

数据科学与工程数学基础 V2.0 / 20220906

数据科学与工程 数学基础

(第1版)
黄定江 编著

数据科学与工程数学基础初稿

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。

数据科学与工程数学基础初稿

序言

数据科学与工程核心课程的系列教材终于要面试了，这是一件鼓舞人心的事。作为华东师范大学数据学院的发起者和见证人，核心课程和系列教材一直是我心心念念的事情。值此教材出版发行之际，我很高兴能被邀请写几句话，做个回顾，分享一些感悟，也展望一下未来。

借着大数据热的东风，依托何积丰院士在 2007 年倡导成立的华东师范大学海量计算研究所，2012 年 6 月在时任 SAP 公司 CTO 史维学博士（Dr. Vishal Sikka）的支持下，我们成立了华东师范大学云计算与大数据研究中心。2013 年 9 月，学校发起成立作为二级独立实体的数据科学与工程研究院，开始软件工程一级学科下自设的数据科学与工程二级学科的博士和硕士研究生培养。在进行研究生培养的探索过程中，我们深切感受到我们的本科生的培养需要反思和改革。因此，到了 2016 年 9 月，研究院改制成数据科学与工程学院，随后就招收了数据科学与工程专业的本科生，第一届本科生已于 2020 年毕业。这是我们学院和专业的简单历史。经过这么几年的实践和思考，我们越发坚信当年对“数据科学与工程”这一名称的选择，“数据学院”和“数据专业”已经得到越来越多的认可，学院的师生也逐渐接受“数据人”这一称呼。

这里我想分享以下几方面的感悟：为什么要办数据专业？怎么办数据专业？教材为什么很重要？对人才培养有什么贡献？

为什么要办数据专业？数据是新能源，这是大家耳熟能详的一句话。说到能源，我们首先想到的是石油，所以大家就习惯把数据比喻成石油。但是，在我们看来，“新能源”对应的英文说法应该是“New Power”。“Data is Power”，这是我们的基本信念，也是我们为什么要办数据学院的根本动机。数据是人类文明史上的第三个重要的 Power，蒸汽能（Steam Power）和电能（Electric Power）引发了第一次和第二次工业革命。如果说蒸汽能和电能造就了从西方开始的两百多年的工业文明，数据能（Data Power）将把人们带入数字文明时代。数据是数字经济发展的重要的生产要素，这个生产要素不同于土地、劳动力，也不同于资本、技术。如果要给数据找一个恰当的比拟，也许只有十九世纪末伟大的发明家尼古拉·特斯拉发明的交流电。数据是新时代的交流电，就像上个世纪，交流电给世界带来的深刻变化一样，因为人们对数据能（Data Power）认识的提高，我们将进入一个“未来已来，一切重构”的时代。数据学院就像一百多年前的电力学院或电气学院。

怎么办数据专业？我们数据学院脱胎于软件工程学院，在此以前还有计算机科学与工程学院，数据相关的研究和偏向管理和图书情报方向的信息系统学科和专业也密切相关，应用数学、概率统计更是数据分析和处理的理论基础，不可或缺。到底什么样的专业才算是数据专业？起初的时候，这对我们来说基本上可以说是一个“灵魂拷问”。为此，我们发起成立了国内十五所高校三十多位知名教授组成立“高校数据科学与工程专业建设协作组”。我们相信，有了先进的理念，再加上集体的力量，数据专业建设的探索之路就能走通。协作组已经召开了四次研讨会，

确定了称为 CST 的专业建设路线图, C 代表 Curriculum (培养计划), S 代表 Syllabus (课程大纲), T 代表 Textbook (教材建设)。在得知我们的工作后, ACM/IEEE 计算机工程学科规范主席 John Impagliazzo 教授邀请我们参与了 ACM/IEEE 数据科学学科规范的制定。协作组达成共识: 专业课程分为基础课、核心课、方向课三类, 核心课是体现专业区分度的一组课程。与数据专业 (DSE) 最相近的专业就是计算机科学与工程 (CSE) 与软件工程 (SE) 两个专业, 我们确定的第一批 DSE 区别于 CSE 和 SE 的 8 门核心课程是: 数据科学与工程导论, 数据科学与工程数学基础, 数据科学与工程算法基础, 应用统计与机器学习、当代人工智能, 云计算系统、分布式计算系统、当代数据管理系统。随后我们又确定两门课纳入这个系列, 分别是: 区块链导论——原理、技术与应用, 数据中台初阶教程。数据专业作为一个新专业, 三类课程的边界还不清晰, 我们重点关注核心课程上面, 核心课有遗漏的知识点可以纳入基础课或方向课。这样可以保证知识体系的完整性, 简单起步, 快速迭代。随着实践和认识的深入, 逐渐明晰三类课程的边界, 形成完善的培养计划。

教材为什么很重要? 建设好一个专业, 培养计划和课程体系固然很重要, 但落实在根本上是教材。一套好的教材是建成一个好的专业的前提, 放眼看去, 无论是国内还是国外, 无论是具体高校还是国家区域层面, 这都是不争的事实, 好的专业都有成体系的好的教材。当然, 现在的教材已经不仅仅指的是单纯的一本教科书, 还有深层次的内容, 比如说具体的教学内容和教学方式。我们都知道, 教材是知识的结晶, 是站到巨人肩膀上去的台阶。在自然科学领域, 确实如此, 一百年前我们民族的仁人志士呼唤“赛先生”, 在中华大地上科学的传播带来了翻天覆地的变化。在更广泛的领域, 教材也还是技术、工艺和文化的传承, 是产业发展的助推器。拿信息技术来举例, 技术的源头和产业的发祥地都在美国和欧洲, 像 IBM、Lucent、Oracle 等跨国企业在我国商业上取得的巨大成功无一不与他们重视教材开发密切相关。试想一下, 我们的学生在课堂上学的都是他们的研究和研发的东西, 等走上工作岗位, 自然会对熟悉的技术和系统有亲近感, 这应该是产业或产品生态最重要的一个环节。本世纪以来, 随着互联网的蓬勃发展, 人们已经深刻认识到, 互联网改变世界。在人类的文明史上, 没有任何一项科研成果像互联网这样深刻地改变人, 改变世界。互联网之所以能改变世界, 是因为它真正发挥了数据的威力。互联网实现了信息技术发展从“以计算为中心”到“以数据为中心”的路径转变。用“昔日王谢堂前燕, 飞入寻常百姓家”来形容很多我们以前甚至当前教材上的一些内容, 可能毫不为过。以互联网为代表的新型产业的发展, 极大地推动了技术的进步, 我们已经到了可以编写自己的教材, 形成自己的技术体系和科学理论体系的时候了。我们是现代科学的后来者, 已经习惯了从科学到技术再到应用的路径, 现在有了成功的应用, 企业也发展出了领先的技术, 学界可以在此基础上发展出技术体系和科学理论体系, 应用、技术和科学的联动才是真正的创新之路。

对人才培养有什么贡献? 在信息技术领域, 迄今为止我们更多的是参考或沿袭了西方发达国家的培养计划和教材体系, 在改革开放以来的四十年, 这种“拿来主义”的做法很有效, 培养了大量的人才, 推动了我国的社会经济发展。但总的来说, 我们的高校在这一领域更像是在培养“驾驶员”, 培养开车的人, 现在到了需要我们来培养自己的造车人的时候了。技术发展趋

势如此，国际形势也对我们提出了这样的要求。我们处在一个大变局的时代，世界充满不确定性，开放和创新是应对不确定性的不二之选。创新成为人才培养的第一性原理，更新观念，变革教育，卓越育人是我们华东师范大学新时期人才培养的基本理念。人才培养是大学的第一要务，科学研究、社会服务和文化传承是大学的另外三大职能，大学通过这三大职能的实现可以更好地服务于人才培养。这也是数据专业核心课程系列教材的建设的指导思想，我们计划久久为功的这一件事是我们践行这一理解的一个小小的行动。

最后，要特别表示感谢，感谢华东师范大学和高等教育出版社的支持和鼓励，感谢数据专业建设协助组的各位老师们的通力协作和辛勤劳动，也要感谢数据学院师生的信任和付出。心有所信，方能行远；因为相信，所以看见。希望作为探路者的艰辛能成为大家学术和职业生涯中的一笔重要财富。

“The best way to predict the future is to invent it” ——Alan Kay

周傲英

华东师范大学

2020 年 11 月于上海

数据科学与工程数学基础初稿

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、Cholosky 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、稀疏优化以及低秩矩阵恢复问题的求解等等, 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上述三类问题的求解, 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, 而随机过程则在深度强化学习中具有广泛的应用。

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、核密度估计和非参数估计等；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；并引出优化问题的最优化条件；介绍数据科学中各种常见的优化问题的对偶性问题。数据科学和机器学习中的很多问题是非凸的，比如最大割问题，可以通过转换为对偶问题进行有效求解。

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内容要点。

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

本书写作参考了 Gilbert Strange 教授的 *Linear Algebra and Its Application* 和 *Linear Algebra and Learning from data*, Larry Wasserman 教授的 *All of Statistics*, Thomaas Cover 教授的 *Elements of Information Theory*, Giuseppe Calafiori 教授和 El Ghaoui Laurent 教授的 *Optimization Models*, Stephen Boyd 教授和 Lieven Vandenberghe 教授的 *Convex Optimization* 等经典数学教材以及 Vladimir Vapnik 教授的 *Statistical learning theory*, Hastie Trevor 教授, Tibshirani Robert 教授和 Friedman Jerome 教授的 *The Elements of Statistical Learning*, Ian Goodfellow, Yoshua Bengio, Aaron Courville 的 *Deep Learning* 等经典的机器学习教材。

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由华东师范大学、哈尔滨工业大学、中国人民大学、中山大学、东北大学、西北工业大学、河南大学以及桂林电子科技大学等 15 所高校组成的数据专业协作组以及华东师范大学出版社和高等教育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。郝珊峰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。

限于作者的知识水平，书中难免有不妥和错误之处，恳请读者不吝批评和指正。

黄定江

2020 年 10 月

数据科学与工程数学基础初稿

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。

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

向量和矩阵

a	标量 (整数或实数)
\mathbf{a}	向量
\mathbf{A}	矩阵
\mathbf{A}	张量
\mathbf{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
$\mathbf{e}^{(i)}$	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_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
$\mathbf{A}_{i,:}$	矩阵 \mathbf{A} 的第 i 行
$\mathbf{A}_{:,i}$	矩阵 \mathbf{A} 的第 i 列
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
$\mathbb{A} \setminus \mathbb{B}$	差集, 即其元素包含于 \mathbb{A} 但不包含于 \mathbb{B}
\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}^T	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
$\text{Null}(\mathbf{A}^T)$	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 对 $\mathbf{x} \in \mathbb{R}^m$ 的 Jacobian 矩阵
$\nabla_{\mathbf{x}}^2 f(\mathbf{x})$ 或 $\mathbf{H}_f(\mathbf{x})$	f 在点 \mathbf{x} 处的 Hessian 矩阵
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
$\text{Cov}(X, Y)$	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
$R_{\text{exp}}(f)$	期望风险
$R_{\text{emp}}(f)$	经验风险
信息论基础	
$I(x_i)$	事件 x_i 的自信息
$I(x_i; y_i)$	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
$\text{Beta}(a, b)$	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
\mathbf{w}	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
$\mathbf{epi}f$	函数 f 的上镜图
$\mathbf{prob}S$	事件 S 的概率
$\mathbf{dom}f$	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
$\text{dist}(A, B)$	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

