla f(\mathbf{x}^{(k)}) \quad (12.42)$$

其中 $\mathbf{H}(\mathbf{x}^{(k)})$ 为 $f(\mathbf{x})$ 在 $\mathbf{x}^{(k)}$ 点的 Hessian 矩阵。

如果 $f(\mathbf{x})$ 是二次函数，则 $\mathbf{H}(\mathbf{x})$ 为常数阵。这时，逼近式(12.40)是准确的。在这种情况下，从任一点 $\mathbf{x}^{(k)}$ 出发，用式(12.42)只要一步即可求出 $f(\mathbf{x})$ 的极小点（假定 $\mathbf{H}(\mathbf{x}^{(k)})$ 正定）。

当 $f(\mathbf{x})$ 不是二次函数时，式(12.40)仅是 $f(\mathbf{x})$ 在 $\mathbf{x}^{(k)}$ 点附近的近似表达式。这时，按式(12.42)求得的极小点，只是 $f(\mathbf{x})$ 的极小点的近似。在这种情况下，人们常取 $-\mathbf{H}(\mathbf{x}^{(k)})^{-1} \nabla f(\mathbf{x}^{(k)})$ 为搜索

方向, 即

$$\begin{cases} \mathbf{p}^{(k)} = -\mathbf{H}(\mathbf{x}^{(k)})^{-1} \nabla f(\mathbf{x}^{(k)}) \\ \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)} \\ \lambda_k : \min_{\lambda} f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)}) \end{cases} \quad (12.43)$$

按照这种方式求函数 $f(\mathbf{x})$ 的极小点的方法, 称作广义牛顿法。式(12.43)确定的搜索方向, 为 $f(\mathbf{x})$ 在点 $\mathbf{x}^{(k)}$ 的牛顿方向。牛顿法的收敛速度很快, 当 $f(\mathbf{x})$ 的二阶导数及其 Hessian 矩阵的逆阵便于计算时, 使用这一方法非常有效。

问题在于, 实际问题中的目标函数往往相当复杂, 计算二阶导数的工作量或者太大, 或者根本不可能。况且, 在 \mathbf{x} 的维数很高时, 计算逆阵也相当费事。为了不计算二阶导数矩阵 $\mathbf{H}(\mathbf{x}^{(k)})$, 从而也不必计算其逆阵 $\mathbf{H}(\mathbf{x}^{(k)})^{-1}$, 我们设法构造另一个矩阵 $\bar{\mathbf{H}}^{(k)}$, 用它来直接逼近二阶导数矩阵的逆阵 $\mathbf{H}(\mathbf{x}^{(k)})^{-1}$ 。

下面就来研究如何构造 $\mathbf{H}(\mathbf{x}^{(k)})^{-1}$ 的近似矩阵 $\bar{\mathbf{H}}^{(k)}$ 。我们要求, 在每一步都能以现有的信息来确定下一个搜索方向; 每做一次迭代, 目标函数值均有所下降; 而且, 这些近似矩阵最后应收敛于解点处的 Hessian 矩阵的逆阵。

当 $f(\mathbf{x})$ 是二次函数时, 其 Hessian 矩阵为常数阵, 可知其在两点 $\mathbf{x}^{(k)}$ 和 $\mathbf{x}^{(k+1)}$ 处的梯度之差等于

$$\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}) = \mathbf{A}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})$$

或

$$\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = \mathbf{A}^{-1}[\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)})] \quad (12.44)$$

对于非二次函数, 仿照二次函数的情形, 要求其 Hessian 矩阵的逆阵的第 $k+1$ 次近似矩阵 $\bar{\mathbf{H}}^{(k+1)}$ 满足关系式

$$\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = \bar{\mathbf{H}}^{(k+1)}[\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)})] \quad (12.45)$$

此式就是所谓的拟牛顿条件。

若令

$$\begin{cases} \Delta \mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}) \\ \Delta \mathbf{x}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \end{cases} \quad (12.46)$$

则式(12.45)变为

$$\Delta \mathbf{x}^{(k)} = \bar{\mathbf{H}}^{(k+1)} \Delta \mathbf{g}^{(k)}$$

现设 $\bar{\mathbf{H}}^{(k)}$ 已知, 并用下式求 $\bar{\mathbf{H}}^{(k+1)}$ (假定 $\bar{\mathbf{H}}^{(k)}$ 和 $\bar{\mathbf{H}}^{(k+1)}$ 都为对称正定阵)

$$\bar{\mathbf{H}}^{(k+1)} = \bar{\mathbf{H}}^{(k)} + \Delta \bar{\mathbf{H}}^{(k)} \quad (12.47)$$

上式中 $\bar{\mathbf{H}}^{(k)}$ 为第 k 次校正矩阵, $\bar{\mathbf{H}}^{(k+1)}$ 应满足拟牛顿条件, 即要求

$$\Delta \mathbf{x}^{(k)} = (\bar{\mathbf{H}}^{(k)} + \Delta \bar{\mathbf{H}}^{(k)}) \Delta \mathbf{g}^{(k)}$$

或

$$\Delta \bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)} = \Delta \mathbf{x}^{(k)} - \bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)} \quad (12.48)$$

由此可以设想 $\Delta \bar{\mathbf{H}}^{(k)}$ 的一种较简单形式为

$$\Delta \bar{\mathbf{H}}^{(k)} = \Delta \mathbf{x}^{(k)} (\mathbf{q}^{(k)})^T - \bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)} (\mathbf{w}^{(k)})^T \quad (12.49)$$

式中 $\mathbf{q}^{(k)}$ 和 $\mathbf{w}^{(k)}$ 为两个特定向量。

将表达式(12.49)代入式(12.48)得

$$\Delta \mathbf{x}^{(k)} (\mathbf{g}^{(k)})^T \Delta \mathbf{g}^{(k)} - \bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)} (\mathbf{w}^{(k)})^T \Delta \mathbf{g}^{(k)} = \Delta \mathbf{x}^{(k)} - \bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)}$$

这就是说, 应使

$$(\mathbf{q}^{(k)})^T \Delta \mathbf{g}^{(k)} = (\mathbf{w}^{(k)})^T \Delta \mathbf{g}^{(k)} = 1 \quad (12.50)$$

由于 $\Delta \bar{\mathbf{H}}^{(k)}$ 应为对称阵, 最简单的方法就是取

$$\begin{cases} \mathbf{q}^{(k)} = \eta_k \Delta \mathbf{x}^{(k)} \\ \mathbf{w}^{(k)} = \xi_k \bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)} \end{cases} \quad (12.51)$$

由式(12.50)

$$\eta_k (\Delta \mathbf{x}^{(k)})^T \Delta \mathbf{g}^{(k)} = \xi_k (\Delta \mathbf{g}^{(k)})^T \bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)} = 1$$

设 $(\Delta \mathbf{x}^{(k)})^T \Delta \mathbf{g}^{(k)}$ 以及 $(\Delta \mathbf{g}^{(k)})^T \bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)}$ 皆不为零, 则有

$$\begin{cases} \eta_k = \frac{1}{(\Delta \mathbf{x}^{(k)})^T \Delta \mathbf{g}^{(k)}} = \frac{1}{(\Delta \mathbf{g}^{(k)})^T \Delta \mathbf{x}^{(k)}} \\ \xi_k = \frac{1}{(\Delta \mathbf{g}^{(k)})^T \bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)}} \end{cases} \quad (12.52)$$

于是得到校正矩阵

$$\Delta \bar{\mathbf{H}}^{(k)} = \frac{\Delta \mathbf{x}^{(k)} (\Delta \mathbf{x}^{(k)})^T}{(\Delta \mathbf{g}^{(k)})^T \Delta \mathbf{x}^{(k)}} - \frac{\bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)} (\Delta \mathbf{g}^{(k)})^T \bar{\mathbf{H}}^{(k)}}{(\Delta \mathbf{g}^{(k)})^T \bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)}} \quad (12.53)$$

从而得到

$$\bar{\mathbf{H}}^{(k+1)} = \bar{\mathbf{H}}^{(k)} + \frac{\Delta \mathbf{x}^{(k)} (\Delta \mathbf{x}^{(k)})^T}{(\Delta \mathbf{g}^{(k)})^T \Delta \mathbf{x}^{(k)}} - \frac{\bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)} (\Delta \mathbf{g}^{(k)})^T \bar{\mathbf{H}}^{(k)}}{(\Delta \mathbf{g}^{(k)})^T \bar{\mathbf{H}}^{(k)} \Delta \mathbf{g}^{(k)}} \quad (12.54)$$

上述矩阵称为尺度矩阵, 在整个迭代过程中它是在不断变化的。有了尺度矩阵, 即可依式(12.43)进行迭代计算。

(2) 计算步骤 现将变尺度法的计算步骤总结如下:

1. 给定初始点 $\mathbf{x}^{(0)}$ 及梯度允许误差 $\varepsilon > 0$;

2. 若

$$\|\nabla f(\mathbf{x}^{(0)})\|^2 \neq \varepsilon$$

则 $\mathbf{x}^{(0)}$ 即为近似极小点, 停止迭代。否则, 转向下一步;

3. 令

$$\bar{\mathbf{H}}^{(0)} = \mathbf{I} \text{(单位阵)}$$

$$\mathbf{p}^{(0)} = -\bar{\mathbf{H}}^{(0)} \nabla f(\mathbf{x}^{(0)})$$

在 $\mathbf{p}^{(0)}$ 方向进行一维搜索, 确定最佳步长 λ_0

$$\min_{\lambda} f(\mathbf{x}^{(0)} + \lambda \mathbf{p}^{(0)}) = f(\mathbf{x}^{(0)} + \lambda_0 \mathbf{p}^{(0)})$$

如此可得下一个近似点

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda_0 \mathbf{p}^{(0)}$$

4. 一般地, 设已得到近似点 $\mathbf{x}^{(k)}$, 算出 $\nabla f(\mathbf{x}^{(k)})$, 若

$$\|\nabla f(\mathbf{x}^{(k)})\|^2 \leq \varepsilon$$

则 $\mathbf{x}^{(k)}$ 即为所求的近似解, 停止迭代; 否则, 按式(12.54)计算 $\bar{\mathbf{H}}^{(k)}$, 并令

$$\mathbf{p}^{(k)} = -\bar{\mathbf{H}}^{(k)} \nabla f(\mathbf{x}^{(k)})$$

在 $\mathbf{p}^{(k)}$ 方向进行一维搜索, 确定最佳最长 λ_k

$$\min_{\lambda} f(\mathbf{x}^{(k)} + \lambda \mathbf{p}^{(k)}) = f(\mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)})$$

其下一个近似点为

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{p}^{(k)}$$

5. 若 $\mathbf{x}^{(k+1)}$ 点满足精度要求, 则 $\mathbf{x}^{(k+1)}$ 即为所求的近似解。否则, 转回第 (4) 步, 直到求出某点满足精度要求为止。

和共轭梯度法相类似, 如果迭代 n 次仍不收敛, 则以 $\mathbf{x}^{(n)}$ 为新的 $\mathbf{x}^{(0)}$, 以这时的 $\mathbf{x}^{(0)}$ 为起点重新开始一轮新的迭代。

上述方法首先由戴维顿 (Davidon) 提出, 后经弗莱彻 (Fletcher) 和鲍威尔 (Powell) 加以改进, 故称 DFP 法, 或 DFP 变尺度法。

例 12.1.5. 试用 DFP 法重新计算例 12.1.4。

解. 和例 12.1.4一样, 仍从 $\mathbf{x}^{(0)} = (-2, 4)^T$ 开始, 并取

$$\bar{\mathbf{H}}^{(0)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\nabla f(\mathbf{x}) = [(3x_1 - x_2 - 2), (x_2 - x_1)]^T$$

$$\nabla f(\mathbf{x}^{(0)}) = (-12, 6)^T$$

$$\mathbf{p}^{(0)} = -\bar{\mathbf{H}}^{(0)} \nabla f(\mathbf{x}^{(0)}) = -\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -12 \\ 6 \end{pmatrix} = \begin{pmatrix} 12 \\ -6 \end{pmatrix}$$