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	5
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	10
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	18
1.3.2 数据	22
1.3.3 模型	25
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
1.4.1 数值线性代数简介	39
1.4.2 概率与信息论简介	39
1.4.3 最优化简介	39
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	40
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	41
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
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	65
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	71
2.2.7 仿射空间	73
2.3 线性映射与线性变换	74
2.3.1 线性映射: 线性模型的观点	75
2.3.2 线性映射的矩阵表示	78
2.3.3 线性变换	84
2.3.4 仿射映射	88
2.4 矩阵的基本特征	91
2.4.1 行列式	91
2.4.2 迹运算	96
2.4.3 对称矩阵与二次型	97
2.4.4 特征值与特征向量	102
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	117
3.1.2 内积与夹角	122
3.1.3 数据科学中常用的相似性度量	128
3.1.4 矩阵的内积与范数	133
3.1.5 范数在机器学习中的应用	139
3.2 正交与投影	140
3.2.1 矩阵的四个基本子空间	140

3.2.2 四个基本子空间的正交性	144
3.2.3 正交投影	147
3.3 正交基与 Gram-Schmidt 正交化	152
3.3.1 标准正交基	152
3.3.2 Gram-Schmidt 正交化	153
3.4 具有特殊结构和性质的矩阵	154
3.4.1 特殊的正交变换矩阵——旋转	155
3.4.2 反射矩阵	159
3.4.3 信号处理中常见的正交矩阵	162
3.5 阅读材料	171
第四章 矩阵分解	177
4.1 数学中常见的具有特殊结构的矩阵	178
4.2 数据科学中常见的矩阵	184
4.2.1 图的矩阵	184
4.2.2 低秩矩阵	193
4.3 LU 分解	196
4.3.1 LU 分解	196
4.3.2 选主元的 LU 分解	201
4.4 QR 分解	205
4.4.1 基于 Gram-Schmidt 正交化的 QR 分解	205
4.4.2 基于 Householder 变换的 QR 分解	208
4.4.3 基于 Givens 变换的 QR 分解	211
4.5 谱分解与 Cholesky 分解	214
4.5.1 谱分解	214
4.5.2 Cholesky 分解	219
4.6 奇异值分解	221
4.6.1 奇异值分解	222
4.6.2 基于奇异值分解的矩阵性质	230
4.6.3 奇异值分解与低秩表示	235
4.7 阅读材料	240
第五章 矩阵计算问题	245
5.1 线性方程组的直接解法	246
5.1.1 线性方程组问题	246

5.1.2 一般线性方程组解的理论	247
5.1.3 容易求解的线性方程组	250
5.1.4 基于矩阵分解的方阵系统的直接解法	255
5.1.5 非方阵系统的直接求解方法	260
5.1.6 敏度分析与其他方法	266
5.2 最小二乘问题	269
5.2.1 最小二乘问题与线性回归	269
5.2.2 最小二乘问题的求解方法	273
5.2.3 最小二乘问题的变体	277
5.2.4 最小二乘问题的解的敏感性	279
5.3 特征值计算	280
5.3.1 矩阵特征值分布范围的估计	280
5.3.2 幂法	283
5.3.3 反幂法	286
5.3.4 特征值计算的应用: Pagerank 网页排名	290
5.4 阅读材料	292
第六章 向量与矩阵微分	298
6.1 向量函数和矩阵函数	299
6.1.1 函数	299
6.1.2 算子	303
6.1.3 泛函	304
6.1.4 机器学习中的风险泛函	305
6.2 统计机器学习中的非概率型函数模型	307
6.2.1 线性模型中的函数	307
6.2.2 感知机模型中的函数	308
6.2.3 支持向量机	310
6.2.4 降维和主成分分析中函数	318
6.2.5 聚类中的函数	319
6.3 深度神经网络中的函数构造	321
6.3.1 深度神经网络模型函数的构造过程	322
6.3.2 激活函数	325
6.4 向量和矩阵函数的梯度	329
6.4.1 向量函数的梯度	330
6.4.2 矩阵函数的梯度	333

6.5 对矩阵微分	335
6.5.1 矩阵微分与偏导数的联系	336
6.5.2 关于逆矩阵的函数的微分	337
6.5.3 关于行列式函数的微分	337
6.6 迹函数的微分和迹微分法	339
6.7 向量值函数和矩阵值函数的梯度	341
6.7.1 向量值函数的梯度	341
6.7.2 矩阵值函数的梯度	342
6.7.3 向量值函数微分	342
6.8 链式法则	344
6.9 反向传播和自动微分	346
6.9.1 反向传播	346
6.9.2 自动微分	348
6.10 高阶微分和泰勒展开	352
6.10.1 Hessian 矩阵	352
6.10.2 线性化和多元泰勒级数	353
6.11 阅读材料	354
第七章 概率基础	358
7.1 概率论基本概念回顾: 数据不确定性描述的观点	358
7.1.1 概率论基本概念	358
7.1.2 概率论公理	360
7.1.3 独立事件和条件概率	361
7.1.4 贝叶斯公式	362
7.2 随机变量及其分布	363
7.2.1 常用的随机变量及其分布	365
7.2.2 多维随机变量及其分布函数	366
7.3 随机变量的数字特征	370
7.3.1 期望	370
7.3.2 方差	372
7.3.3 一些重要分布的期望和方差	373
7.3.4 协方差和相关系数	375
7.3.5 矩和协方差矩阵	377
7.3.6 条件期望	380
7.3.7 方差的应用: 过拟合与偏差-方差分解	381

7.4 概率不等式	384
7.5 大数定律与中心极限定理	389
7.5.1 引言	389
7.5.2 大数定律	390
7.5.3 中心极限定理	392
7.5.4 大数定律的推广及其在统计学习中的应用	394
7.6 随机过程简介	397
7.6.1 马尔可夫链	398
7.6.2 高斯过程	404
7.7 阅读材料	405
第八章 信息论基础	408
8.1 熵、相对熵和互信息	409
8.1.1 自信息	410
8.1.2 熵及其性质	411
8.1.3 联合熵和条件熵	415
8.1.4 互信息和相对熵	417
8.1.5 熵、相对熵和互信息的链式法则	421
8.1.6 信息不等式	421
8.2 连续分布的微分熵和最大熵	423
8.2.1 连续信源的微分熵	423
8.2.2 连续信源的最大熵	426
8.3 信息论在数据科学中的应用	426
8.3.1 基于信息量的度量	426
8.3.2 其他概率相关的度量	428
8.4 阅读材料	431
第九章 概率模型	434
9.1 从概率到统计	435
9.1.1 统计的基本概念	435
9.1.2 模型、统计推断和学习	438
9.2 概率密度函数的估计	444
9.2.1 概率密度估计引入	444
9.2.2 基于频率观点的参数估计方法	446
9.2.3 贝叶斯推断	452

9.2.4	统计决策与贝叶斯估计	459
9.2.5	非参数估计	473
9.3	概率模型与图表示	485
9.3.1	概率模型的有向图表示	485
9.3.2	概率模型的无向图表示	491
9.4	机器学习中的概率模型	498
9.4.1	机器学习的概率思路	498
9.4.2	机器学习中的概率模型	499
9.4.3	深度学习中的概率模型	507
9.4.4	强化学习中的概率模型	515
9.5	阅读材料	515
第十章 优化基础		523
10.1	优化简介	524
10.1.1	数据科学与机器学习中最优化问题的例子	525
10.1.2	其他常见的优化问题举例	527
10.1.3	优化问题的一般形式	530
10.1.4	优化问题的分类	533
10.2	凸集	536
10.2.1	凸集	536
10.2.2	重要的凸集例子	539
10.2.3	保持凸集的运算	543
10.2.4	分离与支撑超平面	548
10.3	凸函数	550
10.3.1	凸函数的定义和基本性质	551
10.3.2	凸函数举例	552
10.3.3	凸函数的性质	553
10.3.4	凸函数的判定条件	554
10.3.5	保凸运算	560
10.3.6	共轭函数	567
10.3.7	次梯度	570
10.4	凸优化	577
10.4.1	凸优化问题	577
10.4.2	典型凸优化及其在数据科学中应用示例	581
10.5	阅读材料	588

10.6 习题	590
10.7 参考文献	592
第十一章 最优性条件和对偶理论	598
11.1 无约束优化的最优性条件	598
11.2 Lagrange 对偶函数	602
11.2.1 Lagrange 函数与对偶函数	602
11.2.2 常见优化问题目标函数的对偶函数	604
11.2.3 Lagrange 对偶函数与共轭函数的联系	606
11.3 Lagrange 对偶问题	608
11.3.1 Lagrange 对偶问题	608
11.3.2 对偶性质	610
11.3.3 常见优化问题的对偶问题及强对偶性	612
11.3.4 强对偶性定理的证明	613
11.3.5 强弱对偶性的极大极小描述	617
11.4 最优性条件	618
11.4.1 互补松弛条件	618
11.4.2 KKT 最优性条件	618
11.4.3 通过解对偶问题求解原问题	620
11.5 数据科学中常见模型的对偶问题	622
11.5.1 线性可分支持向量机	622
11.5.2 线性支持向量机	625
11.6 阅读材料	626
11.7 习题	627
11.8 参考文献	628
第十二章 优化算法	631
12.1 无约束优化	632
12.1.1 线搜索	635
12.1.2 一阶方法	644
12.1.3 二阶方法	658
12.2 约束优化	672
12.2.1 可行方向法	673
12.2.2 外点法	678
12.2.3 内点法	685

12.3	复合优化算法	690
12.3.1	近似点梯度法	691
12.3.2	分块坐标下降法	697
12.3.3	交替方向乘子法	702
12.4	深度学习常用优化算法	705
12.4.1	随机梯度下降	706
12.4.2	动量梯度下降	708
12.4.3	自适应学习速率	710
12.4.4	应用实例：多层感知机	713
12.5	在线凸优化算法简介	715
12.5.1	在线凸优化模型	715
12.5.2	一阶方法	717
12.5.3	二阶方法	720
12.6	阅读材料	723
12.7	习题	724
12.8	参考文献	726

数据科学与工程数学基础初稿

数据科学与工程数学基础初稿

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，那么它们就转变成知识。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，是所有能被输入计算机中，且能被计算机处理的符号的集合，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，

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	噪音	
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

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地理解, 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地存储和组织。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

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

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。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。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，它们的数学模型已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如下：（1）更高效地存储和组织大数据；（2）发现大数据中有别于传统离散关系的新的数据关系，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（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。

如果我们把大数据特性中海量高维数据集抽象为无限数据集；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

N	大数据结构	数学关系
1	表示结构	向量、矩阵、张量、拓扑空间、流形、李群和随机表示等
2	关系结构	相关关系（相似关系、顺序关系、类别关系）和因果关系
3	模型结构	概率相关模型和非概率模型（函数模型）
4	计算结构	优化计算、统计计算
5	运算结构（结构算法）	分类、聚类、回归、降维、排序、密度估计等等

大数据结构：在多元无限快速增长的数据集中，研究数据对象的数学关系和数据运算

图 1.3: 大数据结构

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、度数
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 1.4: 大数据的数学和运算结构

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奇异数据集中，研究数据对象的数学关系和数据运算，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，所以我们在里仍然把它归为应用数学的范畴），见图 1.4。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。首先我们在 1.2 节通过数据科学或人工智能中两类常见的应用场景：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如何把数据驱动的图像感知和自然语言处理问题转变成一个基本的数据分析计算任务。第一个案例是涉及图像识别的感知任务分析，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，计算机视觉没有在一个暑期内解决，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。

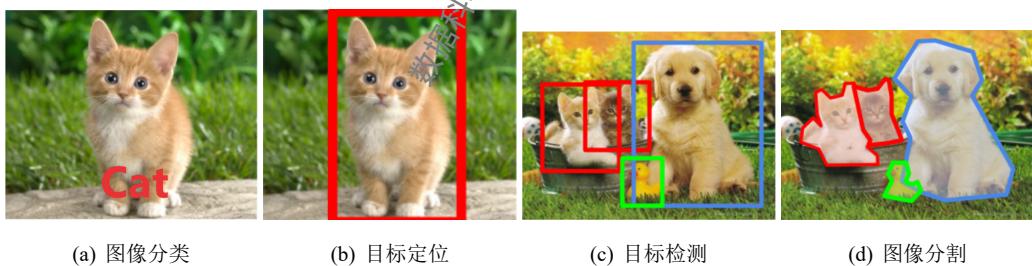


图 1.5: 计算机四大视觉任务

图1.5是计算机视觉领域的四大基本任务，包括图像分类、目标定位、目标检测和图像分割。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；目标定位需要解决图像中目标所在位置的问题；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；图像分割进一步分为语义分割和实例分割，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，而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。

例 1.2.1. 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。

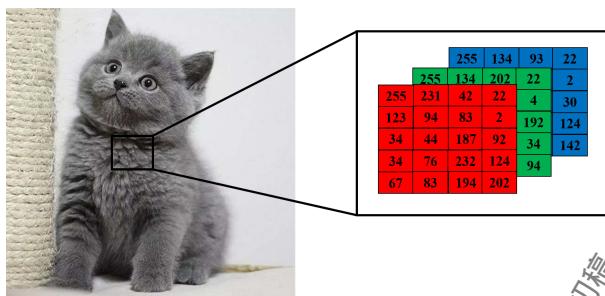


图 1.6: 图像的数据表示

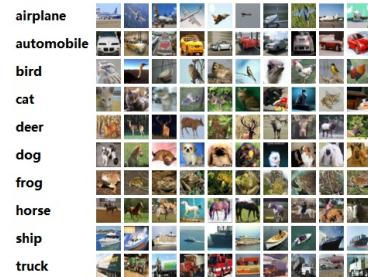


图 1.7: CIFAR-10

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, 我们首先对数据集进行简单划分。在例1.2.1中, 我们选取 CIFAR-10 图像分类数据集如图1.7所示, 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。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

K 最近邻分类 (KNN) 算法是数据挖掘分类技术中最简单的方法之一。所谓 K 最近邻指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 $\mathbf{1}_1$ 和 $\mathbf{1}_2$, 然后计

算它们之间的 L_1 距离：

$$d_1(\mathbf{1}_1, \mathbf{1}_2) = \sum_{p=1}^{3072} |\mathbf{1}_1^p - \mathbf{1}_2^p|$$

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

add
→ 456

图 1.8: 以图片中的一个颜色通道为例

如图1.8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 距离还是其它度量策略呢？这些选择的值被称为超参数。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。

线性分类器

尽管 KNN 算法直观且易于实现，但受限于模型本身的特性，该方法通常性能不佳，并伴随着高昂的计算代价，因此我们寻求一种更强大的方法来解决图像分类问题。

作为另一种具有代表性的分类方法，线性分类器通过特征的线性组合来实现样本分类。该方法通过评分函数得到原始图像到类别分数的映射，接着使用损失函数来量化预测分类标签与真实标签之间的一致性。这样分类任务被转化成一个最优化问题，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。

在本方法中，我们从最简单的概率函数开始，一个线性映射：

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

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。

为了便于可视化。假设图像只有 4 个像素，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} 进行矩阵乘法再加上偏差向量，得到各个分类的

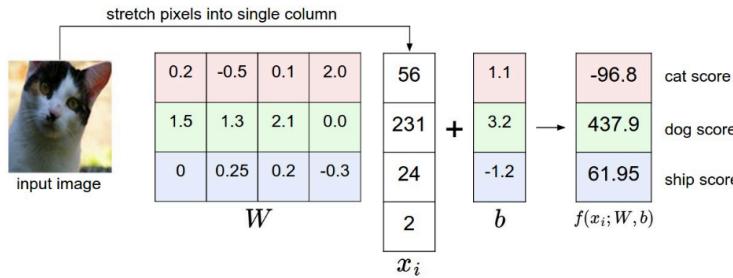


图 1.9: 评分函数可视化。假设图像只有 4 个像素 (也不考虑 RGB 通道), 有 3 个分类 (cat、dog、ship)。首先将图像像素拉伸为一个列向量, 与 W 进行矩阵乘法再加上偏差向量, 然后得到各个分类的分值

分值。需要注意的是, 权重 W 估计不准确: 真实类别猫分类的分值非常低。从图 1.9 中看, 算法认为这个图像是一只狗。这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 函数是最常用的分类器之一, 使用 softmax 函数将一组得分范围在 $(-\infty, +\infty)$ 的映射 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:

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

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

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

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. 随机本地搜索。从随机权重开始，然后生成一个随机的 $\delta\mathbf{W}$ ，只有当 $\mathbf{W} + \delta\mathbf{W}$ 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1.10所示：

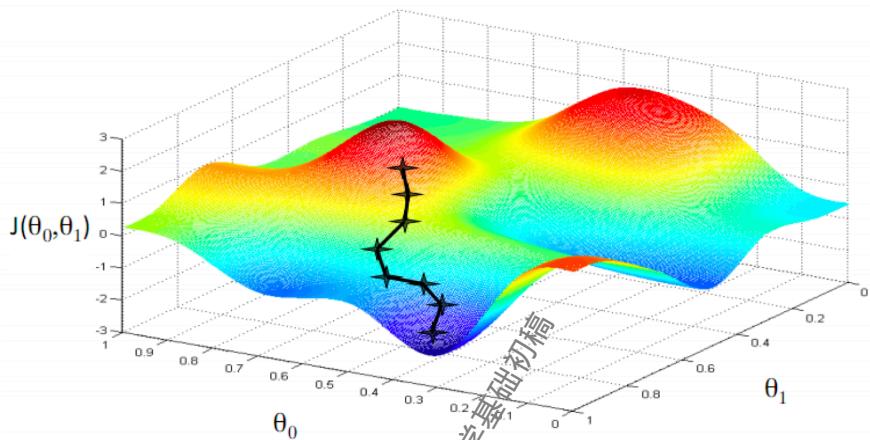


图 1.10: 梯度下降算法

卷积神经网络

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。根据基于参数的函数映射，可将其拓展为一个远比线性函数复杂的函数——卷积神经网络。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这两个部分将会保持相对稳定。

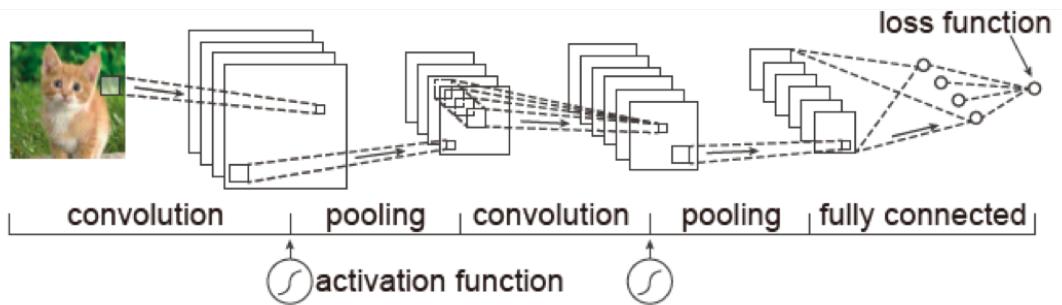


图 1.11: 一个典型的 CNN 架构图

一个典型的卷积神经网络架构如图1.11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-激活层-池化层-全连接层]。

输入层通常是输入卷积神经网络的原始数据或经过预处理的数据，可以是图像识别领域中原始三维的多彩图像，也可以是音频识别领域中经过傅利叶变换的二维波形数据，甚至是自然语言处理中一维表示的句子向量。以图像分类任务为例，输入层输入的图像一般包含RGB三个通道，是一个由长宽分别为 H 和 W 组成的3维像素值矩阵 $H \times W \times 3$ ，卷积网络会将输入层的数据传递到一系列卷积、池化等操作进行特征提取和转化，最终由全连接层对特征进行汇总和结果输出。

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。具体来说，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。而激活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

计算机视觉主要是让计算机具有“看”客观世界的能力，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（支持向量机）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1.12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 1.12: 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 1.2.2. 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。影评分类数据如图 1.13 所示:

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

图 1.13: 两条电影影评数据

这里主要有两个问题需要考虑: 一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); 二是基于影评数字化表示, 对其进行分类建模。考虑 1.2.2 的两条影评。一类是正类影评, 用 1 表示。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, 然后在此基础上, 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, 而连续表示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, 以便计算时可以处理, 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、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

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1.2.2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1.2.2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, 其他都为 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

词袋模型表示也被称为计数向量表示。在这种表示方法中, 把文本看作一个词袋, 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 , 1.2.2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: 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1.2.2中两个影评文本可以用两个 10 维向量表示: 文本 1 表示为: $[1, 1, 1, 1, 1, 1, 1, 0, 0]$, 文本 2 表示为: $[0, 0, 0, 1, 1, 0, 0, 1, 1]$ 。

TF-IDF

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中出现时, 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 TF 越高，那么这个词对这篇文档就越重要；IDF 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 中的 tf-idf 值：

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

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

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。

共现矩阵

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作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:

	the plot of this movie is funny excellent awful indeed									
the	0	1	0	0	0	0	0	0	0	0
plot	1	0	1	0	0	0	0	0	0	0
of	0	1	0	1	0	0	0	0	0	0
this	0	0	1	0	2	0	0	0	0	0
$X =$	$movie$	0	0	0	2	0	2	0	0	0
is	0	0	0	0	2	0	1	0	1	0
funny	0	0	0	0	0	1	0	1	0	0
excellent	0	0	0	0	0	0	1	0	0	0
awful	0	0	0	0	0	1	0	0	0	1
indeed	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, 但由于其稀疏性造成的数据稀疏和维度灾难的问题依旧没有得到解决, 尤其是当语料库非常大时, 这个矩阵会非常稀疏, 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。降维是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这些内容。

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。词嵌入表示可以理解成是一种映射, 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远。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。

连续词袋模型

连续词袋 (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 1.14 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 $\mathbf{w}_t \in \mathbb{R}^{|V|}$, $|V|$ 表示语料库词典中词汇的数量, n 为输入向量的个数(与窗口大小相同), C 是上下文单词的个数, $\mathbf{V} \in \mathbb{R}^{|V| \times n}$ 和 $\mathbf{U} \in \mathbb{R}^{n \times |V|}$ 是两个权重矩阵。

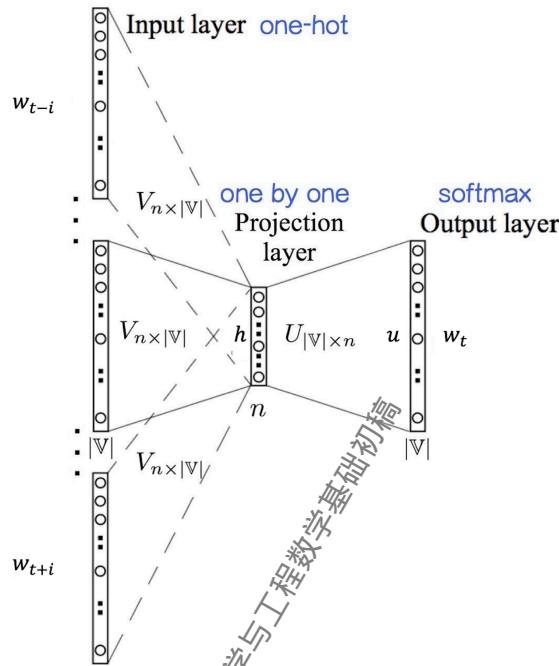


图 1.14: CBOW 模型。 \mathbf{w}_t 为目标词, 其余词 $\mathbf{w}_i, i \neq t$ 为上下文 $\mathbf{w}_{context}$

第一步: 计算隐层 \mathbf{h}

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

第二步: 计算输出层输出

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

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

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作是一个概率分布。

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 \mathbf{V} 和 \mathbf{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 \mathbf{V} 和 \mathbf{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，主要考虑使用传统方法和神经网络方法这两类方法对文本分类问题进行数学建模。例如，使用 TF-IDF 对文档进行表示，然后使用逻辑回归 (Logistics Regression, LR) 模型对文本分类进行数学建模为传统方法。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) 对词向量特征进行提取并用 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 $P(y = 1|\mathbf{x})$ 进行建模，其中 \mathbf{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：