外
傳
記
稿
草

利用一维搜索，即 $\min_{\lambda} f(\mathbf{x}^{(0)} + \lambda \mathbf{p}^{(0)})$ ，可算得

$$\lambda_0 = \frac{5}{17}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda_0 \mathbf{p}^{(0)} = \begin{pmatrix} -2 \\ 4 \end{pmatrix} + \frac{5}{17} \begin{pmatrix} 12 \\ -6 \end{pmatrix} = \begin{pmatrix} \frac{26}{17} \\ \frac{38}{17} \end{pmatrix}^T$$

$$\nabla f(\mathbf{x}^{(1)}) = \left(\frac{6}{17}, \frac{12}{17} \right)^T$$

$$\Delta \mathbf{x}^{(0)} = \mathbf{x}^{(1)} - \mathbf{x}^{(0)} = \left(\frac{26}{17}, \frac{38}{17} \right)^T - (-2, 4)^T = \left(\frac{60}{17}, -\frac{30}{17} \right)^T$$

$$\begin{aligned} \Delta \mathbf{g}^{(0)} &= \nabla f(\mathbf{x}^{(1)}) - \nabla f(\mathbf{x}^{(0)}) \\ &= \left(\frac{6}{17}, \frac{12}{17} \right)^T - (-12, 6)^T = \left(\frac{210}{17}, -\frac{90}{17} \right)^T \end{aligned}$$

$$\begin{aligned} \bar{\mathbf{H}}^{(1)} &= \bar{\mathbf{H}}^{(0)} + \frac{\Delta \mathbf{x}^{(0)} (\Delta \mathbf{x}^{(0)})^T}{(\Delta \mathbf{g}^{(0)})^T \Delta \mathbf{x}^{(0)}} - \frac{\bar{\mathbf{H}}^{(0)} \Delta \mathbf{g}^{(0)} (\Delta \mathbf{g}^{(0)})^T \bar{\mathbf{H}}^{(0)}}{(\Delta \mathbf{g}^{(0)})^T \bar{\mathbf{H}}^{(0)} \Delta \mathbf{g}^{(0)}} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{\left(\frac{60}{17}, -\frac{30}{17} \right)^T \left(\frac{60}{17}, -\frac{30}{17} \right)}{\left(\frac{210}{17}, -\frac{90}{17} \right)^T \left(\frac{60}{17}, -\frac{30}{17} \right)} - \frac{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \left(\frac{210}{17}, -\frac{90}{17} \right)^T \left(\frac{210}{17}, -\frac{90}{17} \right) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}{\left(\frac{210}{17}, -\frac{90}{17} \right)^T \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \left(\frac{210}{17}, -\frac{90}{17} \right)^T} \end{aligned}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{17} \begin{pmatrix} 4 & -2 \\ -2 & 1 \end{pmatrix} - \frac{1}{58} \begin{pmatrix} 49 & -21 \\ -21 & 9 \end{pmatrix} = \frac{1}{986} \begin{pmatrix} 385 & 241 \\ 241 & 891 \end{pmatrix}$$

$$\mathbf{p}^{(1)} = -\bar{\mathbf{H}}^{(1)} \nabla f(\mathbf{x}^{(1)}) = -\frac{1}{986} \begin{pmatrix} 385 & 241 \\ 241 & 891 \end{pmatrix} \begin{pmatrix} \frac{6}{17} \\ \frac{12}{17} \end{pmatrix} = -\begin{pmatrix} \frac{9}{29} \\ \frac{21}{29} \end{pmatrix}$$

再由一维搜索 $\min_{\lambda} f(\mathbf{x}^{(1)} + \lambda \mathbf{p}^{(1)})$ ，得

$$\lambda_1 = \frac{29}{17}$$

从而

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \lambda_1 \mathbf{p}^{(1)} = \begin{pmatrix} \frac{26}{17} \\ \frac{38}{17} \end{pmatrix} + \frac{29}{17} \begin{pmatrix} -\frac{9}{29} \\ -\frac{21}{29} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\nabla f(\mathbf{x}^{(2)}) = (0, 0)^T$$

可知 $\mathbf{x}^{(2)} = (1, 1)^T$ 为极小点。

请勿外传

在以上讨论中, 我们取第一个尺度矩阵 $\bar{\mathbf{H}}^{(0)}$ 为对称正定阵, 以后的尺度矩阵由式(12.54)逐步形成。可以证明, 这样构成的尺度矩阵均为对称正定阵。由此可知其搜索方向 $\mathbf{p}^{(k)} = -\bar{\mathbf{H}}^{(k)} \nabla f(\mathbf{x}^{(k)})$ 为下降方向, 这就可以保证每次迭代均能使目标函数值有所改善。

当把 DFP 变尺度法用于正定二次函数时, 产生的搜索方向为共轭方向, 因而也具有有限步收敛的性质。若将初始尺度矩阵也取为单位矩阵, 对这种函数来说, DFP 法就与共轭梯度法一样了。

还要指出, 可以采用不同的方法来构造尺度矩阵 $\bar{\mathbf{H}}^{(k)}$, 从而就有不同的变尺度法。DFP 法属于拟牛顿法的一种。开始时取 $\bar{\mathbf{H}}^{(0)} = \mathbf{I}$, 这相当于第一步采用最速下降法。以后的 $\bar{\mathbf{H}}^{(k)}$ 接近于 $\mathbf{H}(\mathbf{x}^{(k)})^{-1}$, 当达到极小点时, 从理论上讲, 这时的尺度矩阵应等于该点处 Hessian 矩阵的逆阵。

例 12.1.6. 试用 DFP 法求

$$\min f(\mathbf{x}) = 4(x_1 - 5)^2 + (x_2 - 6)^2$$

解. 取

$$\bar{\mathbf{H}}^{(0)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{x}^{(0)} = \begin{pmatrix} 8 \\ 9 \end{pmatrix}$$

由于

$$\nabla f(\mathbf{x}) = [8(x_1 - 5), 2(x_2 - 6)]^T$$

$$\nabla f(\mathbf{x}^{(0)}) = (24, 6)^T$$

故

$$\begin{aligned} \mathbf{x}^{(1)} &= \mathbf{x}^{(0)} + \lambda_0 \mathbf{p}^{(0)} = \mathbf{x}^{(0)} + \lambda_0 [-\bar{\mathbf{H}}^{(0)} \nabla f(\mathbf{x}^{(0)})] \\ &= \begin{pmatrix} 8 \\ 9 \end{pmatrix} - \lambda_0 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 24 \\ 6 \end{pmatrix} = \begin{pmatrix} 8 \\ 9 \end{pmatrix} - \lambda_0 \begin{pmatrix} 24 \\ 6 \end{pmatrix} \\ &= \begin{pmatrix} 8 - 24\lambda_0 \\ 9 - 6\lambda_0 \end{pmatrix} \end{aligned}$$

$$f(\mathbf{x}^{(1)}) = 4[(8 - 24\lambda_0) - 5]^2 + [(9 - 6\lambda_0) - 6]^2$$

令

$$\frac{df(\mathbf{x}^{(1)})}{d\lambda_0} = 0$$

可得

$$\lambda_0 = \frac{17}{130}$$

$$\mathbf{x}^{(1)} = [(8 - 24\lambda_0), (9 - 6\lambda_0)]^T = (4.862, 8.215)^T$$

$$\Delta \mathbf{x}^{(0)} = \mathbf{x}^{(1)} - \mathbf{x}^{(0)} = (-3.138, -0.785)^T$$

$$f(\mathbf{x}^{(1)}) = 4.985$$

$$\nabla f(\mathbf{x}^{(1)}) = (-1.108, 4.431)^T$$

$$\Delta \mathbf{g}^{(0)} = \nabla f(\mathbf{x}^{(1)}) - \nabla f(\mathbf{x}^{(0)}) = (-25.108, -1.569)^T$$

由此可得

$$\begin{aligned}\bar{\mathbf{H}}^{(1)} &= \bar{\mathbf{H}}^{(0)} + \frac{\Delta \mathbf{x}^{(0)} (\Delta \mathbf{x}^{(0)})^T}{(\Delta \mathbf{g}^{(0)})^T \Delta \mathbf{x}^{(0)}} - \frac{\bar{\mathbf{H}}^{(0)} \Delta \mathbf{g}^{(0)} (\Delta \mathbf{g}^{(0)})^T \bar{\mathbf{H}}^{(0)}}{(\Delta \mathbf{g}^{(0)})^T \bar{\mathbf{H}}^{(0)} \Delta \mathbf{g}^{(0)}} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{(-3.138, -0.785)^T (-3.138, -0.785)}{(-25.108, -1.569) (-3.138, -0.785)^T} - \\ &\quad \frac{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} (-25.108, -1.569)^T (-25.108, -1.569) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}{(-25.108, -1.569) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} (-25.108, -1.569)^T} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0.1231 & 0.0308 \\ 0.0308 & 0.0077 \end{pmatrix} - \begin{pmatrix} 0.9961 & 0.0622 \\ 0.0622 & 0.0039 \end{pmatrix} = \begin{pmatrix} 0.1270 & -0.0315 \\ -0.0315 & 1.0038 \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\mathbf{x}^{(2)} &= \mathbf{x}^{(1)} - \lambda_1 \bar{\mathbf{H}}^{(1)} \nabla f(\mathbf{x}^{(1)}) \\ &= \begin{pmatrix} 4.862 \\ 8.215 \end{pmatrix} - \lambda_1 \begin{pmatrix} 0.1270 & -0.0315 \\ -0.0315 & 1.0038 \end{pmatrix} \begin{pmatrix} -1.108 \\ 4.431 \end{pmatrix}\end{aligned}$$

如上求最佳步长，可得

$$\lambda_1 = 0.4942$$

代入上式得

$$\mathbf{x}^{(2)} = (5, 6)^T$$

这就是极小点。

若将该问题的目标函数 $f(\mathbf{x})$ 表示成式(12.25)的形式，可知

$$\mathbf{A} = \begin{pmatrix} 8 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\mathbf{A}^{-1} = \begin{pmatrix} \frac{1}{8} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$$

现计算出该问题的 $\bar{H}^{(2)}$

$$\bar{H}^{(2)} = \begin{pmatrix} 1.25 \times 10^{-1} & -8.882 \times 10^{-16} \\ -8.882 \times 10^{-16} & 5.00 \times 10^{-1} \end{pmatrix}$$

可知二者实际相等。

在以上几节中，我们介绍了求解无约束优化问题的解析法，这些方法只是众多算法中的一部分。一般认为，从迭代次数上考虑，变尺度法所需迭代次数较少，共轭梯度法次之，最速下降法所需迭代次数最多。但从每次迭代所需的计算工作量来看，却正好相反，最速下降法最简单，变尺度法比它们都繁琐。

12.2 约束优化

实际工作中遇到的大多数优化问题，其变量的取值多受到一定限制，这种限制由约束条件来体现。带有约束条件的优化问题称为约束优化问题，其一般形式为

$$\left\{ \begin{array}{l} \min f(\mathbf{x}) \\ h_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, m \\ g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \dots, l \end{array} \right. \quad (12.55)$$

或

$$\left\{ \begin{array}{l} \min f(\mathbf{x}) \\ g_j(\mathbf{x}) \geq 0, \quad j = 1, 2, \dots, l \end{array} \right. \quad (12.56)$$

求解约束优化问题要比求解无约束优化问题困难得多。对有约束的极小化问题来说，除了要使目标函数在每次迭代有所下降之外，还要时刻注意解的可行性问题（某些算法的中间步骤除外），这就给寻优工作带来了很大困难。为了实际求解和（或）简化其优化工作，可采用以下方法：将约束问题化为无约束问题；将约束优化问题转化为线性规划问题，以及能将复杂问题变换为较简单的其它方法。

现考虑上述一般的约束优化问题，假定 $f(\mathbf{x})$ 、 $h_i(\mathbf{x})$ 和 $g_j(\mathbf{x})(i = 1, 2, \dots, m; j = 1, 2, \dots, l)$ 具有一阶连续偏导数。

设 $\mathbf{x}^{(0)}$ 是约束优化的一个可行解，它当然满足所有约束。现考虑某一不等式约束条件 $g_j(\mathbf{x}) \geq 0$ ， $\mathbf{x}^{(0)}$ 满足它有两种可能：某一为 $g_j(\mathbf{x}^{(0)}) > 0$ ，这时，点 $\mathbf{x}^{(0)}$ 不是处于由这一约束条件形成的可行域边界上，因而这一约束对 $\mathbf{x}^{(0)}$ 点的微小摄动不起限制作用，从而称这个约束条件是 $\mathbf{x}^{(0)}$ 点的不起作用约束（或无效约束）；其二是 $g_j(\mathbf{x}^{(0)}) = 0$ ，这时 $\mathbf{x}^{(0)}$ 点处于该约束条件形成的可行域边界上，它对 $\mathbf{x}^{(0)}$ 的摄动起到了某种限制作用，故称这个约束是 $\mathbf{x}^{(0)}$ 点的起作用约束（有效约束）。

显而易见，等式约束对所有可行点来说都是起作用约束。

假定 $\mathbf{x}^{(0)}$ 是约束优化(12.56)的一个可行点, 现考虑此点的某一方向 \mathbf{d} , 若存在实数 $\lambda_0 > 0$, 使对于任意 $\lambda \in [0, \lambda_0]$ 均有

$$\mathbf{x}^{(0)} + \lambda \mathbf{d} \quad (12.57)$$

满足约束条件, 就称方向 \mathbf{d} 是 $\mathbf{x}^{(0)}$ 点的一个可行方向。

若 \mathbf{d} 是可行点 $\mathbf{x}^{(0)}$ 处的任一可行方向 (图12.13), 则对该点的所有起作用约束

$$g_j(\mathbf{x}) \geq 0$$

均有

$$\nabla g_j(\mathbf{x}^{(0)})^T \mathbf{d} \geq 0, \quad j \in J \quad (12.58)$$