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

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。因此可以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 \mathbf{h} 的概念。经过 RNN 编码后， \mathbf{h} 可以提取序列数据的特征。RNN 架构图如下图1.15所示：第 t 时刻的输入以及第 $t-1$ 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1.16。在处理文本数据时，图1.16中的 \mathbf{x}_1 可以看作是第一个单词的词向量， \mathbf{x}_2 可以看作是第二个单词的词向量，依次类推在处理语音数据时，此时 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n$ 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。

在文本分类问题中，对于一个包含 n 个单词的文本 $W = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n)$ ，我们使用 RNN 对文本进行序列建模编码，如下图1.17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。

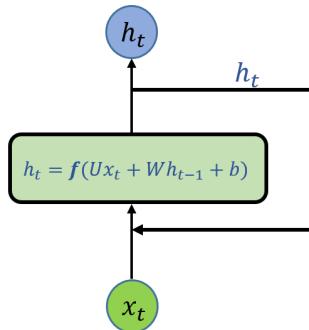


图 1.15: RNN 结构图

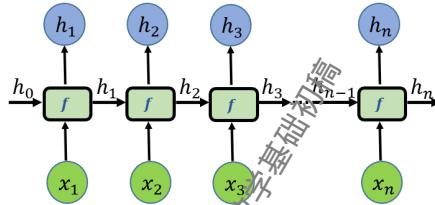


图 1.16: RNN 按时间展开

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:

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

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

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

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:

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

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。

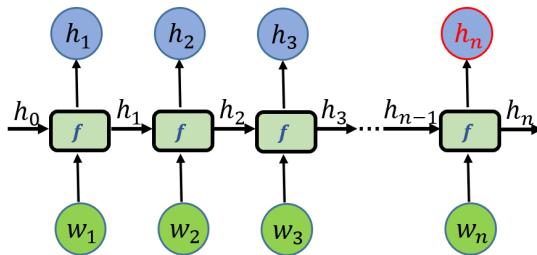


图 1.17: RNN 对文本进行编码

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不一样，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、降维等等，通常首先要把数据进行恰当地表示，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、概率论和信息论、优化理论，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

上一节通过两个例子讨论了当前人工智能中的处理任务可以转换为数据分析中的分类或降维等运算任务。我们把这种数据驱动的人工智能称为数据智能。机器学习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。首先给出机器学习概览，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，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，因此我们对本书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，或者说使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，可以归结为机器学习，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？粗略地说，深度学习是主要使用深度神经网络的机器学习算法，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1.18所示。

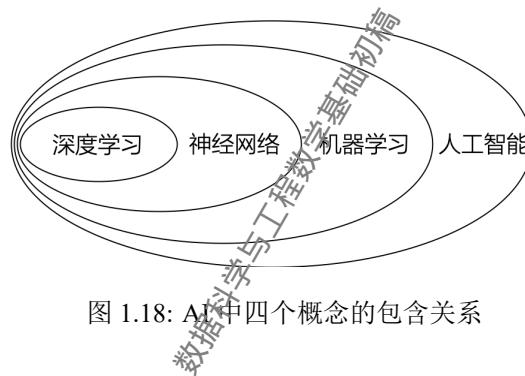


图 1.18: AI 中四个概念的包含关系

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1.19所示。

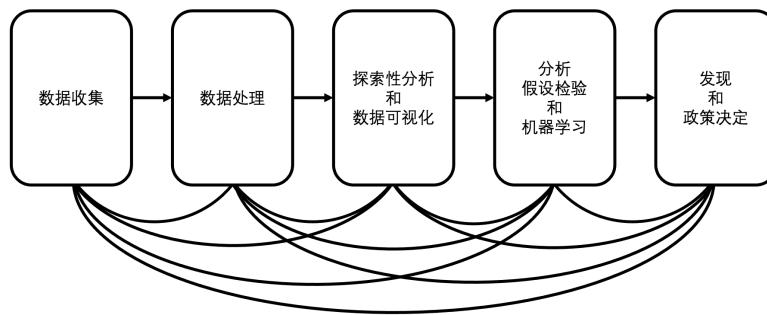


图 1.19: 数据分析与机器学习在数据全生命周期所处的阶段

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1.20左边表格表示一些客户情况数据，包括客户的基本信息特征和商家对其类别的标记，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，再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。

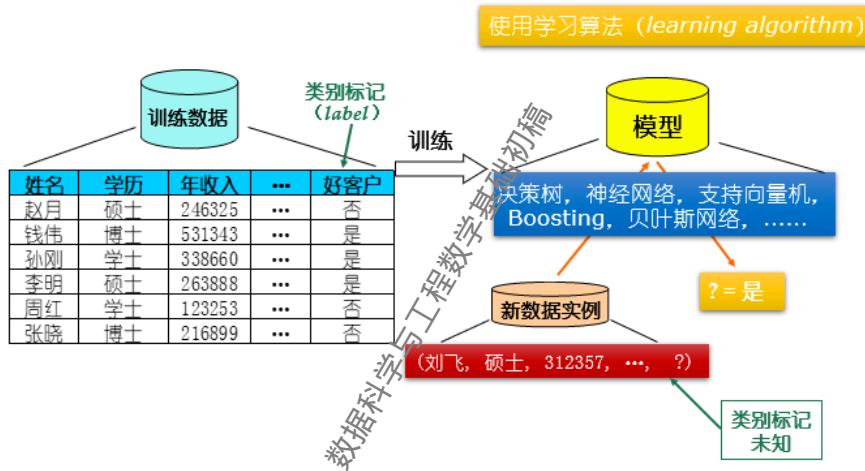


图 1.20: 典型的机器学习过程

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；假设要学习的模型属于某个函数的集合，称为假设空间并且应用某个评价准则，从假设空间中选取一个最优的模型，使它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，其中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。

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。当输出变量取有限个离散值时, 预测问题就成了分类问题, 这时输出变量可以是连续变量, 也可以是离散的。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, 由训练数据来学习模型的结构和参数, 这几类任务都是无标记信息的。

这些任务按照是否从有无标记数据中学习, 可以分为监督学习、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, 最后得到学习结果为函数。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。主动学习比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可夫决策过程, 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

考虑以下场景。比如有一份人力资源数据。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。

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

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，利用领域知识，例如学位可分为学士学位、硕士学位、博士学位，或者邮政编码，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)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万为单位。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

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

示, 年薪全部转换成以万为单位。在转换成计算机读取的数据后, 接下来我们需要给这些数据赋予数学结构, 并使用这些数据建立机器学习模型。

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 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。除了人力资源数据可以表示为向量外, 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。

数据集中 N 个输入 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 经过合适转换后, 按行排成一个 $N \times D$ 的二元数组, 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, 每个输入实例 \mathbf{x}_n 有与之相关联的输出标签 \mathbf{y}_n 。这时, 数据集被写为一组实例标签对或输入输出对: $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$, 也称为样本或样本点。图1.21表示一维输入 x 和对应标签 y 的实例。

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 章中详细介绍向量和矩阵的基本概念和运算。

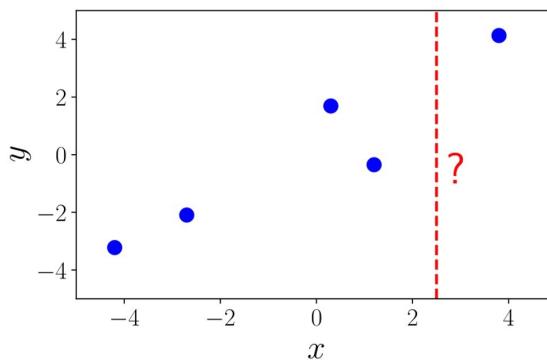


图 1.21: 线性回归的实例数据 (x_n, y_n) :
 $\{(-4.200, -3.222), (-2.700, -2.093),$
 $(+0.300, +1.690), (+1.200, -0.348),$
 $(+3.800, +4.134)\}$, 注意 $x = 2.5$ 处的函数值不属于训练数据

注记 1. 因为数据用向量表示, 所以可以处理数据来更好的表示数据。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 $\phi(x_n)$ 来表示 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, 但可以附加上一个图结构。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标记, 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, 则对它们不予区分; 有时假设输入空间与特征空间为不同的空间, 则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 $P(z|x)$, 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。

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, 且给定特定输入实例 (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1.22表示一个可用于计算输入值 x 的预测值的函数。

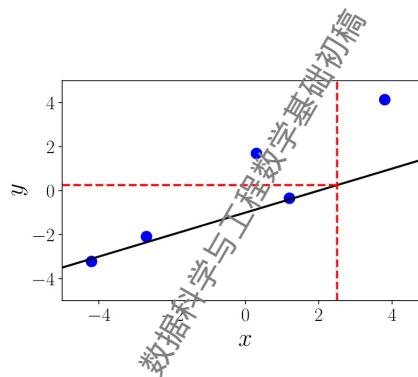


图 1.22: 实例函数在 $x = 2.5$ 时的预测: $f(2.5) = 0.25$.

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。

例 1.3.1. 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1.22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。

例 1.3.2. 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 1.23 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。

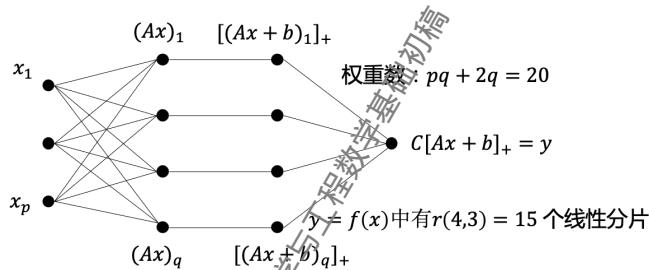


图 1.23: 数据向量 \mathbf{x} 的分段线性函数的神经网络构造

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; 图的描述, 也即概率图模型。

图 1.24 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。

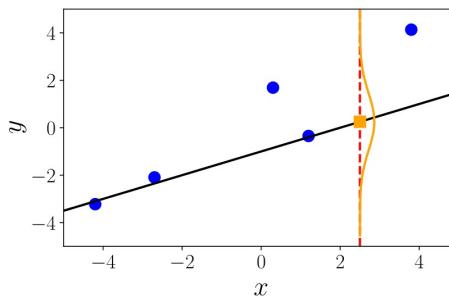


图 1.24: 实例函数 (黑色实心对角线) 及其在 $x=2.5$ 时的预测不确定性 (绘制为高斯分布)

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 $P(Y|X)$ 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。因为假设空间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

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

（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。

损失函数越小，模型越好。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 X 、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

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

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会产生“过拟合”现象。这时需要采用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, 结构风险最小化准则和正则化, 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

训练机器学习模型的目的是处理未知测试数据。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集和测试集。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, 模型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 $\mathbf{R}_{exp}(f)$ 时, 会发生过拟合现象。

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惩罚项。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:

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

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。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：

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

其中， $\| w \|$ 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：

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

其中， $\| w \|$ 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。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。

例 1.3.3. 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：

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

例 1.3.4. 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，计算原来位置到新位置的距离作为信息损失（即原来位置到 1 维直线的距离）可得：

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

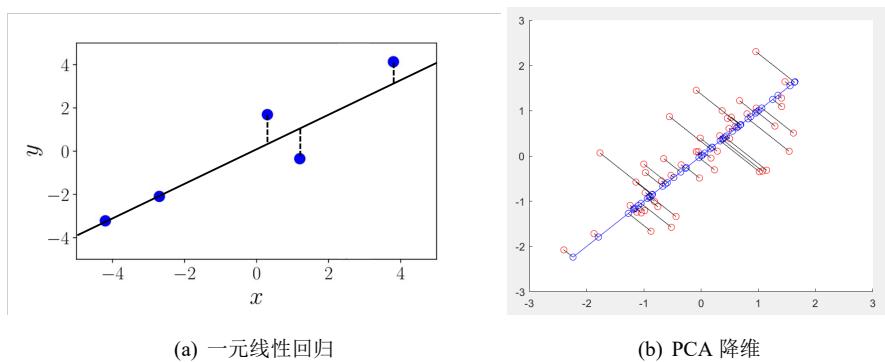


图 1.25: 监督学习与无监督学习

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。若验证集有足够的数据，用它对模型进行选择是有效的。当数据是不充足，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；多次把给定的数据按比例进行切分，每次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