其中 J 为这个点所有起作用约束下标的集合。

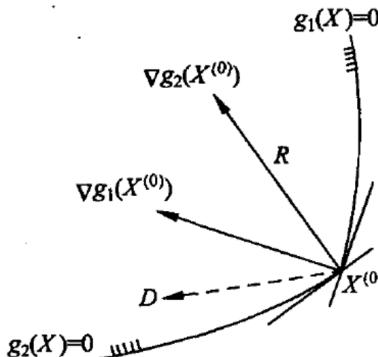


图 12.13

另一方面, 由泰勒公式

$$g_j(\mathbf{x}^{(0)} + \lambda \mathbf{d}) = g_j(\mathbf{x}^{(0)}) + \lambda \nabla g_j(\mathbf{x}^{(0)})^T \mathbf{d} + o(\lambda)$$

可知对所有起作用约束, 当 $\lambda > 0$ 足够小时, 只要

$$\nabla g_j(\mathbf{x}^{(0)})^T \mathbf{d} > 0, \quad j \in J \quad (12.59)$$

就有

$$g_j(\mathbf{x}^{(0)} + \lambda \mathbf{d}) \geq 0, \quad j \in J$$

此外, 对 $\mathbf{x}^{(0)}$ 点的不起作用约束, 由约束函数的连续性, 当 $\lambda > 0$ 足够小时亦有上式成立。从而, 只要方向 \mathbf{d} 满足式(12.59), 即可保证它是 $\mathbf{x}^{(0)}$ 点的可行方向。

考虑约束优化的某一可行点 $\mathbf{x}^{(0)}$, 对该点的任一方向 \mathbf{d} 来说, 若存在实数 $\lambda'_0 > 0$, 使对任意 $\lambda \in [0, \lambda'_0]$ 均有

$$f(\mathbf{x}^{(0)} + \lambda \mathbf{d}) < f(\mathbf{x}^{(0)})$$

就称方向 \mathbf{d} 为 $\mathbf{x}^{(0)}$ 点的一个下降方向。

将目标函数 $f(\mathbf{x})$ 在点 $\mathbf{x}^{(0)}$ 处作一阶泰勒展开, 可知满足条件

$$\nabla f(\mathbf{x}^{(0)})^T \mathbf{d} < 0 \quad (12.60)$$

的方向 \mathbf{d} 必为 $\mathbf{x}^{(0)}$ 点的下降方向。

如果方向 \mathbf{d} 既是 $\mathbf{x}^{(0)}$ 点的可行方向, 又是这个点的下降方向, 就称它是该点的可行下降方向。如果 $\mathbf{x}^{(0)}$ 点不是极小点, 继续寻优时的搜索方向就应从该点的可行下降方向中去找。显然, 若某点存在可行下降方向, 它就不会是极小点。另外, 若某点为极小点, 则在该点不存在可行下降方向。

定理 12.2.1. 设 \mathbf{x}^* 是约束优化式(12.56)的一个局部极小点, 目标函数 $f(\mathbf{x})$ 在 \mathbf{x}^* 处可微, 而且

$g_j(\mathbf{x})$ 在 \mathbf{x}^* 处可微, 当 $j \in J$

$g_j(\mathbf{x})$ 在 \mathbf{x}^* 处连续, 当 $j \notin J$ 则在 \mathbf{x}^* 点不存在可行下降方向, 从而不存在向量 \mathbf{d} 同时满足:

$$\begin{cases} \nabla f(\mathbf{x}^*)^T \mathbf{d} < 0 \\ \nabla g_j(\mathbf{x}^*)^T \mathbf{d} > 0, \quad j \in J \end{cases} \quad (12.61)$$

这个定理显然是成立的。事实上, 若存在满足式(12.61)的方向 \mathbf{d} , 则沿该方向搜索可找到更好的可行点, 从而与 \mathbf{x}^* 为极小点的假设矛盾。

式(12.61)的集合意义十分明显。满足该条件的方向 \mathbf{d} , 与点 \mathbf{x}^* 处目标函数负梯度方向的夹角为锐角, 与点 \mathbf{x}^* 处起作用约束梯度方向的夹角也为锐角。

12.2.1 可行方向法

现考虑约束优化式(12.56), 设 $\mathbf{x}^{(k)}$ 是它的一个可行解, 但不是要求的极小点。为了求它的极小点或近似极小点, 根据以前所说, 应在 $\mathbf{x}^{(k)}$ 点的可行下降方向中选取某一方向 $\mathbf{d}^{(k)}$, 并确定步长 λ_k , 使

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{d}^{(k)} \text{ 满足约束条件} \\ f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)}) \end{cases} \quad (12.62)$$

若满足精度要求, 迭代停止, $\mathbf{x}^{(k+1)}$ 就是所要的点。否则, 从 $\mathbf{x}^{(k+1)}$ 出发继续进行迭代, 直到满足要求为止。上述这种方法称为可行方向法, 它具有下述特点: (1) 迭代过程中所采用的搜索方向为可行方向; (2) 所产生的迭代点列 $\{\mathbf{x}^{(k)}\}$ 始终在可行域内; (3) 目标函数值单调下降。由此可见, 很多方法都可以归入可行方向法一类。但我们通常所说的可行方向法, 一般指的是 Zoutendijk 在 1960 年提出的算法及其变形, 下面就来说明 Zoutendijk 的可行方向法。

设 $\mathbf{x}^{(k)}$ 点的起作用约束集非空, 为求 $\mathbf{x}^{(k)}$ 点的可行下降方向, 可由下述不等式组确定向量 \mathbf{d}

$$\begin{cases} \nabla f(\mathbf{x}^{(k)})^T \mathbf{d} < 0 \\ \nabla g_j(\mathbf{x}^{(k)})^T \mathbf{d} > 0, \quad j \in J \end{cases} \quad (12.63)$$

这等价于由下面的不等式组求向量 \mathbf{d} 和实数 η

$$\begin{cases} \nabla f(\mathbf{x}^{(k)})^T \mathbf{d} \leq \eta \\ -\nabla g_j(\mathbf{x}^{(k)})^T \mathbf{d} \leq \eta, \quad j \in J \\ \eta < 0 \end{cases} \quad (12.64)$$

现使 $\nabla f(\mathbf{x}^{(k)})^T \mathbf{d}$ 和 $-\nabla g_j(\mathbf{x}^{(k)})^T \mathbf{d}$ (对所有 $j \in J$) 的最大值极小化 (必须同时限制向量 \mathbf{d} 的模), 即可将上述选取搜索方向的工作, 转换为求解下述线性规划问题

$$\begin{cases} \min \eta \\ \nabla f(\mathbf{x}^{(k)})^T \mathbf{d} \leq \eta \\ -\nabla g_j((\mathbf{x}^{(k)})^T \mathbf{d}) \leq \eta, \quad j \in J(\mathbf{x}^{(k)}) \\ -1 \leq d_i \leq 1, \quad i = 1, 2, \dots, n \end{cases} \quad (12.65)$$

式中 $d_i (i = 1, 2, \dots, n)$ 为向量 \mathbf{d} 的分量。在式(12.64)中加入最后一个限制条件, 为的是使该线性规划有有限最优解; 由于我们的目的在于寻找搜索方向 \mathbf{d} , 只需知道 \mathbf{d} 的各分量的相对大小即可。

将线性规划式(12.65)的最优解记为 $(\mathbf{d}^{(k)}, \eta_k)$, 如果求出的 $\eta_k = 0$, 说明在 $\mathbf{x}^{(k)}$ 点不存在可行下降方向, 在 $\nabla g_j(\mathbf{x}^{(k)})$ (此处 $j \in J(\mathbf{x}^{(k)})$) 线性无关的条件下, $\mathbf{x}^{(k)}$ 满足 KKT 条件。若解出的 $\eta_k < 0$, 则得到可行下降方向 $\mathbf{d}^{(k)}$, 这就是我们所要的搜索方向。

上述可行方向法的迭代步骤如下:

1. 确定允许误差 $\varepsilon_1 > 0$ 和 $\varepsilon_2 > 0$, 选初始近似点 $\mathbf{x}^{(0)}$ 满足约束条件, 并令 $k := 0$;

2. 确定起作用约束指标集

$$J(\mathbf{x}^{(k)}) = \{j | g_j(\mathbf{x}^{(k)}) = 0, 1 \leq j \leq l\}$$

(1) 若 $J(\mathbf{x}^{(k)}) = \emptyset$ (\emptyset 为空集), 而且 $\|\nabla f(\mathbf{x}^{(k)})\| \leq \varepsilon_1$, 停止迭代, 得点 $\mathbf{x}^{(k)}$;

(2) 若 $J(\mathbf{x}^{(k)}) = \emptyset$, 但 $\|\nabla f(\mathbf{x}^{(k)})\| > \varepsilon_1$, 则取搜索方向 $\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$, 然后转向第 5 步;

(3) 若 $J(\mathbf{x}^{(k)}) \neq \emptyset$, 转下一步;

3. 求解线性规划

$$\begin{cases} \min \eta \\ \nabla f(\mathbf{x}^{(k)})^T \mathbf{d} \leq \eta \\ -\nabla g_j((\mathbf{x}^{(k)})^T \mathbf{d}) \leq \eta, \quad j \in J(\mathbf{x}^{(k)}) \\ -1 \leq d_i \leq 1, \quad i = 1, 2, \dots, n \end{cases}$$

设它的最优解是 $(\mathbf{d}^{(k)}, \eta_k)$:

4. 检验是否满足

$$|\eta_k| \leq \varepsilon_2$$

若满足则停止迭代, 得到点 $\mathbf{x}^{(k)}$; 否则, 以 $\mathbf{d}^{(k)}$ 为搜索方向, 并转下一步;

5. 解下述一维优化问题

$$\lambda_k : \min_{0 \leq \lambda \leq \bar{\lambda}} f(\mathbf{x}^{(k)} + \lambda \mathbf{d}^{(k)})$$

此处

$$\bar{\lambda} = \max\{\lambda | g_j(\mathbf{x}^{(k)} + \lambda \mathbf{d}^{(k)}) \geq 0, \quad j = 1, 2, \dots, l\}$$

6. 令

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda_k \mathbf{d}^{(k)}$$

$$k := k + 1$$

转回第 2 步。

例 12.2.1. 用可行方向法解下述约束优化问题

$$\begin{cases} \max \bar{f}(\mathbf{x}) = 4x_1 + 4x_2 - x_1^2 - x_2^2 \\ x_1 + 2x_2 \leq 4 \end{cases}$$

解. 先将该约束优化问题写成

$$\begin{cases} \max f(\mathbf{x}) = 4x_1 + 4x_2 - x_1^2 - x_2^2 \\ g_1(\mathbf{x}) = -x_1 - 2x_2 \geq 4 \end{cases}$$

取初始可行点 $\mathbf{x}^{(0)} = (0, 0)^T, f(\mathbf{x}^{(0)}) = 0$

$$\begin{aligned} \nabla f(\mathbf{x}) &= \begin{pmatrix} 2x_1 - 4 \\ 2x_2 - 4 \end{pmatrix}, \quad \nabla f(\mathbf{x}^{(0)}) = \begin{pmatrix} -4 \\ -4 \end{pmatrix} \\ \nabla g_1(\mathbf{x}) &= (-1, -2)^T \end{aligned}$$

$g_1(\mathbf{x}^{(0)}) = 4 > 0$, 从而 $J(\mathbf{x}^{(0)}) = \emptyset$ (空集)。由于

$$\|\nabla f(\mathbf{x}^{(0)})\|^2 = (-4)^2 + (-4)^2 = 32$$

所以 $\mathbf{x}^{(0)}$ 不是 (近似) 极小点。现取搜索方向

$$\mathbf{d}^{(0)} = -\nabla f(\mathbf{x}^{(0)}) = (4, 4)^T$$

从而

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda \mathbf{d}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} 4 \\ 4 \end{pmatrix} = \begin{pmatrix} 4\lambda \\ 4\lambda \end{pmatrix}$$

将其代入约束条件, 并令 $g_1(\mathbf{x}^{(1)}) = 0$, 解得 $\bar{\lambda} = 1/3$ 。

$$f(\mathbf{x}^{(1)}) = -16\lambda - 16\lambda + 16\lambda^2 + 16\lambda^2 = 32\lambda^2 - 32\lambda$$

令 $f(\mathbf{x}^{(1)})$ 对 λ 的导数等于零, 解得 $\lambda = 1/2$ 。因 λ 大于 $\bar{\lambda} (\bar{\lambda} = 1/3)$, 故取 $\lambda_0 = \bar{\lambda} = 1/3$ 。

$$\mathbf{x}^{(1)} = \left(\frac{4}{3}, \frac{4}{3} \right)^T, \quad f(\mathbf{x}^{(1)}) = -\frac{64}{9}$$

$$\nabla f(\mathbf{x}^{(1)}) = \left(-\frac{4}{3}, -\frac{4}{3} \right)^T, \quad g_1(\mathbf{x}^{(1)}) = 0$$

现构成下述线性规划问题

$$\begin{cases} \min \eta \\ -\frac{4}{3}d_1 - \frac{4}{3}d_2 \leq \eta \\ d_1 + 2d_2 \leq \eta \\ -1 \leq d_1 \leq 1, \quad -1 \leq d_2 \leq 1 \end{cases}$$

从而得到, $\eta = -4/10$, 搜索方向

$$\mathbf{d}^{(1)} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} 1.0 \\ -0.7 \end{pmatrix}$$