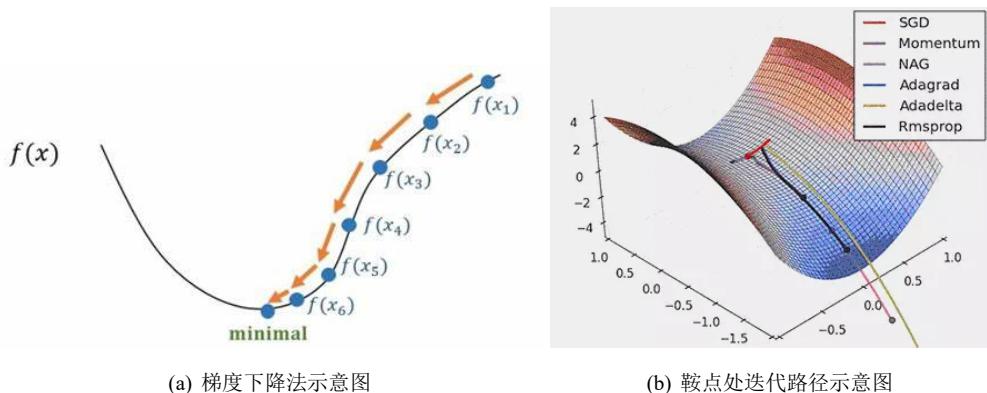


图 1.26: 优化算法

度寻找方向迭代并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：

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

图1.26(a)给出了梯度下降示意图，图1.26(b)给出了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。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。

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。

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

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，泛化误差就是所学习到的模型的期望风险。

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

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：(1) 它是样本容量的函数，当样本容量增加时，泛化上界趋于 0；(2) 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。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。通过带有标签训练集训练出来一个模型，用于判断新输入数据的类型。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。分类可以是二分类问题（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，回归运算是求该方程的回归系数。

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，从而达到对因变量的预测，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，若不知道数据之间的关系，也不知道可以分为多少类。则可以使用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 $f: \mathbf{x} \mapsto \mathbf{y}$ ，其中 \mathbf{x} 是原始数据点的表达，目前多使用向量表达形式， \mathbf{y} 是数据点映射后的低维向量表达，通常 \mathbf{y} 的维度小于 \mathbf{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，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。

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 \mathbf{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 算法属于无监督学习方法。

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1.27所示）。

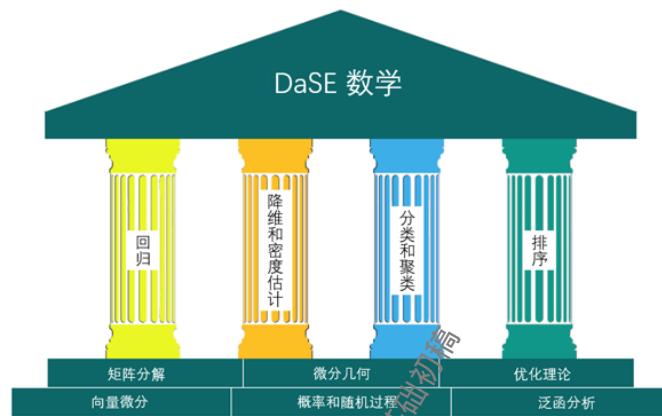


图 1.27: 数据科学与工程的数学基础

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1.28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。最后是优化理论，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，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。

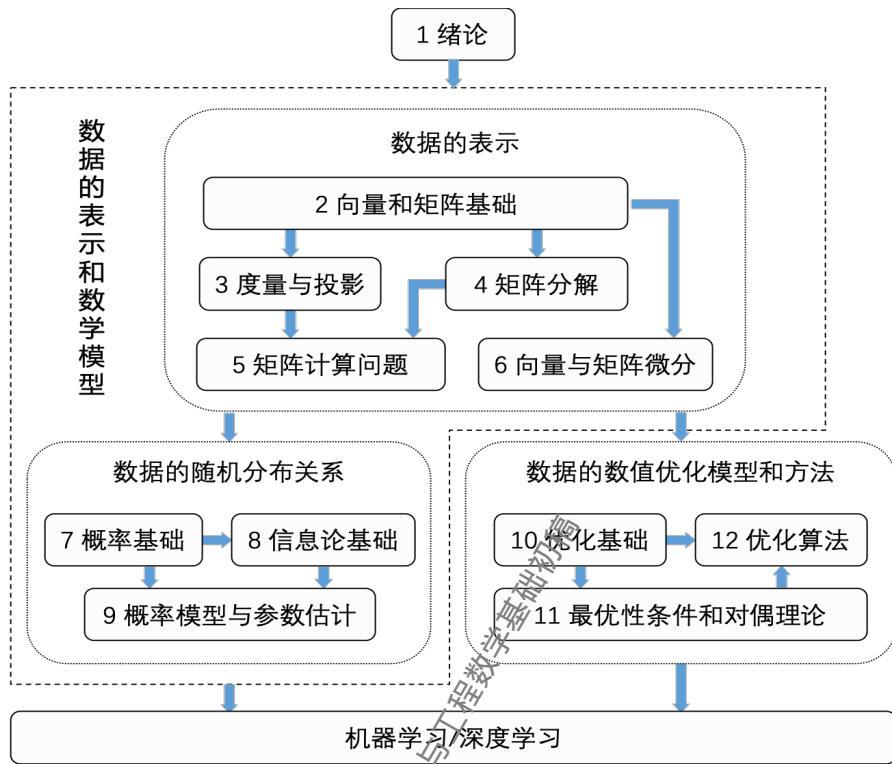


图 1.28: 数据科学与工程的数学基础内容组织结构流程图

1.4.1 数值线性代数简介

1.4.2 概率与信息论简介

1.4.3 最优化简介

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1.29是16世纪出版物中的 9×9 矩阵。

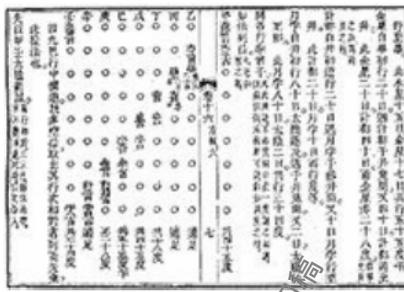


图 1.29: 古代中国的线性代数文本

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。H. Markowitz 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.

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，可以看作空间中的一个点。通过横向或纵向的排列方式，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，为助我们理解这一运算的性质。

本章的内容概览图如下：

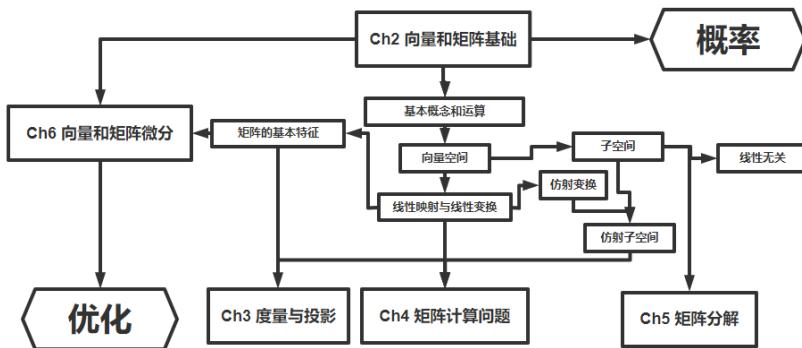


图 2.1: 本章导图

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。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，即一个球需要 4 个数来刻画。而确定一个刚体位置则需要 6 个数来表示。

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。特别地，在数据科学、模式分析、人工智能和机器学习的语境下，我们可以粗略地称之为数据向量。

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.

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。

向量的定义

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。例如在例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, 但也有着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 $n \leq 2$ 的向量是向量的特殊情形, 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。

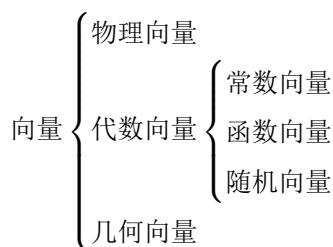
(2) 几何向量: 为了将物理向量可视化, 常用带方向的 (简称“有向”) 线段表示。这种有向线段称为几何向量。例如, $\mathbf{v} = \overrightarrow{AB}$ 表示的有向线段, 其起点为 A , 终点为 B 。

(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

矩阵在线性代数中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

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

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 $[0, 1]$ 中为“double”类型值，其中 0 表示黑色，1 表示白色）。

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。

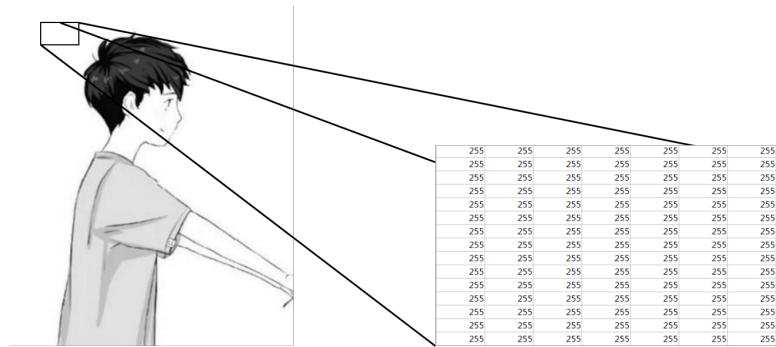


图2.2: 图像的表示

矩阵的定义

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

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

称为 \mathbb{K} 上的 $m \times n$ 矩阵，记为 $A = (a_{ij})_{m \times n}$ 或 $A_{m \times n}$ ，表中的每一个数都称为矩阵 A 的一个元素。若 $m = n$ ，则 $n \times n$ 矩阵 $A = (a_{ij})_{m \times n}$ 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，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 A 的每个元素是实数，也即 $a_{ij} \in \mathbb{R}$ 时，则称 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 A 的每个元素是复数，也即 $a_{ij} \in \mathbb{C}$ 时，则称 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：矩阵的元素为函数。
- (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 的行数相同时, 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, $x = (x_1, x_2, \dots, x_n)^T$ 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 $b = Ax$ 。

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

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) $IA = A$,
- (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由一些小矩阵组成的。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 \mathbf{A}, \mathbf{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 \mathbf{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 \mathbf{A} 作一初等行变换就相当于在 \mathbf{A} 左边乘上相应的 $m \times m$ 的初等矩阵; 对 \mathbf{A} 作一初等列变换就相当于在 \mathbf{A} 右边乘上相应的 $n \times n$ 的初等矩阵。

例 2.1.5. 设矩阵 \mathbf{A}

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

对矩阵 \mathbf{A} 作一初等行变换 (如: 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 的行列式不为 0, 有关行列式的概念将在 2.4 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:

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

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

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“等价”这一概念:

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存在初等矩阵 $P_1, \dots, P_l, Q_1, \dots, Q_t$, 使得

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

由此可得如下定理:

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

$$A = P_1 P_2 \cdots P_m. \tag{2.1}$$

将 (2.1) 式改写一下, 有

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

因此, 有如下结论。

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

$$(\mathbf{A} \ \mathbf{I}) = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 3 & 4 & 0 & 1 \end{pmatrix} \xrightarrow{\text{初等行变换}} \begin{pmatrix} 1 & 2 & 1 & 0 \\ 0 & -2 & -3 & 1 \end{pmatrix} \xrightarrow{\text{初等行变换}} \begin{pmatrix} 1 & 0 & -2 & 1 \\ 0 & -2 & -3 & 1 \end{pmatrix} \xrightarrow{\text{初等行变换}} \begin{pmatrix} 1 & 0 & -2 & 1 \\ 0 & 1 & \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