由此

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \lambda \mathbf{d}^{(1)} = \begin{pmatrix} 4/3 + \lambda \\ 4/3 - 0.7\lambda \end{pmatrix}$$

$$f(\mathbf{x}^{(2)}) = 1.49\lambda^2 - 0.4\lambda - 7.111$$

令 $\frac{df(\mathbf{x}^{(2)})}{d\lambda} = 0$, 得到 $\lambda = 0.134$ 。现暂用该步长, 算出

$$\mathbf{x}^{(2)} = \begin{pmatrix} 4/3 + 0.134 \\ 4/3 - 0.7 \times 0.134 \end{pmatrix} = \begin{pmatrix} 1.467 \\ 1.239 \end{pmatrix}$$

因 $g_1(\mathbf{x}^{(2)}) = 0.055 > 0$, 上面算出的 $\mathbf{x}^{(2)}$ 为可行点, 说明选取 $\lambda_1 = 0.134$ 正确。

继续迭代下去, 可得最优解为 $\mathbf{x}^* = (1.6, 1.2)^T$, $f(\mathbf{x}^*) = -7.2$ 。

原来问题的最优解不变, 其目标函数值

$$\bar{f}(\mathbf{x}^*) = -f(\mathbf{x}^*) = 7.2$$

12.2.2 制约函数法

本节介绍求解约束优化问题的制约函数法。使用这种方法，可将约束优化问题的求解，转化为求解一系列无约束优化问题，因而也称这种方法为无约束极小化技术，简记为 SUMT(sequential unconstrained minimization technique)。常用的制约函数基本上有两类：一位惩罚函数（或称罚函数(penalty function)）；一为障碍函数(barrier function)，对应于这两种函数，SUMT 有外点法和内点法。

外点法

考虑约束优化问题式(12.56)，为求其最优解，构造一个函数 $\psi(t)$

$$\psi(t) = \begin{cases} 0, & \text{当 } t \geq 0 \\ \infty, & \text{当 } t < 0 \end{cases} \quad (12.66)$$

现把 $g_j(\mathbf{x})$ 视为 t ，显然

当 \mathbf{x} 满足约束条件时， $\psi(g_j(\mathbf{x})) = 0, \quad j = 1, 2, \dots, l;$

当 \mathbf{x} 不满足约束条件时， $\psi(g_j(\mathbf{x})) = \infty$ 。

再构造函数

$$\varphi(\mathbf{x}) = f(\mathbf{x}) + \sum_{j=1}^l \psi(g_j(\mathbf{x})) \quad (12.67)$$

现求解无约束问题

$$\min \varphi(\mathbf{x}) \quad (12.68)$$

若该问题有解，假定其解为 \mathbf{x}^* ，则由式(12.66)应有 $\psi(g_j(\mathbf{x}^*)) = 0$ 。这就是说点 \mathbf{x}^* 满足约束条件。因而， \mathbf{x}^* 不仅是问题式(12.68)的极小解，它也是原问题式(12.56)的极小解。这样一来，就把有约束问题式(12.56)的求解化成了求解无约束问题式(12.68)。

但是，用上述方法构造的函数 $\psi(t)$ 在 $t = 0$ 处不连续，更没有导数。为此，将该函数修改为

$$\psi(t) = \begin{cases} 0, & \text{当 } t \geq 0 \\ t^2, & \text{当 } t < 0 \end{cases} \quad (12.69)$$

修改后的函数 $\psi(t)$ ，当 $t = 0$ 时导数等于零，而且 $\psi(t)$ 和 $\psi'(t)$ 对任意 t 都连续。当 \mathbf{x} 满足约束条件时仍有

$$\sum_{j=1}^l \psi(g_j(\mathbf{x})) = 0$$

当 \mathbf{x} 不满足约束条件时

$$0 < \sum_{j=1}^l \psi(g_j(\mathbf{x})) < \infty$$

我们取一个充分大的数 $M > 0$, 将 $\varphi(\mathbf{x})$ 改为

$$P(\mathbf{x}, M) = f(\mathbf{x}) + M \sum_{j=1}^l \psi(g_j(\mathbf{x})) \quad (12.70)$$

或等价地

$$P(\mathbf{x}, M) = f(\mathbf{x}) + M \sum_{j=1}^l [\min(0, g_j(\mathbf{x}))]^2 \quad (12.71)$$

从而可使 $\min P(\mathbf{x}, M)$ 的解 $\mathbf{x}(M)$ 为原问题的极小解或近似极小解。若求得的 $\mathbf{x}(M)$ 满足约束条件, 则它必定是原问题的极小解。事实上, 对于所有满足约束条件的 \mathbf{x}

$$\begin{aligned} f(\mathbf{x}) + M \sum_{j=1}^l \psi(g_j(\mathbf{x})) &= P(\mathbf{x}, M) \\ &\geq P(\mathbf{x}(M), M) = f(\mathbf{x}(M)) \end{aligned}$$

即当 \mathbf{x} 满足约束条件时, 有 $f(\mathbf{x}) \geq f(\mathbf{x}(M))$ 。

函数 $P(\mathbf{x}, M)$ 称为惩罚函数, 其中的第二项 $M \sum_{j=1}^l \psi(g_j(\mathbf{x}))$ 称惩罚项。图12.14示出了这种惩罚项的例子, 图中左半部表示约束条件 $g(\mathbf{x}) = x - a \geq 0$ 的情形, 右半部则表示 $g(x) = b - x \geq 0$ 的情形。

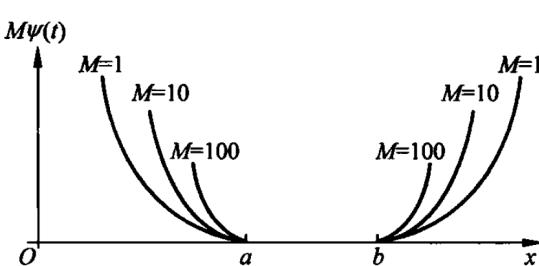


图 12.14

若对于某一个(惩)罚因子 M , 例如说 M_1 , $\mathbf{x}(M_1)$ 不满足约束条件, 就加大罚因子的值, 随着 M 值得增加, 惩罚函数中的惩罚项所起的作用随之增大, $\min P(\mathbf{x}, M)$ 的解 $\mathbf{x}(M)$ 与约束集的“距离”就越来越近, 当

$$0 < M_1 < M_2 < \cdots < M_k < \cdots$$

趋于无穷大时, 点列 $\{\mathbf{x}(M_k)\}$ 就从可行域的外部趋于原问题式(12.56)的极小点 \mathbf{x}_{\min} (此处假设点列 $\{\mathbf{x}(M_k)\}$ 收敛)。

可对外点法作如下经济解释: 把目标函数 $f(\mathbf{x})$ 看成“价格”, 约束条件看成某种“规定”, 采购人可在规定范围内购置最便宜的东西。此外对违反规定制定了一种“罚款”政策, 若符合规定, 罚款为零; 否则, 要收罚款。此外, 采购人付出的总代价应是价格和罚款的总和。采购者

的目标是使总代价最小，这就是上述的无约束问题。当罚款规定得很苛刻时，违反规定支付的罚款很高，这就迫使采购人符合规定。在数学上表现为罚因子 M_k 足够大时，上述无约束问题的最优解应满足约束条件，而成为约束条件的最优解。

外点法的迭代步骤如下：

1. 取 $M_1 > 0$ （例如说取 $M_1 = 1$ ），允许误差 $\varepsilon > 0$ ，并令 $k := 1$ ；
2. 求无约束优化问题的最优解：

$$\min_{\mathbf{x}} P(\mathbf{x}, M_k) = P(\mathbf{x}^{(k)}, M_k)$$

式中

$$P(\mathbf{x}, M_k) = f(\mathbf{x}) + M_k \sum_{j=1}^l [\min(0, g_j(\mathbf{x}))]^2$$

3. 若对某一个 $j (1 \leq j \leq l)$ 有

$$-g_j(\mathbf{x}^{(k)}) \geq \varepsilon$$

则取 $M_{k+1} > M_k$ （例如， $M_{k+1} = cM_k$ ， $c = 5$ 或 10 ），令 $k_t := k + 1$ ，并转向第 2 步。否则，停止迭代，得

$$\mathbf{x}_{\min} \approx \mathbf{x}^{(k)}$$

例 12.2.2. 求解约束优化问题

$$\begin{cases} \min f(\mathbf{x}) = x_1 + x_2 \\ g_1(\mathbf{x}) = -x_1^2 + x_2 \geq 0 \\ g_2(\mathbf{x}) = x_1 \geq 0 \end{cases}$$

解：构造罚函数

$$P(\mathbf{x}, M) = x_1 + x_2 + M \{ [\min(0, (-x_1^2 + x_2))]^2 + [\min(0, x_1)]^2 \}$$

$$\frac{\partial P}{\partial x_1} = 1 + 2M[\min(0, (-x_1^2 + x_2)(-2x_1))] + 2M[\min(0, x_1)]$$

$$\frac{\partial P}{\partial x_2} = 1 + 2M[\min(0, (-x_1^2 + x_2))]$$

对于不满足约束条件的点 $\mathbf{x} = (x_1, x_2)^T$ ，有

$$-x_1^2 + x_2 < 0, \quad x_1 < 0$$

令

$$\frac{\partial P}{\partial x_1} = \frac{\partial P}{\partial x_2} = 0$$

得 $\min P(\mathbf{x}, M)$ 的解为

$$\mathbf{x}(M) = \left(-\frac{1}{2(1+M)}, \left(\frac{1}{4(1+M)^2} - \frac{1}{2M} \right) \right)^T$$

取 $M = 1, 2, 3, 4$, 可得出以下结果:

$$M = 1 : \mathbf{x} = (-1/4, -7/16)^T$$

$$M = 2 : \mathbf{x} = (-1/6, -2/9)^T$$

$$M = 3 : \mathbf{x} = (-1/8, -29/192)^T$$

$$M = 4 : \mathbf{x} = (-1/10, -23/200)^T$$

可知 $\mathbf{x}(M)$ 从约束条件外面逐步逼近约束条件的边界, 当 $M \rightarrow \infty$ 时, $\mathbf{x}(M)$ 趋于原问题的极小解 $\mathbf{x}_{\min} = (0, 0)^T$ (见图 12.15)。

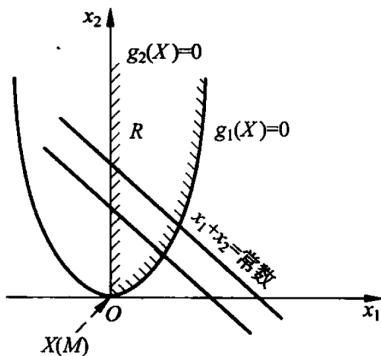


图 12.15

以上叙述说明, 外点法的一个重要特点, 就是函数 $P(\mathbf{x}, M)$ 是在 \mathbb{R}^n 上进行优化, 初始点可任意选择, 这给计算带来了很大方便。而且外点法也可用于非凸函数的最优化。

最后还要指出, 外点法不只适用于含有不等式约束条件的约束优化问题, 对于等式约束条件或同时含有等式和不等式约束条件的问题也同样适用。此外, 惩罚函数也可以采用其它形式。

内点法

如果要求每次迭代得到的近似解都在可行域内, 以便观察目标函数值的变化情况 (有时可能需要这样); 或者, 如果 $f(\mathbf{x})$ 在可行域外的性质比较复杂, 甚至没有定义, 这时就无法使用外点法。

内点法和外点法不同, 它要求迭代过程始终在可行域内部进行。为此, 我们把初始点取在可行域内部 (即既不在可行域外, 也不在可行域边界上), 这种可行点称为内点或严格内点, 并

在可行域的边界上设置一道“障碍”，使迭代点靠近可行域的边界时，给出的新目标函数值迅速增大，从而使迭代点始终留在可行域内部。

我们仿照外点法，通过函数叠加的办法来改造原目标函数，使得改造后的目标函数（称为**障碍函数**）具有这种性质：在可行域的内部与其边界较远的地方，障碍函数与原来的目标函数 $f(\mathbf{x})$ 尽可能相近；而在接近可行域的边界时可以有任意大的值。可以想见，满足这种要求的障碍函数，其极小解自然不会在可行域的边界上达到。这就是说，用障碍函数来代替（近似）原目标函数，并在可行域内部使其极小化，虽然可行域是一个闭集，但因极小点不在闭集的边界上，因而实际上是具有无约束性质的优化问题，可借助于无约束优化的方法进行计算。

根据上述分析，即可将约束优化式(12.56)转化为下述一系列无约束性质的极小化问题：

$$\min_{\mathbf{x} \in R_0} \bar{P}(\mathbf{x}, r_k) \quad (12.72)$$

其中

$$\bar{P}(\mathbf{x}, r_k) = f(\mathbf{x}) + r_k \sum_{j=1}^l \frac{1}{g_j(\mathbf{x})}, \quad (r_k > 0) \quad (12.73)$$