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} 各元素的行、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。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. 某食堂烹饪菜肴 F_1, \dots, F_n , 其需要的食材分别为 C_1, \dots, C_m 。为了烹饪一单位的菜肴 F_j 需要 a_{ij} 单位的食材 C_i , 其中 $i = 1, \dots, m; j = 1, \dots, n$ 。目的是找到最佳的烹饪计划, 也即如果有 b_i 单位的 C_i 可供使用, 那么应该烹饪多少 (设为 x_j) 单位的菜肴 F_j 使得恰好用尽资源。如果我们烹饪 x_1, \dots, x_n 单位的对应菜肴, 我们一共需要 $a_{11}x_1 + \dots + a_{1n}x_n$ 单位食材 C_i 。最优烹饪计划 $(x_1, \dots, x_n) \in \mathbb{R}^n$, 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

$$Ax = b \tag{2.2}$$

我们称(2.2)为线性方程组的一般形式, 并且 x_1, \dots, x_n 是该线性方程组的未知数。满足(2.2)的每个 n 维向量 $(x_1, \dots, x_n) \in \mathbb{R}^n$ 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)的解集和求解方法会在 5.1 节中介绍。

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

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

使用行初等变换法求解这个线性方程组。即

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

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：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。

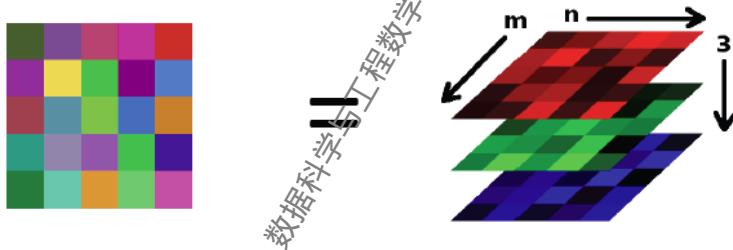


图 2.3: 彩色图像的张量表示

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时，称为数据的低维结构表示；用张量来表示时，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 2.3 节接下来要讲的向量空间之间的线性映射和 6.1 节中向量的矩阵函数密切相关。

2.2 向量空间

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，比如一些文本和一些图像都可以构成集合，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乘这两种运算。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

例 2.2.1. 我们首先来看一个鸢尾花 (*Iris*) 数据集降维的具体例子。鸢尾花数据集¹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 (*sepal length*)、花萼宽度 (*sepal width*)、花瓣长度 (*petal length*)、花瓣宽度 (*petal width*) 4 个特征预测鸢尾花会属于 *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，将 4 维的鸢尾花数据用 PCA 降维至 2 维并将其可视化，其中红、绿、蓝三种颜色的数据点分别表示为 *Setosa*, *Versicolour*, *Virginica*。

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

¹<http://archive.ics.uci.edu/ml/datasets/Iris>

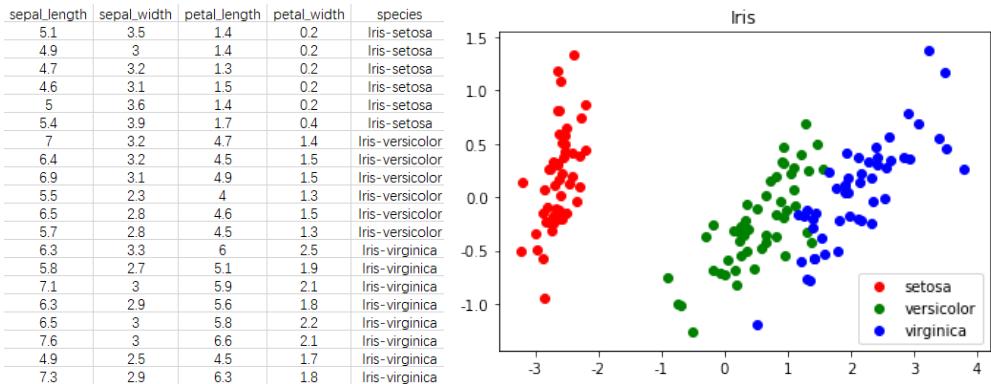


图 2.4: Iris 数据集 (左), PCA 降维可视化 (右)

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,一定性质是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} 上定义了加法, 在 \mathbb{K} 与集合 \mathbb{V} 上定义了数乘, 并且 $\forall \mathbf{a}, \mathbf{b} \in \mathbb{V}$ 及 \forall 数 $\lambda \in \mathbb{K}$, 有 $\mathbf{a} + \mathbf{b}, \lambda \mathbf{a} \in \mathbb{V}$, 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 2.2.2. 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 2.2.3. 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 2.2.4. 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 2.2.5. 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 向量空间

假设一个“大”集合是一个线性空间或者向量空间, 如果有一个它所包含的“小”集合仍然是一个向量空间, 则该“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 2.2.6. 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 2.2.7. 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 $\mathbf{0}$ 。

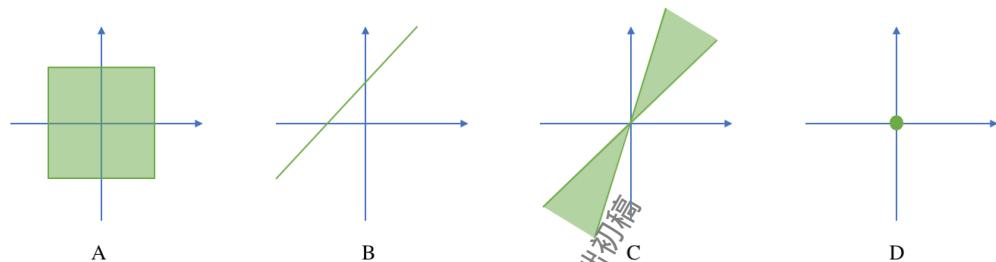


图 2.5: \mathbb{R}^2 中的一些子空间

例 2.2.8. $\mathbb{V} = \{x | a^T x = 0, x \in \mathbb{R}^n\}$ 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 2.2.9. 设 $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{k \times n}, b \in \mathbb{R}^m, b \neq 0$,

(1) 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,

证明. $\forall x_1, x_2 \in \mathbb{R}^n$, 满足 $Ax_1 = \mathbf{0}, Ax_2 = \mathbf{0}$, 则 $A(ax_1 - cx_2) = aAx_1 + cAx_2 = \mathbf{0}$ 。 \square

(2) 方程组 $Ax = b$ 的解空间 $\mathbb{V} = \{x | Ax = b, x \in \mathbb{R}^n\}$, 不是 \mathbb{R}^n 中的子空间,

证明. 设 $x \in \mathbb{R}^n$, 满足 $Ax = b$, 但 $A(x - x) = 0$, 故不是子空间。 \square

(3) 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。

证明. 对 $\forall x_1, x_2 \in \mathbb{R}^n$, 满足 $Ax_1 = \mathbf{0}, Ax_2 = \mathbf{0}$, 且 $Bx_1 = \mathbf{0}, Bx_2 = \mathbf{0}$, 则 $A(ax_1 - cx_2) = aAx_1 - cAx_2 = \mathbf{0}, B(ax_1 - cx_2) = aBx_1 + cBx_2 = \mathbf{0}$ \square

子空间的并集仍是子空间吗？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。

由定义可以看出，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 2.2.10. 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} 可唯一地表示成 $\mathbf{x} = \mathbf{y}_1 + \mathbf{y}_2$ ($\mathbf{y}_1 \in \mathbb{Y}_1, \mathbf{y}_2 \in \mathbb{Y}_2$) 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 $\mathbf{y}_1 = \mathbf{y}_2 = \mathbf{0}$ 。

推论 2.2.1. 和 $\mathbb{Y}_1 + \mathbb{Y}_2$ 为直和的必要充分条件是： $\mathbb{Y}_1 \cap \mathbb{Y}_2 = \{\mathbf{0}\}$ 。

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

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

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

和两个子空间的直和一样，我们有

定理 2.2.3. $\mathbb{Y}_1, \mathbb{Y}_2, \dots, \mathbb{Y}_s$ 是 \mathbb{Y} 的一些子空间，下面这些条件是等价的：

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

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

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

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

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

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 2.2.11. 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 2.2.12. 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 $\sum_{i=1}^r \lambda_i \mathbf{a}_i = 0$, 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 $\sum_{i=1}^r \lambda_i \mathbf{a}_i = 0$, 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 2.2.13. 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 ($\mathbf{a}_2 \neq \mathbf{a}_1 + \mathbf{a}_3$)。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 2.2.14. 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, 因为

$$\mathbf{a}_1 = \frac{1}{2} \mathbf{b}_1 - \frac{1}{2} \mathbf{b}_2, \quad \mathbf{a}_2 = \frac{1}{2} \mathbf{b}_1 + \frac{1}{2} \mathbf{b}_2;$$

$$\mathbf{b}_1 = \mathbf{a}_1 + \mathbf{a}_2, \quad \mathbf{b}_2 = -\mathbf{a}_1 + \mathbf{a}_2;$$

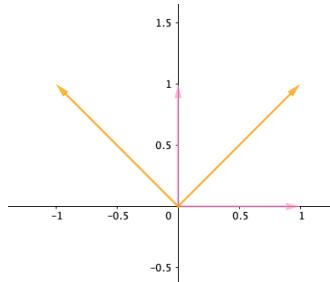


图 2.6: 向量组等价

2.2.5 生成集、基底与坐标

在例 2.2.14 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 $\text{span}(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r)$ 的一个生成集。

定理 2.2.4. 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, 也就是在这 3 个向量的生成集中存在某个向量可以用其它 3 个向量线性表出。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它 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 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$, 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 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ 就是 \mathbb{V} 的一组基。

例 2.2.15. 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 2.2.5. 令 \mathbb{V} 是一向量空间, $\mathbb{B} \subseteq \mathbb{V}, \mathbb{B} \neq \emptyset$, 则下列命题等价:

- \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 3.1 节中, 我们将会严格说明向量的长度。

例 2.2.16. 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 2.2.17. 对于一个由向量 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 张成的向量空间 $\mathbb{U} \subseteq \mathbb{R}^4$

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 1 \end{pmatrix},$$

我们关心 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 是否是 \mathbb{U} 的一组基。为此, 我们需要确认 $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 是否线性无关。因此, 我们需要解 $\sum_{i=1}^3 \lambda_i \mathbf{x}_i = \mathbf{0}$ 。

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

$$(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

从而我们可以发现 $\mathbf{x}_1, \mathbf{x}_2$ 是线性无关的, $\lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 = \mathbf{0}$ 当且仅当 $\lambda_1 = \lambda_2 = 0$ 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 2.2.6. 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{X} 的一个子空间，且

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

我们给出两个子空间和的维数与各自维数的关系。

定理 2.2.7. 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，几何空间中向量所张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 $1, \mathbf{x}, \dots, \mathbf{x}^{n-1}$ 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，对无限维空间也成立。在本书中，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 2.2.18. 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 $f(x) = a_0 + a_1 x + \dots + a_{n-1} x^{n-1}$ 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

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

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 2.2.19. 在 n 维向量空间 \mathbb{V} 中, 显然

$$\left\{ \begin{array}{l} \varepsilon_1 = (1, 0, \dots, 0), \\ \varepsilon_2 = (0, 1, \dots, 0), \\ \vdots \\ \varepsilon_n = (0, 0, \dots, 1) \end{array} \right.$$

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

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

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,

$$\left\{ \begin{array}{l} \varepsilon_1' = (1, 1, \dots, 1), \\ \varepsilon_2' = (0, 1, \dots, 1), \\ \vdots \\ \varepsilon_n' = (0, 0, \dots, 1) \end{array} \right.$$

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

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

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

例 2.2.20. 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 2.2.9. 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

$$PAQ = \begin{pmatrix} I_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$$

由于初等行列变换都是可逆的,

$$A = P^{-1} \begin{pmatrix} I_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} Q^{-1}$$