或

$$\bar{P}(\mathbf{x}, r_k) = f(\mathbf{x}) - r_k \sum_{j=1}^l \log(g_j(\mathbf{x})), \quad (r_k > 0) \quad (12.74)$$

$$R_0 = \{\mathbf{x} | g_j(\mathbf{x}) > 0, \quad j = 1, 2, \dots, l\} \quad (12.75)$$

式(12.73)和式(12.74)右端第二项称为**障碍项**。易见，在可行域的边界上（即至少有一个 $g_j(\mathbf{x}) = 0$ ）， $\bar{P}(\mathbf{x}, r_k)$ 为正无穷大。

如果从可行域内部的某一点 $\mathbf{x}^{(0)}$ 出发，按无约束极小化方法对式(12.72)进行迭代（在进行一维搜索时要使用控制步长，以免迭代点跑到 R_0 之外），则随着**障碍因子** r_k 的逐步减小，即

$$r_1 > r_2 > \dots > r_k > \dots > 0$$

障碍项所起的作用也越来越小，因而，求出的 $\min \bar{P}(\mathbf{x}, r_k)$ 的解 $\mathbf{x}(r_k)$ 也逐步逼近原问题式(12.56)的极小解 \mathbf{x}_{\min} 。若原来问题的极小解在可行域的边界上，则随着 r_k 减小，障碍作用逐步降低，所求出的障碍函数的极小解就会不断靠近边界，直至满足某一精度要求为止。

内点法的迭代步骤如下：

1. 取 $r_1 > 0$ （例如取 $r_1 = 1$ ），允许误差 $\varepsilon > 0$ ；
2. 找出一可行点 $\mathbf{x}^{(0)} \in R_0$ ，并令 $k = 1$ ；
3. 构造障碍函数，障碍项可采用倒数函数（式(12.73)），也可采用对数函数（例如式(12.74)）；
4. 以 $\mathbf{x}^{(k-1)} \in R_0$ 为初始点，对障碍函数进行无约束极小化（在 R_0 内）：

$$\begin{cases} \min_{\mathbf{x} \in R_0} \bar{P}(\mathbf{x}, r_k) = \bar{P}(\mathbf{x}^{(k)}, r_k) \\ \mathbf{x}^{(k)} = \mathbf{x}(r_k) \in R_0 \end{cases} \quad (12.76)$$

式中 $\bar{P}(\mathbf{x}, r_k)$ 见式(12.73)或(12.74)；

5. 检验是否满足收敛准则

$$r_k \sum_{j=1}^l \frac{1}{g_j(\mathbf{x}^{(k)})} \leq \varepsilon$$

或

$$\left| r_k \sum_{j=1}^l \log(g_j(\mathbf{x}^{(k)})) \right| \leq \varepsilon$$

如满足上述准则，则以 $\mathbf{x}^{(k)}$ 为原问题的近似极小解 \mathbf{x}_{\min} ；否则，取 $r_{k+1} < r_k$ （例如取 $r_{k+1} = r_k/10$ 或 $r_k/5$ ），令 $k := k + 1$ ，转向第 3 步继续进行迭代。

值得指出的是，根据情况，收敛准则也可采用不同的形式，例如：

$$\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| < \varepsilon$$

或

$$\|f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k-1)})\| < \varepsilon$$

例 12.2.3. 试用内点法求解

$$\begin{cases} \min f(\mathbf{x}) = \frac{1}{3}(x_1 + 1)^3 + x_2 \\ g_1(\mathbf{x}) = x_1 - 1 \geq 0 \\ g_2(\mathbf{x}) = x_2 \geq 0 \end{cases}$$

解. 构造障碍函数

$$\bar{P}(\mathbf{x}, r) = \frac{1}{3}(x_1 + 1)^3 + x_2 + \frac{r}{x_1 - 1} + \frac{r}{x_2}$$

$$\frac{\partial \bar{P}}{\partial x_1} = (x_1 + 1)^2 - \frac{r}{(x_1 - 1)^2} = 0$$

$$\frac{\partial \bar{P}}{\partial x_2} = 1 - \frac{r}{x_2^2} = 0$$

联立解上述两个方程，得

$$x_1(r) = \sqrt{1 + \sqrt{r}}, \quad x_2(r) = \sqrt{r}$$

如此得最优解：

$$\mathbf{x}_{\min} = \lim_{r \rightarrow 0} \left(\sqrt{1 + \sqrt{r}}, x_2(r) = \sqrt{r} \right)^T = (1, 0)^T$$

由此例可解析求解，故可如上进行。但很多问题不便用解析法，而需用迭代法求解。

例 12.2.4. 使用内点法解

$$\begin{cases} \min f(\mathbf{x}) = x_1 + x_2 \\ g_1(\mathbf{x}) = -x_1^2 + x_2 \geq 0 \\ g_2(\mathbf{x}) = x_1 \geq 0 \end{cases}$$

传外切解法
内点法
高数
草稿

解。障碍项采用自然对数函数，得障碍函数如下：

$$\bar{P}(\mathbf{x}, r) = x_1 + x_2 - r \log(-x_1^2 + x_2) - r \log x_1$$

各次迭代结果示于表12.3和图12.16。

障碍因子	r	$x_1(r)$	$x_2(r)$
r_1	1.000	0.500	1.250
r_2	0.500	0.309	0.595
r_3	0.250	0.183	0.283
r_4	0.100	0.085	0.107
r_5	0.0001	0.000	0.000

表 12.3

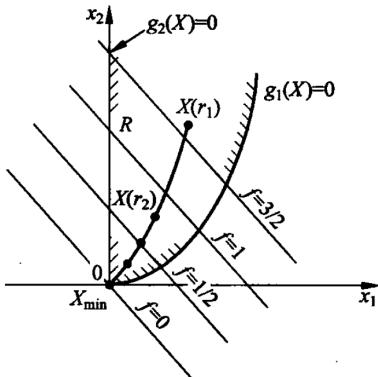


图 12.16

我们知道，内点法的迭代过程必须由某个内点开始。在处理实际问题时，如果不能找出某个内点作为初始点，迭代就无法展开。下面说明初始内点的求法。求初始内点本身也是一个迭代过程。

先任找一点 $\mathbf{x}^{(0)}$ 为初始点，令

$$S_0 = \{j | g_j(\mathbf{x}^{(0)}) \leq 0, \quad 1 \leq j \leq l\}$$

$$T_0 = \{j | g_j(\mathbf{x}^{(0)}) > 0, \quad 1 \leq j \leq l\}$$

如果 S_0 为空集，则 $\mathbf{x}^{(0)}$ 为初始内点；若 S_0 非空，则以 S_0 中的约束函数为假拟目标函数，并以 T_0 中的约束函数为障碍项，构成一无约束优化问题，对这一问题进行极小化，可得一个新点

$\mathbf{x}^{(1)}$ 。然后检验 $\mathbf{x}^{(1)}$ 是否为内点，若仍不为内点，如上继续进行，并减小障碍因子 r ，直到求出一个内点为止。

求初始内点的迭代步骤如下：

1. 任取一点 $\mathbf{x}^{(0)}, r_0 > 0$ （例如 $r_0 = 1$ ），令 $k = 0$ ；

2. 定出指标集 S_k 及 T_k

$$S_k = \{j | g_j(\mathbf{x}^{(k)}) \leq 0, \quad 1 \leq j \leq l\}$$

$$T_k = \{j | g_j(\mathbf{x}^{(k)}) > 0, \quad 1 \leq j \leq l\}$$

3. 检查集合 S_k 是否为空集，若为空集，则 $\mathbf{x}^{(k)}$ 在 R_0 内，初始内点找到，迭代停止，否则转向第 4 步；

4. 构造函数

$$\tilde{P}(\mathbf{x}, r_k) = - \sum_{j \in S_k} g_j(\mathbf{x}) + r_k \sum_{j \in T_k} \frac{1}{g_j(\mathbf{x})}, \quad (r_k > 0)$$

以 $\mathbf{x}^{(k)}$ 为初始点，在保持对集合

$$\tilde{R}_k = \{\mathbf{x} | g_j(\mathbf{x}) > 0, \quad j \in T_k\}$$

可行的情况下，极小化 $\tilde{P}(\mathbf{x}, r_k)$ ，即

$$\min \tilde{P}(\mathbf{x}, r_k), \quad \mathbf{x} \in \tilde{R}_k$$

得 $\mathbf{x}^{(k+1)}, \mathbf{x}^{(k+1)} \in \tilde{R}_k$ ，转向第 5 步；

5. 令 $0 < r_{k+1} < r_k$ （比如说 $r_{k+1} = r_k/10, k := k + 1$ ），转向第 2 步。

12.3 深度学习常用优化算法

深度学习算法在许多情况下都涉及到优化算法，但是用于深度模型训练的优化算法与传统的优化算法在几个方面有所不同。在大多数机器学习问题中，我们关注的点是在测试集上的不可解的性能度量 P 。因此，我们只是间接地优化 P 。我们希望通过降低代价函数 $J(\theta)$ 来提高 P ，这一点不同于纯优化最小化 J 本身。训练深度模型的优化算法通常也会包括一些用于机器学习目标函数特定结构上的特殊优化。

通常，代价函数可写为训练集上的平均，如

$$J(\theta) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \hat{P}_{data}} L(f(\mathbf{x}; \theta), \mathbf{y}) \tag{12.77}$$

其中， L 是每个样本的损失函数， $f(\mathbf{x}; \theta)$ 是输入为 \mathbf{x} 时所预测的输出， \hat{P}_{data} 是经验分布，监督学习中， \mathbf{y} 是目标输出。在本节中，我们只介绍不带正则化的监督学习，即 L 的变量是 $f(\mathbf{x}; \theta)$ 和 \mathbf{y} 。

12.3.1 随机梯度下降

机器学习算法和一般优化算法不同的一点是，机器学习算法的目标函数通常可以分解为训练样本上对的求和。机器学习中的优化算法在计算参数的每一次更新时通常基于使用整个代价函数中仅仅一部分项来估计代价函数的期望值。

例如，最大似然估计问题可以在对数空间中分解成每个样本的总和：

$$\boldsymbol{\theta}_{ML} = \arg \max_{\boldsymbol{\theta}} \sum_{i=1}^m \log_{P_{model}}(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}; \boldsymbol{\theta}) \quad (12.78)$$

最大化这个总和等价于最大化训练集在经验分布上的期望：

$$J(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \hat{P}_{data}} \log_{P_{model}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) \quad (12.79)$$

优化算法用到的目标函数 J 中大多数性质也是训练集上的期望。例如，最常用的性质是梯度：

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \hat{P}_{data}} \nabla_{\boldsymbol{\theta}} \log_{P_{model}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) \quad (12.80)$$

准确计算这个期望的计算量非常大，因为需要在整个数据集上的每个样本上评估模型。在实践中，我们可以从数据集中随机采样少量的样本，然后计算这些样本上的平均值，这么做有两个好处：

首先， n 个样本均值的标准差是 σ/\sqrt{n} ，其中 σ 是样本值真实的标准差，分母 \sqrt{n} 表明使用更多样本来估计梯度的方法是低于线性的。换句话说，一个基于 100 个样本，另一个基于 10,000 个样本，后者用于梯度计算的计算量是前者的 100 倍，但却只降低了 10 倍的均值标准差。如果能够快速计算出梯度估计值，而不是缓慢计算准确值，那么大多数优化算法会收敛地更快（就总的计算量而言，而不是指更新次数）。

另一个从小数目样本中获得梯度的统计估计的动机是训练集的冗余。在最坏情况下，训练集中所有 m 个样本可以是彼此相同的拷贝。基于采样的梯度估计可以使用单个样本计算出正确的梯度，而比原来的做法少花了 m 倍时间。实践中，我们不太可能真的遇到这种最坏情况，但我们可能会发现大量样本都对梯度做出了非常相似的贡献。

使用整个训练集的优化算法被称为 **batch** 或 **确定性梯度算法**，因为它们会同时在大 batch 中处理所有的样本。每次只使用单个样本的优化算法有时被称为 **随机** 或者 **在线** 算法。其中“在线”通常是指从连续产生的数据流中抽取样本的情况，而不是从一个固定大小的训练集中遍历多次采样的情况。

大多数用于深度学习的算法介于以上两者之间，使用一个以上，而又不是全部的训练样本。传统上，这些会被称为 **minibatch** 或 **minibatch** 随机方法，通常将它们简单地称为 **随机** 方法。

随机方法的典型示例是随机梯度下降 (SGD)，其算法概括如下：

要求： 学习速率 ϵ_k ，初始参数 $\boldsymbol{\theta}$

while 没有达到停止准则 **do**

 从训练集中采包含 m 个样本 $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ 的 minibatch，对应目标为 $\mathbf{y}^{(i)}$ ；

计算梯度估计: $\hat{\mathbf{g}} \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(\mathbf{x}^{(i)}; \theta), \mathbf{y}^{(i)})$