取 P^{-1} 的前 r 列作为 F , Q^{-1} 的前 r 行作为 G , 即为满足条件 $A = FG$ 的一组矩阵。 \square

这样的矩阵分解并不是唯一的。可以对 F 右乘可逆矩阵 X , G 左乘 X^{-1} , 则 $F_1 = FX$, $G_1 = X^{-1}G$ 也是满足条件的一组矩阵。

$A = FG$, 也就是说

$$A = f_1g_1 + f_2g_2 + \dots + f_rg_r$$

其中 f_i 是 \mathbf{F} 的第 i 列, \mathbf{g}_i 是 \mathbf{G} 的第 i 行。 $f_i \mathbf{g}_i$ 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。关于秩-1 矩阵我们在 4.1 节会详细介绍。

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。如图2.7所示。

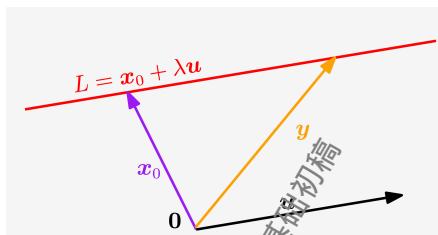


图 2.7: 仿射子空间

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

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

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 $\mathbf{x}_0 + k_1 \mathbf{a}_1 + k_2 \mathbf{a}_2 + \dots + k_m \mathbf{a}_m$, 这一结论由定义可直接获得。

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. 已知线性方程组 $\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{b} \neq \mathbf{0}$ 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 $\mathbf{A}\mathbf{x} = \mathbf{0}$ 的解空间为 \mathbb{V} , 它是一个子空间; 设 \mathbf{x}_0 是 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 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,

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

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来表达语义的相似度。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、正交性和距离(包括余弦距离)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。

下面我们将从映射的定义出发，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 线性映射：线性模型的观点

线性映射的定义

定义 2.3.1. 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 2.3.1. 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 2.3.2. 考察映射 $\mathcal{T}: \mathbb{R}^3 \rightarrow \mathbb{R}^2$

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

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

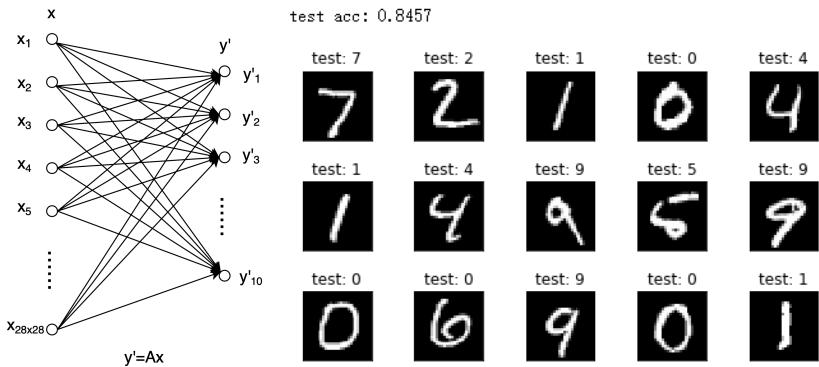


图 2.8: 线性映射分类准确率

容易看出, 映射 $T_1 : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ 不满足线性关系式, 故不是线性映射; 而映射 $T_2 : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ 满足线性关系式, 为线性映射。

例 2.3.3. 考虑映射 $T_Q(A) = Q^{-1}AQ$, 其中 Q 是可逆矩阵。因为

$$\begin{aligned} T_Q(\alpha A + \beta B) &= Q^{-1}(\alpha A + \beta B)Q \\ &= \alpha Q^{-1}AQ + \beta Q^{-1}BQ \\ &= \alpha T_Q(A) + \beta T_Q(B) \end{aligned}$$

所以 $T_Q(A) = Q^{-1}AQ$ 是关于 A 的线性映射。

例 2.3.4. 在 MNIST 数字识别的例 2.3.3 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 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} , MNIST 数字识别就是想要找到一个映射, 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 x , 其标签向量为 y , 通过我们选择的线性映射

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

$$y' = Ax$$

其中 y' 也是 \mathbb{V} 中的向量, 希望 $y' \approx y$ 。选择合适的损失函数, 通过优化得到 A 。图 6.10(e)给出了线性映射分类手写数字的方程解。

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 2.3.2. 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 2.3.3. 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

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

例 2.3.5. 根据逆映射的定义,

- 恒等映射的逆映射是其本身;
- 在例 2.3.3 中定义了映射 $\mathbf{T}_Q(A) = Q^{-1}AQ$ 的逆映射为 $\mathbf{T}_{Q^{-1}}$ 。

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

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

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。

例 2.3.7. 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, 但是如果我们将利用线性映射的复合构造分类器, 即

$$\begin{aligned} \mathbf{h}_1 &= f_1(\mathbf{x}) = \mathbf{A}_1\mathbf{x} \\ \mathbf{y}' &= f_2(\mathbf{h}_1) = \mathbf{A}_2\mathbf{h}_1 \end{aligned}$$

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

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

得到的仍然是一个线性映射, 并不能提高分类的准确性。图 6.10(b)给出线性映射与线性映射复合后分类手写数字的准确率, 与图 6.10(a)相比, 可以看出准确率并没有大的提升。

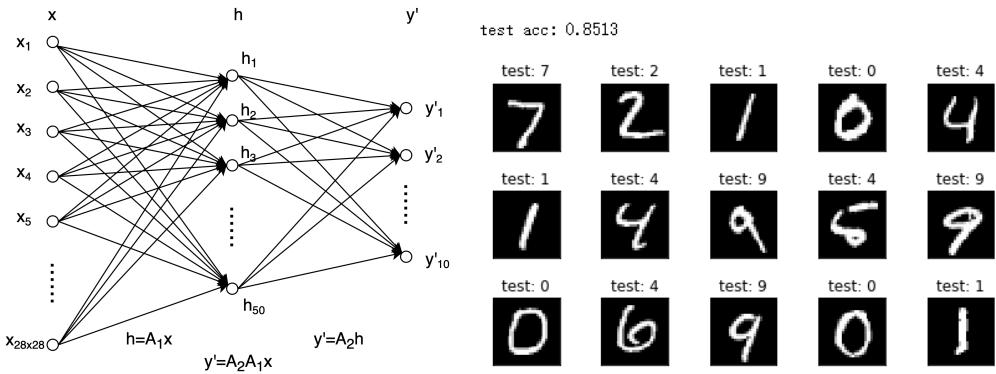


图 2.9: 双层线性网络准确率

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。

2.3.2 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 2.3.4. 考虑一个 n 维向量空间 \mathbb{V} 的基底 $\{b_1, \dots, b_n\}$, $B = (b_1, \dots, b_n)$ 称为 \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 $B = (b_1, \dots, b_n)$, 任何的 $x \in \mathbb{V}$, 我们得到 x 关于 B 的唯一表示 (线性组合)

$$x = \alpha_1 b_1 + \dots + \alpha_n b_n$$

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

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

是 x 关于 B 的坐标表示, 即 $x = B\alpha$ 。

例 2.3.8. 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。如图2.10所示:

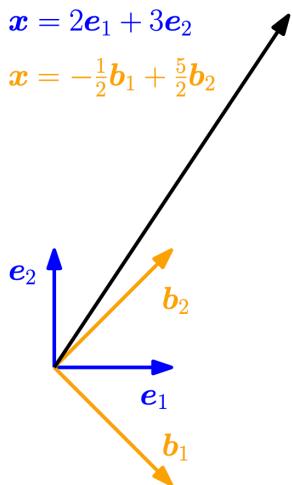


图 2.10: 不同基下的坐标

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 2.3.5. 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 $j \in \{1, \dots, n\}$

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

$\Phi(\mathbf{b}_j)$ 是关于 \mathbf{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。 $\Phi(\mathbf{b}_j)$ 在 \mathbb{W} 有序基 \mathbf{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

$$\Phi(\mathbf{B}) = \mathbf{C}A_\Phi.$$

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 $\hat{\mathbf{y}}$:

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

$A_\Phi\hat{\mathbf{x}}$ 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 2.3.9. 考虑同态 $\Phi : \mathbb{V} \rightarrow \mathbb{W}$, \mathbb{V} 的有序基 $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3)$, \mathbb{W} 的有序基 $\mathbf{C} = (\mathbf{c}_1, \dots, \mathbf{c}_4)$, 有:

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

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 $\Phi(\mathbf{b}_j)$ 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

$$\begin{aligned}\Phi : \mathbb{V} &\rightarrow \mathbb{W} \\ \Psi : \mathbb{W} &\rightarrow \mathbb{X} \\ \Psi \circ \Phi : \mathbb{V} &\rightarrow \mathbb{X}\end{aligned}$$

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 2.3.10. 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

$$\begin{aligned}\mathbf{A}_\Phi &= 3\mathbf{I} \\ \mathbf{A}_\Psi &= (1, 1, \dots, 1)\end{aligned}$$

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

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

接下来, 我们研究改变 \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 \mathbf{B} , \mathbb{W} 的基底是 \mathbf{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 2.3.11. 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

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

这便简化了 \mathbf{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 \mathbf{A}_Φ , 在基 $\tilde{\mathbf{B}}$ 与 $\tilde{\mathbf{C}}$ 下的变换矩阵为 $\tilde{\mathbf{A}}_\Phi$ 。则有

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

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

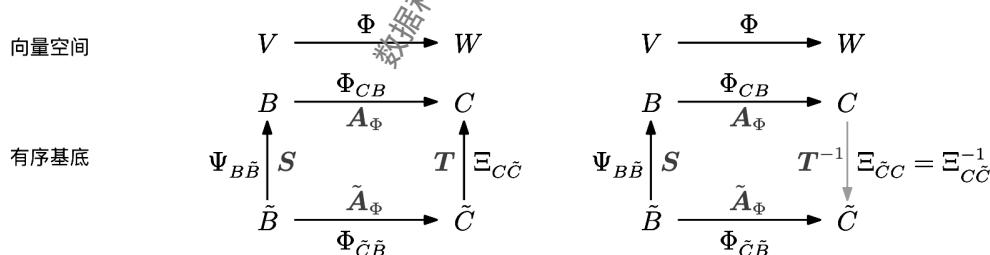


图 2.11: 不同基底下的坐标关系

当我们分别在 \mathbb{V} 中把基底从 \mathbf{B} 变换为 $\tilde{\mathbf{B}}$ 以及在 \mathbb{W} 中把基底从 \mathbf{C} 变换为 $\tilde{\mathbf{C}}$, 我们可以通过一些步骤来得到相应的变换矩阵 $\tilde{\mathbf{A}}_\Phi$ 。如图2.11所示。

- 首先, 我们写出关于新基底 $\tilde{\mathbf{B}}$ 下坐标和旧基底 \mathbf{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 \mathbf{A}_Φ 将坐标映射到以 \mathbf{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:

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

图2.11描述了不同基底下的坐标关系。

定义 2.3.6. 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 2.3.7. 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 2.3.12. 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

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

我们寻找一个新的基下的 Φ 的变换矩阵。令新的基分别为

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

所以

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

由定理2.3.3可得新基底的变换矩阵如下:

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

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

线性映射的像与核是两个重要的线性子空间。

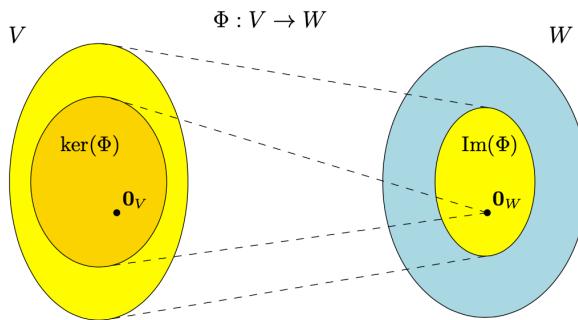


图 2.12: 核空间与像空间

定义 2.3.8. 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) :