应用更新: $\theta \leftarrow \theta - \epsilon_k \hat{\mathbf{g}}$

end while

SGD 算法中的一个关键参数是学习速率。在实践中, 随着时间的推移有必要逐渐降低学习速率, 在第 k 次迭代得学习速率我们记为 ϵ_k 。

逐步降低学习速率的原因是 SGD 在梯度估计引入的噪源 (m 个训练样本的随机采样) 并不会在极小值处消失。相比之下, 当我们使用 batch 梯度下降到达极小值时, 整个代价函数的真实梯度会变得很小, 甚至为 **0**, 因此 batch 梯度下降可以使用固定的学习速率。保证 SGD 收敛的一个充分条件是

$$\sum_{k=1}^{\infty} \epsilon_k = \infty \quad (12.81)$$

且

$$\sum_{k=1}^{\infty} \epsilon_k^2 < \infty \quad (12.82)$$

实践中, 一般会线性衰减学习速率到第 τ 次迭代:

$$\epsilon_k = (1 - \alpha)\epsilon_0 + \alpha\tau_r \quad (12.83)$$

其中, $\alpha = \frac{k}{\tau}$ 。在 τ 步迭代之后, 一般使 ϵ 保持常数。

学习速率可通过试验和误差来选取, 通常最好的选择方法是画出目标函数值随时间变化的学习曲线。使用线性时间表时, 参数选择为 $\epsilon_0, \epsilon_\tau, \tau$ 。通常 τ 被设为需要反复遍历训练样本几百次的迭代次数, 且设为大于 1% 的 ϵ_0 。所以主要问题是如何设置 ϵ_0 。若 ϵ_0 太大, 学习曲线将会剧烈振荡, 代价函数值通常会明显增加。温和的振荡是良好的, 特别是训练于随即代价函数上, 例如由信号丢失引起的代价函数。如果学习速率太慢, 那么学习进程会缓慢。如果初始学习速率太低, 那么学习可能会卡在一个相当高的损失值。通常, 就总训练时间和最终损失值而言, 最优初始学习速率会高于大约迭代 100 步后输出最好效果的学习速率。因此, 通常最好是检测最早的几次迭代, 使用一个高于此时效果最佳学习速率的学习速率, 但又不能太高以致严重的不稳定性。

SGD 和相关的 minibatch 或在线基于梯度的优化的最重要性质是每一步更新的计算时间不会随着训练样本数目而增加。即使训练样本数目非常大时, 这也能收敛。对于足够大的数据集, SGD 可能会在处理整个训练集之前就收敛到最终测试集误差的某个固定容差范围内。

研究优化算法的收敛率, 一般会衡量额外误差 $J(\theta) - \min_{\theta} J(\theta)$, 即当前代价函数超出最低可能损失的量。SGD 应用于凸问题时, k 步迭代后的额外误差量级是 $O(\frac{1}{\sqrt{k}})$, 在强凸情况下是 $O(\frac{1}{k})$ 。除非假定额外的条件, 否则这些界限不能进一步改进。batch 梯度下降在理论上比随机梯度下降有更好的收敛率, 然而, Carmér 界限 (Carmér, 1946; Rao, 1945) 指出, 泛化误差的下降速度不会快于 $O(\frac{1}{k})$ 。Bottou 和 Bousquet(2008) 由此认为对于机器学习任务, 不值得探寻收敛快于

$O(\frac{1}{k})$ 的优化算法——更快的收敛可能对应着过拟合。此外，渐进分析掩盖了随机梯度下降在少量更新步之后的很多优点。对于大数据集，SGD 初始快速更新只需非常少量样本计算梯度的能力远远超过了其缓慢的渐进收敛。我们也可以权衡 batch 梯度下降和随机梯度下降两者的特点，在学习过程中逐渐增大 minibatch 的大小。

12.3.2 动量梯度下降

虽然随机梯度下降仍然是非常受欢迎的优化方法，但学习速率有时会很慢。动量方法 (Polyak, 1964) 旨在加速学习，特别是处理高曲率，小但一致的梯度，或是带噪扰的梯度。动量算法积累了之前梯度指数级衰减的移动平均，并且继续沿该方向移动。动量的效果如图 12.17 所示。

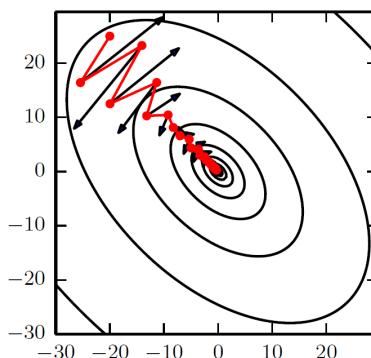


图 12.17

动量的主要目的是解决两个问题：Hessian 矩阵的不良条件数和随机梯度的方差。我们通过图 12.17 说明动量如何克服第一个问题。轮廓线描绘了一个二次损失函数（具有不良条件数的 Hessian 矩阵）。横跨轮廓的红色路径表示动量学习规则所遵循的路径，它使该函数最小化。我们在每个步骤画一个箭头，指示梯度下降将在该点采取的步骤。可以看到，一个条件数较差的二次目标函数看起来像一个长而窄的山谷或陡峭的峡谷。动量正确地纵向穿过峡谷，而梯度步骤则会浪费时间在峡谷的窄轴上来回移动。

从形式上看，动量算法引入了变量 v 充当速度的角色——它代表参数在参数空间移动的方向和速度。速度被设为负梯度的指数衰减平均。名称 **动量** (momentum) 来自物理类比，根据牛顿运动定律，负梯度是移动参数空间中粒子的力。动量在物理学上是质量乘以速度。在动量学习算法中，我们假设是单位质量，因此速度向量 v 也可以看作是粒子的动量。超参数 $\alpha \in [0, 1)$ 决

定了之前梯度的贡献衰减得有多快。其更新规则如下：

$$\begin{cases} \mathbf{v} \leftarrow \alpha \mathbf{v} - \epsilon \nabla_{\theta} \left(\frac{1}{m} \sum_{i=1}^m L(f(\mathbf{x}^{(i)}; \theta), \mathbf{y}^{(i)}) \right) \\ \theta \leftarrow \theta + \mathbf{v} \end{cases} \quad (12.84)$$

速度 \mathbf{v} 累计了梯度元素 $\nabla_{\theta} \left(\frac{1}{m} \sum_{i=1}^m L(f(\mathbf{x}^{(i)}; \theta), \mathbf{y}^{(i)}) \right)$ 。相对于 ϵ 的 α 越大，之前梯度对现在方向的影响也越大。

带动量的 SGD 算法如下所示：

要求： 学习速率 ϵ , 动量参数 α , 初始参数 θ , 初始速度 \mathbf{v}

while 没有达到停止准则 **do**

从训练集中采包含 m 个样本 $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ 的 minibatch, 对应目标为 $\mathbf{y}^{(i)}$;

计算梯度估计: $\mathbf{g} \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(\mathbf{x}^{(i)}; \theta), \mathbf{y}^{(i)})$

计算速度更新: $\mathbf{v} \leftarrow \alpha \mathbf{v} - \epsilon \mathbf{g}$

应用更新: $\theta \leftarrow \theta + \mathbf{v}$

end while

之前，步长只是梯度范数乘以学习速率（这里的“步长”与传统优化算法的“步长”概念不同）。现在，步长取决于梯度序列的大小和排列。当许多连续的梯度指向相同的方向时，步长最大。如果动量动量算法总是观测到梯度 \mathbf{g} ，那么它会在方向 $-\mathbf{g}$ 上不停加速，直到达到最后速度的步长为

$$\frac{\epsilon \|\mathbf{g}\|}{1 - \alpha} \quad (12.85)$$

因此将动量的超参数视为 $\frac{1}{1-\alpha}$ 有助于理解。例如， $\alpha = 0.9$ 对应着最大速度 10 倍于梯度下降算法。

在实践中， α 的一般取值为 0.5, 0.9 和 0.99。和学习速率一样， α 也会随着时间变化。一般初始值是一个较小的值，随后会慢慢变大。随着时间推移改变 α 没有收缩 ϵ 更重要。

我们可以将动量算法视为模拟连续时间下牛顿动力学下的粒子。这种物理类比有助于直觉上理解动量和梯度下降算法是如何表现的。

粒子在任意时间点的位置由 $\theta(t)$ 给定。粒子会受到净力 $\mathbf{f}(t)$ 。该力会导致粒子加速:

$$\mathbf{f}(t) = \frac{\partial^2}{\partial t^2} \theta(t) \quad (12.86)$$

与其将其视为位置的二阶微分方程，我们不如引入表示粒子在时间 t 处速度的变量 $\mathbf{v}(t)$ ，将牛顿动力学重写为一阶微分方程:

$$\mathbf{v}(t) = \frac{\partial}{\partial t} \theta(t) \quad (12.87)$$

$$\mathbf{f}(t) = \frac{\partial}{\partial t} \mathbf{v}(t) \quad (12.88)$$

由此，动量算法包括通过数值模拟求解微分方程。求解微分方程的一个简单数值方法是欧拉方法，通过在每个梯度方向上小且有限的步来简单模拟该等式定义的动力学。

这解释了动量更新的基本形式，但具体是哪些力呢？一个力正比于代价函数的负梯度 $-\nabla_{\theta} J(\theta)$ 。该力推动粒子沿着代价函数表面下坡的方向移动。梯度下降算法基于每个梯度简单地更新一步，而使用动量算法的牛顿方案则使用该力改变粒子的速度。我们可以将粒子视作在冰面上滑行的冰球。每当它沿着表面最陡的部分下降时，它会沿该方向加速滑行，直到开始向上滑动为止。

另一个力也是必要的。如果代价函数的梯度是唯一的力，那么粒子可能永远不会停下来。想象一下，假设理想情况下冰面没有摩擦，一个冰球从山谷的一端下滑，上升到另一端，永远来回振荡。要解决这个问题，我们添加另一个正比于 $-\mathbf{v}(t)$ 的力。在物理术语中，此力对应于粘性阻力，就像粒子必须通过一个抵抗介质，如糖浆。这会导致粒子随着时间推移逐渐失去能量，最终收敛到局部极小点。

为什么要特别使用 $-\mathbf{v}(t)$ 和粘性阻力呢？部分原因是因为 $-\mathbf{v}(t)$ 在数学上的便利——速度的整数幂很容易处理。然而，其它物理系统具有基于速度的其他整数幂的其他类型的阻力。例如，颗粒通过空气时会受到正比于速度平方的湍流阻力，而颗粒沿着地面移动时会受到恒定大小的摩擦力。这些选择都不合适。湍流阻力，正比于速度的平方，在速度很小时会很弱。不够强大到使粒子停下来。非零值初始速度的粒子仅受到湍流阻力，会从初始位置永远地移动下去，和初始位置的距离大概正比于 $O(\log t)$ 。因此我们必须使用速度较低幂次的力。如果幂次为零，相当于无摩擦，那么力太强了。当代价函数的梯度表示的力很小但非零时，由于摩擦导致的恒力会使得粒子在达到局部极小点之前就停下来。粘性阻力避免了这两个问题——它足够弱，可以使梯度引起的运动直到达到最小，但又足够强，使得坡度不够时可以阻止运动。

受 Nesterov 加速梯度算法启发，Sutskever(2013) 提出了动量算法的一个变种。这种情况的更新规则如下：

$$\begin{cases} \mathbf{v} \leftarrow \alpha \mathbf{v} - \epsilon \nabla_{\theta} \left[\frac{1}{m} \sum_{i=1}^m L(\mathbf{f}(\mathbf{x}^{(i)}; \theta + \alpha \mathbf{v}), \mathbf{y}^{(i)}) \right] \\ \theta \leftarrow \theta + \mathbf{v} \end{cases} \quad (12.89)$$

其中参数 α 和 ϵ 发挥了和标准动量方法中类似的作用。Nesterov 动量和标准动量之间的区别体现在梯度计算上。Nesterov 动量中，梯度计算在施加当前速度之后。因此，Nesterov 动量可以解释为往标准动量方法中添加了一个校正因子。

Nesterov 动量的随机梯度下降算法如下所示：

要求： 学习速率 ϵ ，动量参数 α ，初始参数 θ ，初始速度 \mathbf{v}

while 没有达到停止准则 **do**

 从训练集中采包含 m 个样本 $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ 的 minibatch，对应目标为 $\mathbf{y}^{(i)}$ ；

 应用临时更新： $\tilde{\theta} \leftarrow \theta + \alpha \mathbf{v}$

计算梯度 (在临时点): $\mathbf{g} \leftarrow \frac{1}{m} \nabla_{\tilde{\theta}} \sum_i L(f(\mathbf{x}^{(i)}; \tilde{\theta}), \mathbf{y}^{(i)})$

计算速度更新: $\mathbf{v} \leftarrow \alpha \mathbf{v} - \epsilon \mathbf{g}$

应用更新: $\theta \leftarrow \theta + \mathbf{v}$

end while

在凸 batch 梯度的情况下, Nesterov 动量将额外误差收敛率从 $O(1/k)$ (k 步后) 改进到 $O(1/k^2)$ 。可惜, 在随机梯度的情况下, Nesterov 动量没有改进收敛率。

12.3.3 自适应学习速率

神经网络研究员早就意识到学习速率肯定是最难以设置的超参数之一, 因为它对模型的性能有显著的影响。通常, 损失函数高度敏感于参数空间中的某些方向, 动量算法虽然可以在一定程度缓解这些问题, 但这样做的代价是引入了另一个超参数。在这种情况下, 自然会问有没有其他方法。如果我们相信方向敏感度在某种程度上是轴对齐的, 那么每个参数设置不同的学习速率, 在整个学习过程中自动适应这些学习速率是有道理的。

Delta-bar-delta 算法 (Jacobs, 1988) 是一个早期的在训练时适应模型参数各自学习速率的启发式方法。该方法基于一个很简单的想法, 如果损失对于某个给定模型参数的偏导保持相同的符号, 那么学习速率应该增加。如果对于该参数的偏导变化了符号, 那么学习速率应减小。当然, 这种方法只能应用于全 batch 优化中。

最近, 提出了一些增量 (或者基于 minibatch) 的算法来自适应模型参数的学习速率。这节将简要回顾其中一些算法。

AdaGrad

AdaGrad 算法独立地适应所有模型参数的学习速率, 按照每个参数的梯度历史值的平方和的平方根成反比缩放每个参数 (Duchi, 2011)。具有损失最大偏导的参数相应地有一个快速下降的学习速率, 而具有小偏导的参数在学习速率上有相对较小的下降。净效果是在参数空间中更为平缓的倾斜方向会取得更大的进步。

在凸优化背景中, AdaGrad 算法具有一些令人满意的理论性质。然而, 经验上已经发现, 对于训练深度神经网络模型而言, 从训练开始时积累梯度平方会导致有效学习速率过早和过量的减小。AdaGrad 在某些深度学习模型上效果不错, 但不是全部。

AdaGrad 算法如下所示:

要求: 全局学习速率 ϵ , 初始参数 θ , 小常数 δ (为了数值稳定大约设为 10^{-7})

初始化梯度累计变量 $\mathbf{r} = \mathbf{0}$;

while 没有达到停止准则 **do**

从训练集中采包含 m 个样本 $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ 的 minibatch, 对应目标为 $\mathbf{y}^{(i)}$;

计算梯度: $\mathbf{g} \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(\mathbf{x}^{(i)}; \theta), \mathbf{y}^{(i)})$

累积平方梯度: $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{g} \odot \mathbf{g}$ (\odot 表示逐元素相乘)

计算更新: $\Delta\theta \leftarrow -\frac{\epsilon}{\delta + \sqrt{r}} \odot \mathbf{g}$ (逐元素地应用除和求平方根)

应用更新: $\theta \leftarrow \theta + \Delta\theta$

end while

RMSProp

RMSProp 算法 (Hinton, 2012) 修改 AdaGrad 以在非凸设定下效果更好, 改变梯度积累为指数加权的移动平均。AdaGrad 旨在应用于凸问题时快速收敛。当应用于非凸函数训练神经网络时, 学习轨迹可能穿过了很多不同的结构, 最终到达一个局部是凸的碗状的区域。AdaGrad 根据平方梯度的整个历史收缩学习速率, 可能使得学习速率在达到这样的凸结构前就变得太小了。RMSProp 使用指数衰减平均以丢弃遥远过去的历史, 使其能够在找到碗状凸结构后快速收敛, 它就像一个初始化于该碗状结构的 AdaGrad 算法实例。

RMSProp 的标准形式如下所示, 相比于 AdaGrad, 使用移动平均引入了一个新的超参数 ρ , 用来控制移动平均的长度范围。

要求: 全局学习速率 ϵ , 衰减速率 ρ , 初始参数 θ , 小常数 δ (为了数值稳定大约设为 10^{-6})

初始化梯度累计变量 $\mathbf{r} = \mathbf{0}$;

while 没有达到停止准则 **do**

从训练集中采包含 m 个样本 $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ 的 minibatch, 对应目标为 $\mathbf{y}^{(i)}$;

计算梯度: $\mathbf{g} \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(\mathbf{x}^{(i)}; \theta), \mathbf{y}^{(i)})$

累积平方梯度: $\mathbf{r} \leftarrow \rho \mathbf{r} + (1 - \rho) \mathbf{g} \odot \mathbf{g}$

计算更新: $\Delta\theta \leftarrow -\frac{\epsilon}{\delta + \sqrt{r}} \odot \mathbf{g}$

应用更新: $\theta \leftarrow \theta + \Delta\theta$

end while

RMSProp 也可以结合 Nesterov 动量, 其形式如下所示:

要求: 全局学习速率 ϵ , 衰减速率 ρ , 动量系数 α

初始化梯度累计变量 $\mathbf{r} = \mathbf{0}$;

while 没有达到停止准则 **do**

从训练集中采包含 m 个样本 $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ 的 minibatch, 对应目标为 $\mathbf{y}^{(i)}$;

计算临时更新: $\tilde{\theta} \leftarrow \theta + \alpha v$
 计算梯度: $\mathbf{g} \leftarrow \frac{1}{m} \nabla_{\tilde{\theta}} \sum_i L(f(\mathbf{x}^{(i)}; \tilde{\theta}), \mathbf{y}^{(i)})$
 累积梯度: $\mathbf{r} \leftarrow \rho \mathbf{r} + (1 - \rho) \mathbf{g} \odot \mathbf{g}$
 计算速度更新: $v \leftarrow \alpha v - \frac{\epsilon}{\sqrt{r}} \odot \mathbf{g}$
 应用更新: $\theta \leftarrow \theta + v$

end while

经验上, RMSProp 已被证明是一种有效且实用的深度神经网络优化算法。目前它是深度学习从业者经常采用的优化方法之一。

Adam

Adam(kingma and Ba, 2014) 是另一种学习速率自适应的优化算法。“Adam”这个名字派生自短语“adaptive moments”。在前述算法背景下, 它也许最好被看作结合了 RMSProp 和动量的具有一些重要区别的变种。首先, 在 Adam 中, 动量直接并入了梯度一阶矩(指数加权)的估计。将动量加入 RMSProp 最直观的方法是将动量应用于缩放后的梯度。结合缩放的动量使用没有明确的理论动机。其次, Adam 包括偏置修正, 修正从原点初始化的一阶矩(动量项)和(非中心的)二阶矩的估计。RMSProp 也采用了(非中心的)二阶矩估计, 然而缺失了修正因子。因此, 不像 Adam, RMSProp 二阶矩估计可能在训练初期有很高的偏置。Adam 通常被认为对超参数的选择相当鲁棒, 尽管学习速率有时需要改为与建议的默认值不同的值。

Adam 算法如下所示:

要求: 步长 ϵ (建议默认为 0.001), 矩估计的指数衰减速率 ρ_1, ρ_2 (建议分别默认为 0.9 和 0.999), 用于数值稳定的小常数 δ (建议默认为 10^{-8}), 初始参数 θ

初始化一阶和二阶矩变量 $s = 0, r = 0$

初始化时间步 $t = 0$

while 没有达到停止准则 **do**

从训练集中采包含 m 个样本 $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ 的 minibatch, 对应目标为 $\mathbf{y}^{(i)}$;

计算梯度: $\mathbf{g} \leftarrow \frac{1}{m} \nabla_{\theta} \sum_i L(f(\mathbf{x}^{(i)}; \theta), \mathbf{y}^{(i)})$

$t \leftarrow t + 1$

更新有偏一阶矩估计: $s \leftarrow \rho_1 s + (1 - \rho_1) \mathbf{g}$

更新有偏二阶矩估计: $r \leftarrow \rho_2 r + (1 - \rho_2) \mathbf{g} \odot \mathbf{g}$

修正一阶矩的偏差: $\hat{s} \leftarrow \frac{s}{1 - \rho_1^t}$

修正二阶矩的偏差: $\hat{r} \leftarrow \frac{r}{1 - \rho_2^t}$

计算更新: $\Delta \theta \leftarrow -\epsilon \frac{\hat{s}}{\sqrt{\hat{r}} + \delta}$

应用更新: $\theta \leftarrow \theta + \Delta\theta$
end while

12.4 阅读材料

关于优化算法的系统研究始于 20 世纪 40 年代后期, 1951 年 Kuhn 和 Tucker 提出了著名的 KKT 条件。此后, 无论在基本理论还是在使用算法的研究方面都发展很快。目前, 优化算法已成为数学规划中内容十分丰富的一个分支。限于篇幅, 本篇仅叙述了优化算法中最基本的一些概念和算法, 力图便于读者掌握一些重要方法, 并为进一步深入学习打好基础。

在求解无约束优化问题的方法中, 变尺度法、共轭梯度法占有十分重要的地位。如欲进一步研究, 除本篇末列出的参考文献外, 还可参阅: 邓乃杨等著, 无约束最优化计算方法. 北京: 科学出版社, 1982 年。

对约束优化问题, 除本篇提到者外, 梯度投影法、简约梯度法、约束变尺度法、乘子罚函数法、序列二次规划法和起作用约束集法 (active set method) 等都是很重要的方法。其中简约梯度法和起作用约束集法对处理线性约束非线性目标函数十分有效。处理一般非线性约束非线性目标函数的有效方法, 有待进一步研究。

在最后一节中, 我们讨论了一系列算法, 通过自适应每个模型参数的学习速率以解决优化深度模型中的难题。此时, 一个自然的问题是: 该选择哪种算法呢?

遗憾的是, 目前在这一点上没有达成共识。Schaul(2014) 展示了许多优化算法在大量学习任务上极具价值的比较。虽然结果表明, 具有自适应学习速率 (以 RMSProp 和 AdaDelta 为代表) 的算法族表现得相当鲁棒, 不分伯仲, 但没有哪个算法能脱颖而出。

目前, 最流行并且使用很高的优化算法包括 SGD、具动量的 SGD、RMSProp、具动量的 RMSProp、AdaDelta 和 Adam。此时, 选择哪一个算法似乎主要取决于使用者对算法的熟悉程度 (以便调节超参数)。

12.5 习题

习题 12.1. 试用斐波那契法求函数

$$f(x) = x^2 - 6x + 2$$

在区间 $[0, 10]$ 上的极小点, 要求缩短后的区间长度不大于原区间长度的 8%。

习题 12.2. 试用 0.618 法重做习题 1.1, 并将计算结果与斐波那契法所得计算结果进行比较。

习题 12.3. 试用最速下降法求解

$$\min f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2$$

选初始点 $\mathbf{x}^{(0)} = (2, -2, 1)^T$, 要求做三次迭代, 并验证相邻两步的搜索方向正交。

习题 12.4. 试用最速下降法求函数

$$f(\mathbf{x}) = -(x_1 - 2)^2 - 2x_2^2$$

的极大点。先以 $\mathbf{x}^{(0)} = (0, 0)^T$ 为初始点进行计算, 求出极大点; 再以 $\mathbf{x}^{(0)} = (0, 1)^T$ 为初始点进行两次迭代。最后比较从上述两个不同初始点出发的寻优过程。

习题 12.5. 试用牛顿法重新解习题 1.4。

习题 12.6. 试用牛顿法求解

$$\max f(\mathbf{x}) = \frac{1}{x_1^2 + x_2^2 + 2}$$

取初始点 $\mathbf{x}^{(0)} = (4, 0)^T$, 用最佳步长进行。然后采用固定步长 $\lambda = 1$, 观察迭代情况, 并加以分析说明。

习题 12.7. 试用共轭梯度法求二次函数

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x}$$

的极小点, 此处

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$$

习题 12.8. 令 $\mathbf{x}^{(i)} (i = 1, 2, \dots, n)$ 为一组 \mathbf{A} 共轭向量 (假定为列向量), \mathbf{A} 为 $n \times n$ 对称正定阵, 试证

$$\mathbf{A}^{-1} = \sum_{i=1}^n \frac{\mathbf{x}^{(i)} (\mathbf{x}^{(i)})^T}{(\mathbf{x}^{(i)})^T \mathbf{A} \mathbf{x}^{(i)}}$$

习题 12.9. 试用变尺度法求解

$$\min f(\mathbf{x}) = (x_1 - 2)^2 + (x_1 - 2x_2)^2$$

取初始点 $\mathbf{x}^{(0)} = (0.00, 3.00)^T$, 要求近似极小点处梯度的模不大于 0.5。

习题 12.10. 试以 $\mathbf{x}^{(0)} = (0, 0)^T$ 为初始点, 使用

(1) 最速下降法 (迭代 4 次);

(2) 牛顿法;

(3) 变尺度法。求解无约束优化问题

$$\min f(\mathbf{x}) = 2x_1^2 + x_2^2 + 2x_1x_2 + x_1 - x_2$$

并绘图表示使用上述各方法的寻优过程。

习题 12.11. 分析约束优化问题

$$\begin{cases} \min f(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 3)^2 \\ x_1^2 + (x_2 - 2) \geq 4 \\ x_2 \leq 2 \end{cases}$$

在以下各点的可行下降方向 (使用式(12.58)和式(12.59)):

(1) $\mathbf{x}^{(1)} = (0, 0)^T$; (2) $\mathbf{x}^{(2)} = (2, 2)^T$; (3) $\mathbf{x}^{(3)} = (3, 2)^T$ 。并绘图表示各点可行下降方向的范围。

习题 12.12. 试用可行方向法求解

$$\begin{cases} \min f(\mathbf{x}) = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \\ x_1 + x_2 \leq 2 \\ x_1 + 5x_2 \leq 5 \\ x_1, x_2 \geq 0 \end{cases}$$

习题 12.13. 试用 SUMT 外点法求解

$$\begin{cases} \min f(\mathbf{x}) = x_1^2 + x_2^2 \\ x_2 = 1 \end{cases}$$

并求出罚因子等于 1 和 10 时的近似解。

习题 12.14. 试用 SUMT 外点法求解

$$\begin{cases} \max f(\mathbf{x}) = x_1 \\ (x_2 - 2) + (x_1 - 1)^3 \leq 0 \\ (x_1 - 1)^3 - (x_2 - 2) \leq 0 \\ x_1, x_2 \geq 0 \end{cases}$$

习题 12.15. 试用 SUMT 内点法求解

$$\begin{cases} \min f(\mathbf{x}) = (x + 1)^2 \\ x \geq 0 \end{cases}$$

习题 12.16. 试用 SUMT 内点法求解

$$\begin{cases} \min f(\mathbf{x}) = x \\ 0 \leq x \leq 1 \end{cases}$$

12.6 参考文献

- [1] 南京大学数学系计算数学专业编. 最优化方法. 北京: 科学出版社, 1978
- [2] 王德人编. 非线性方程组解法与最优化方法. 北京: 人民教育出版社, 1978
- [3] 中国科学院数学研究所运筹室编. 最优化方法. 北京: 科学出版社, 1980
- [4] 马仲蕃、魏权龄、赖炎连编. 数学规划讲义. 北京: 中国人民大学出版社, 1981
- [5] 薛嘉庆编. 最优化原理与方法. 北京: 冶金工业出版社, 1983
- [6] 席少霖、赵凤治编著. 最优化计算方法. 上海: 上海科学技术出版社, 1983
- [7] 郭耀煌等编著. 运筹学与工程系统分析. 北京: 中国建筑工业出版社, 1986
- [8] 徐光辉主编, 刘彦佩、程侃副主编. 运筹学基础手册. 北京: 科学出版社, 1999
- [9] M. 啊佛里耳著. 李元熹等译. 非线性规划——分析与方法. 上海: 上海科学技术出版社, 1979
- [10] D.M. 希梅尔布劳著, 张义燊等译. 实用非线性规划. 北京: 科学出版社, 1981
- [11] D.G. 鲁恩伯杰著, 夏尊铨等译. 线性与非线性规划引论. 北京: 科学出版社, 1980
- [12] David A Wismer, Chattergy R. Introduction To Nonlinear Optimization: A Problem Solving Approach, North-Holland Publishing Company, 1978
- [13] Mokhtar S Bazaraa, Shetty C M. Nonlinear Programming: Theory and Algorithms, John Wiley & Sons, 1979
- [14] Philip E Gill. Walter Murray and Margaret H. Wright, Practical Optimization, Academic Press, 1981
- [15] Fletcher R. Practical Methods of Optimization, Vol. 2, John Wiley & Sons, 1981
- [16] Edited by A. Bachem, M. Grötschel and B. korte, Mathematical Programming: The State of the Art, Bonn 1982, Springer-Verlag, 1983
- [17] Bottou L. and Bousquet, O. The tradeoffs of large scale learning. In NIPS' 2008, 2008
- [18] Bottou L. Online algorithms and stochastic approximations. In D. Saad, editor, Online Learning in Neural Networks. Cambridge University Press, Cambridge, UK, 1998
- [19] Polyak B. T. Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics, 4(5), 1-17, 1964
- [20] Sutskever I., Vinyals O., and Le Q. V. Sequence to sequence learning with neural networks. In NIPS' 2014, arXiv:1409.3215, 2014
- [21] Jacobs R. A. Increased rates of convergence through learning rate adaptation. Neural networks, 1(4), 295-307, 1988
- [22] Duchi J., Hazan E., and Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 2011

- [23] Hinton G. E. Tutorial on deep learning. IPAM Graduate Summer School: Deep Learning, Feature Learning, 2012
- [24] Kingma D. and Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014
- [25] Schaul T., Antonoglou I., and Silver D. Unit tests for stochastic optimization. In International Conference on Learning Representations, 2014

草稿请勿外传

索引

- ∞ 范数, 119
- l_0 范数, 141
- l_1 范数, 141
- l_2 范数, 141
- 1 范数, 118
- 2 范数, 118
- F 范数, 134
- Givens 旋转矩阵, 159
- Givens 变换, 217
- Givens 变换矩阵, 218
- Givens 旋转, 220
- Gram-Schmidt 正交化, 156, 208
- Haar 矩阵, 165
- Hadamard 矩阵, 167
- Householder 变换, 164
- Jensen 不等式, 525
- KKT 条件, 576
- k 阶主子式, 105
- k 阶顺序主子式, 105
- Lagrange 乘子, 562
- Lagrange 对偶函数, 562
- Lagrange 乘子向量, 562
- Lagrange 函数, 562
- LS 问题, 279
- LU 分解, 200
- Pagerank, 299
- p 范数, 118
- QR 分解, 207
- Slater 条件, 571
- softmax 函数, 375
- 一维搜索, 594
- 一阶矩, 388
- 上三角形线性方程组, 264
- 上三角矩阵, 264
- 上境图, 530
- 下三角矩阵, 264
- 下三角线性方程组, 264
- 下单峰函数, 597
- 下降算法, 593
- 不可能事件, 379
- 不定二次型, 104
- 不等式约束, 505
- 两个矩阵相似, 85
- 两个矩阵等价, 85
- 严格凸, 523
- 严格凹, 523
- 严格分离, 522
- 严格可行, 571
- 严格局部极小点, 592

- 严格最优, 506
中心极限定理, 413
中心矩, 395
二次型, 101
二次规划, 546
二阶收敛, 596
二阶锥规划, 547
亏损阵, 108
互信息, 438
互补松弛性, 575
代数余子式, 96
代数向量, 50
代数重数, 107
仿射包, 511
仿射子空间, 74
仿射映射, 91
仿射组合, 510
仿射维数, 511
仿射集合, 510
优化变量, 505
优化问题, 505
伯努利随机变量, 384
伴随矩阵, 98
似然函数, 465
似然函数的共轭先验, 475
余弦相似度, 131
依分布收敛, 408
依坐标收敛, 120
依概率收敛, 408
依范数收敛, 122
信息熵, 432
信赖域问题, 573
偶排列, 95
傅里叶矩阵, 171
像, 76
像域的容积/原域的容积, 98
像空间, 85
先验概率, 381
克罗内克积, 168
克莱姆法则, 97
全排列, 95
全概率公式, 381
共轭函数, 535
共轭梯度法, 610
共轭矩阵, 169
共轭转置, 169
内积, 122
内积空间, 123
几何向量, 50
几何规划, 549
几何重数, 107
凸优化问题, 539
凸函数, 523, 539
凸包, 515
凸组合, 512
凸集, 511
函数, 77
函数向量, 51
分片线性函数, 529
分离超平面, 520
分类, 128
切比雪夫距离, 129
列向量, 52
列空间, 86, 143
初等矩阵, 56
初等行变换, 56
判别模型, 387
半定规划, 550
半空间, 514

- 协方差, 394
协方差矩阵, 395
单位矩阵, 54
单位范数球, 120
单射, 77
单项式函数, 549
双射, 77
变换矩阵, 82
可交换群, 161
可逆矩阵, 57
合同矩阵, 102
同态, 79
同构, 79
同解方程组, 259
后验概率, 381
向量, 50
向量的模, 117
向量的正交分解, 148
向量空间, 62
向量组, 50
向量组等价, 67
向量范数, 118
回归模型, 468
图模型, 454
圆盘定理, 291
均值, 388
坐标, 70
基, 69
基变换, 84
基数函数, 119
基础解系, 259
增广矩阵, 259
复合映射, 77
复向量空间, 62
复数的模, 117
复矩阵, 52
多面体, 515
大数定律, 409
奇排列, 95
子空间, 63
子空间正交, 147
子空间的交, 65
子空间的和, 65
子空间的直和, 65
实向量空间, 62
实矩阵, 52
对偶可行, 563
对偶最优解, 568
对称矩阵, 101
导出组, 259
局部最优, 506
岭回归, 141, 472
左零空间, 144
差熵, 445
常数向量, 51
幂法, 293
平凡子空间, 63
平均互信息, 439
平均条件互信息, 443
平均联合互信息, 443
平稳点(驻点), 592
平面旋转变换, 218
广义牛顿法, 617
度量空间(距离空间), 126
开半空间, 514
张成的子空间, 68
张量, 60
弱对偶性, 570
强对偶性, 570
必然事件, 379

- 恒等映射, 78
惩罚函数, 630
惩罚项, 630
感知机, 579
扩展函数, 524
投影, 150
投影矩阵, 150
拉普拉斯分布, 385
拟牛顿条件, 617
损失函数, 369
搜索方向, 594
支撑函数, 529
支撑超平面, 522
改进的 Slater 条件, 571
数乘, 53
数域, 50
数据处理定理, 444
数量乘积, 55
整体最优, 506
整数规划, 507
斐波那契数, 598
方差, 390
方阵, 52
无交连, 147
无效约束, 623
无限维线性空间, 70
映射, 76
曼哈顿距离, 129
最优值, 505
最优对偶间隙, 570
最优解, 505
最佳步长, 594
最大似然估计值, 464
最大似然估计量, 464
最小二乘解, 280
最小二乘问题, 279
最速下降法, 370, 603
有向图模型, 457
有序基, 81
有效约束, 623
有限维线性空间, 70
期望, 388
期望损失, 388
条件期望, 399
条件概率, 381
条件概率密度函数, 386
条件熵, 437
极大线性无关组, 66
标准基, 69
标准差, 390
标准形, 102
标准正交, 127
标准正交基, 155
标量乘积, 55
核空间, 85
核范数, 142
梯度下降, 369
梯度下降法, 370
梯度流, 343
概率, 380
概率分布, 377
欠定方程组, 257
欧氏空间, 123
欧氏距离, 125, 129
正交, 127
正交投影, 150
正交矩阵, 127
正交群, 161
正交补, 148
正半定二次型（负半定二次型）, 104

- 正半定矩阵（负半定矩阵）, 104
正定二次函数, 608
正定二次型（负定二次型, 104
正定矩阵（负定矩阵）, 104
正态变量的线性变换不变性, 397
正惯性指数, 104
正项式函数, 549
步长, 594
残差向量, 279
汉明距离, 132
泛函, 77
混合中心矩, 395
混合整数规划, 508
混合矩, 395
满射, 77

熵, 432
熵函数, 433
熵函数的链规则, 442
爬山算法, 371
物理向量, 50
特征值, 105
特征向量, 105
特征多项式, 106
特征子空间, 106
特征根, 106
特征矩阵, 106
特征系, 106
状态空间模型, 374
独立, 380
独立同分布, 388
生成方法, 387
生成模型, 387
生成集, 68
目标函数, 369, 505

相似变换, 87
相关系数, 394
相容性条件, 134
相对内部, 511
相对误差, 596
相对边界, 511
矩, 395
矩阵, 52
矩阵 A 的（列）向量化, 133
矩阵 A 和 B 等价, 57
矩阵内积, 133
矩阵的分块, 55
矩阵的幂, 54
矩阵的转置, 58
离散型概率密度函数, 383
秩, 72
秩-1 分解, 73
稀疏规则算子 (Lasso), 141
等式约束, 505
算子, 77
算子范数, 135
系数矩阵, 259
累积分布函数, 382
约束准则, 571
约束函数, 505
线性分式函数, 520
线性变换, 87
线性搜索, 371
线性收敛, 596
线性无关, 66
线性映射, 78
线性替换, 101
线性相关, 66
线性矩阵不等式, 518
线性空间, 62

- 线性组合, 66
线性表出, 66
线段, 510
组合优化, 508
组合系数, 66
经验损失, 416
经验风险, 416
结构化概率模型, 454
绝对误差, 596
维数, 69
缩短率, 597
群, 161

联合熵, 437
聚类, 128
自信息量, 431
自同态, 79
自同构, 79
范数球, 515
行列式, 95
行向量, 52
行秩, 72
行空间, 143
规范化的 Hadamard 矩阵, 167
规范形, 104
试算点, 597
谱, 106
贝叶斯概率, 380
负向量, 50
负对数, 535
负惯性指数, 104
负梯度方向, 603
负矩阵, 54
赋范线性空间, 118
超定方程组, 257

超平面, 514
超线性收敛, 596
边缘概率质量函数, 386
边际概率密度函数, 386
连续概率密度函数, 383
迹, 99

适定方程组, 257
逆事件, 379
逆序, 95
逆序数, 95
逆映射, 79
逆矩阵, 57
透视函数, 519
部分和, 517
酉矩阵, 170
锥, 513
锥组合, 513
闵氏距离, 128
阿贝尔群, 161
随机事件, 378
随机变量, 382
随机向量, 51
随机试验, 378
障碍函数, 633
障碍项, 633
零向量, 50
零子空间, 63
零矩阵, 54
零空间, 144
非亏损阵, 108
非凸优化, 509
非平凡子空间, 63
非线性优化, 509
非负齐次, 513

非退化的线性替换, 101

鞍点, 370

频率派概率, 380

马尔可夫过程, 300

驻点, 370

(广义) 矩阵范数, 133

草稿请勿外传