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

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 $\Phi(\mathbf{0}_V) = \mathbf{0}_W$, 因此 $\mathbf{0}_V \in \ker(\Phi)$; 也就是说零空间永远非空。
- $Im(\Phi) \subseteq \mathbb{W}$ 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 $\ker(\Phi) = \mathbf{0}$;
- $\text{rank}(\mathbf{A}) = \dim(Im(\Phi))$;
- Φ 的核空间是方程 $A\mathbf{x} = \mathbf{0}$ 的解空间。

定义 2.3.9. \mathbf{A} 的列向量张成的空间叫做列空间。

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

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

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

所以 Φ 的像空间是可以由 \mathbf{A} 的列向量张成的。

例 2.3.13. 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 $Ax = 0$ 。

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

最终我们可以给出

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

关于列空间的定义我们在 3.2 节会进一步详细介绍。

2.3.3 线性变换

线性变换的定义

在本小节中, 我们主要讨论向量空间 V 到自身的映射, 称为 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 2.3.10. 设 V 是数域 \mathbb{K} 上的向量空间, 如果对任何向量 $x, y \in V$ 及任意的 $\alpha, \beta \in \mathbb{K}$, 有

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

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

例 2.3.14. 下列线性映射是线性变换:

- 恒等映射 $\epsilon: V \rightarrow V, \epsilon(x) = x$ 。
- 例 2.3.3 中的线性映射 $T_Q(A) = Q^{-1}AQ$, 其中 Q 是可逆矩阵。我们称其为矩阵的相似变换。

设 V 是数域 \mathbb{K} 上的 n 维向量空间, $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 是 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 V 中任一向量 ξ 可以被基 $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ 线性表出, 即有

$$\varepsilon = x_1 \varepsilon_1 + x_2 \varepsilon_2 + \dots + x_n \varepsilon_n$$

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 $A\xi$ 与基的像 $A\epsilon_1, A\epsilon_2, \dots, A\epsilon_n$ 之间也存在：

$$\begin{aligned} A\xi &= A(x_1\epsilon_1 + x_2\epsilon_2 + \dots + x_n\epsilon_n) \\ &= x_1A(\epsilon_1) + x_2A(\epsilon_2) + \dots + x_nA(\epsilon_n). \end{aligned} \quad (2.3)$$

上式表明，如果我们知道了基 $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ 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 $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ 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

$$A\epsilon_i = B\epsilon_i, \quad i = 1, 2, \dots, n,$$

那么 $A = B$ 。

证明. A 与 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 ξ ，等式 $A\xi = B\xi$ 成立，由 (2.3) 得，

$$\begin{aligned} A\xi &= x_1A(\epsilon_1) + x_2A(\epsilon_2) + \dots + x_nA(\epsilon_n) \\ &= x_1B(\epsilon_1) + x_2B(\epsilon_2) + \dots + x_nB(\epsilon_n) = B\xi. \end{aligned}$$

□

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 $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ 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

$$A\epsilon_i = a_i, \quad i = 1, 2, \dots, n. \quad (2.4)$$

证明. 我们来作出所要的线性变换，设

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

是向量空间 \mathbb{V} 的任意一个向量，我们定义 \mathbb{V} 的变换 A 为

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

下面先来证明变换 A 是线性的。

在 \mathbb{V} 中任取两个向量，

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

于是

$$\begin{aligned}\mathbf{b} + \mathbf{c} &= \sum_{i=1}^n (b_i + c_i) \boldsymbol{\varepsilon}_i, \\ k\mathbf{b} &= \sum_{i=1}^n kb_i \boldsymbol{\varepsilon}_i, k \in \mathbb{K}\end{aligned}$$

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

$$\boldsymbol{\varepsilon}_i = 0\boldsymbol{\varepsilon}_1 + \cdots + 0\boldsymbol{\varepsilon}_{i-1} + 1\boldsymbol{\varepsilon}_i + 0\boldsymbol{\varepsilon}_{i+1} + \cdots + 0\boldsymbol{\varepsilon}_n, i = 1, 2, \dots, n,$$

所以

$$\mathcal{A}\boldsymbol{\varepsilon}_i = 0\mathbf{a}_1 + \cdots + 0\mathbf{a}_{i-1} + 1\mathbf{a}_i + 0\mathbf{a}_{i+1} + \cdots + 0\mathbf{a}_n = \mathbf{a}_i, i = 1, 2, \dots, n.$$

□

结合以上两点, 则有

定理 2.3.6. 设 $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \dots, \boldsymbol{\varepsilon}_n$ 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

$$\mathcal{A}\boldsymbol{\varepsilon}_i = \mathbf{a}_i, \quad i = 1, 2, \dots, n.$$

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 2.3.11. 设 $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \dots, \boldsymbol{\varepsilon}_n$ 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:

$$\left\{ \begin{array}{l} \mathcal{A}\boldsymbol{\varepsilon}_1 = a_{11}\boldsymbol{\varepsilon}_1 + a_{21}\boldsymbol{\varepsilon}_2 + \cdots + a_{n1}\boldsymbol{\varepsilon}_n, \\ \mathcal{A}\boldsymbol{\varepsilon}_2 = a_{12}\boldsymbol{\varepsilon}_1 + a_{22}\boldsymbol{\varepsilon}_2 + \cdots + a_{n2}\boldsymbol{\varepsilon}_n, \\ \vdots \\ \mathcal{A}\boldsymbol{\varepsilon}_n = a_{1n}\boldsymbol{\varepsilon}_1 + a_{2n}\boldsymbol{\varepsilon}_2 + \cdots + a_{nn}\boldsymbol{\varepsilon}_n. \end{array} \right.$$

用矩阵来表示就是

$$\begin{aligned}\mathcal{A}(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n) &= (\mathcal{A}\varepsilon_1, \mathcal{A}\varepsilon_2, \dots, \mathcal{A}\varepsilon_n) \\ &= (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)\mathcal{A}\end{aligned}$$

其中

$$\mathcal{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}.$$

矩阵 \mathcal{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 \mathcal{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

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \mathcal{A} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

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

$$\begin{aligned}\mathcal{A}\xi &= (\mathcal{A}\varepsilon_1, \mathcal{A}\varepsilon_2, \dots, \mathcal{A}\varepsilon_n) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \\ &= (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n) \mathcal{A} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.\end{aligned}$$

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

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = A \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}.$$

□

伸缩与旋转

我们先来看一个例子:

例 2.3.15. 考虑线性变换 \mathcal{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 2.13 所示。

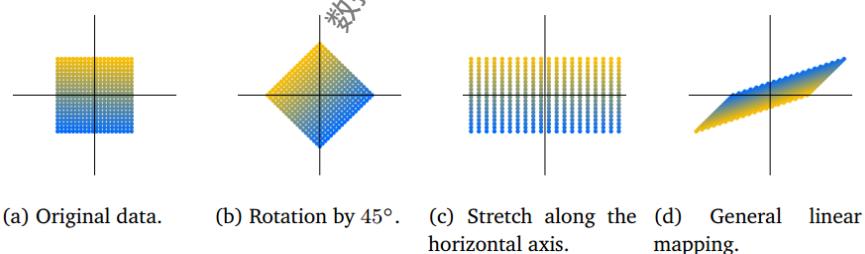


图 2.13: 线性变换对原数据的影响

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 将会对原数据的横坐标进行 2 倍拉伸, 矩阵 A_3 融合了旋转和拉伸操作。

2.3.4 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 2.3.12. 设两个线性空间 \mathbb{V}, \mathbb{W} 与一个线性映射 $\Phi: \mathbb{V} \rightarrow \mathbb{W}$,

$$\phi(\mathbf{x}) = \mathbf{a} + \Phi(\mathbf{x})$$

是一个仿射映射, 又称仿射变换。其中映射 $\phi: \mathbb{V} \rightarrow \mathbb{W}$ 且 $\mathbf{a} \in \mathbb{W}$ 。

例 2.3.16. 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

$$\mathbb{W} = \mathbf{A}\mathbf{c} + \mathbf{b} + L(\mathbf{A}\mathbf{d})$$

例 2.3.17. 考虑如何将图 2.14 旋转一个角度。

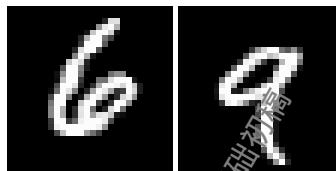


图 2.14: MNIST 数据集中的数字图片

以原图像的点 $\mathbf{s} = (s_1, s_2)$ 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 $\mathbf{d} = (d_1, d_2)$, 逆时针旋转角度为 δ 。则一个像素的原位置 $\mathbf{x} = (x_1, x_2)$ 与旋转后的目标位置 $\mathbf{y} = (y_1, y_2)$ 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。

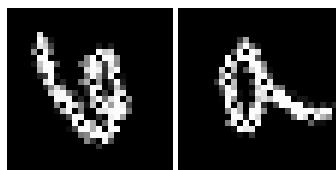


图 2.15: 旋转后的结果

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。由上述的像素原始位置与旋转后的目标位置的关系公式并利用矩阵的逆, 可得到变化后的目标像素 $\mathbf{y} = (y_1, y_2)$ 的原坐标 $\mathbf{x} = (x_1, x_2)$ 为:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \cos \delta & \sin \delta \\ -\sin \delta & \cos \delta \end{pmatrix} \begin{pmatrix} y_1 - d_1 \\ y_2 - d_2 \end{pmatrix} + \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$$

旋转后的图像如图2.15所示。

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，参见图6.10(c)的分类准确率。同样，由于多个仿射函数的复合函数仍然是仿射函数，如图6.10(d)的分类准确率所示，也不能提高模型的预测能力。

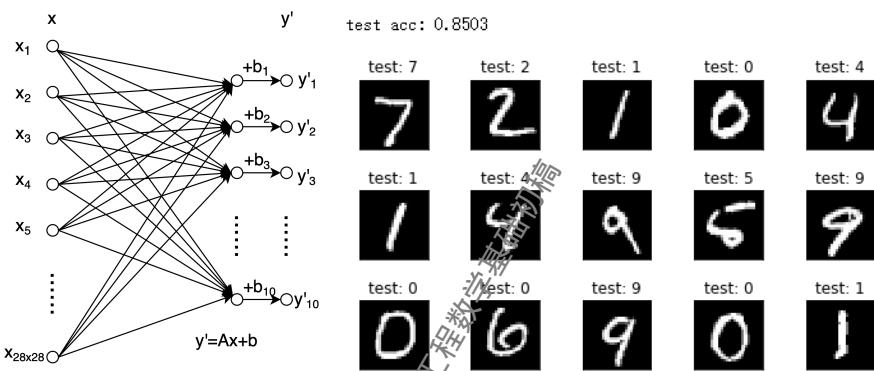


图 2.16: 仿射映射模型分类准确率

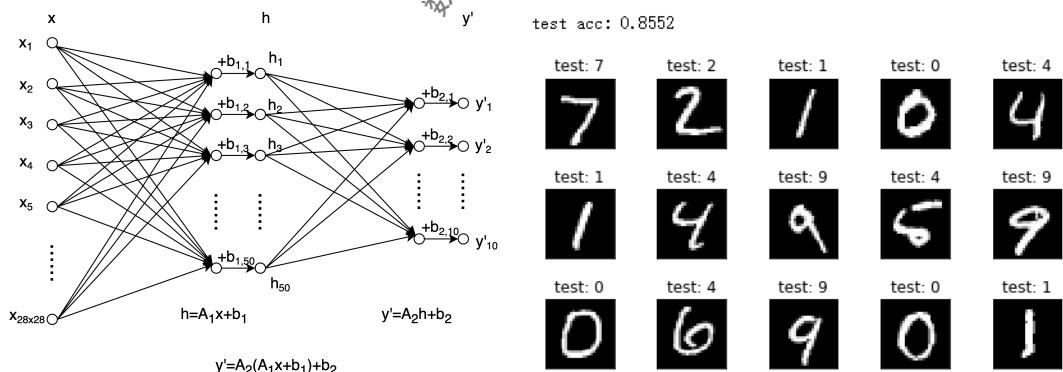


图 2.17: 仿射映射复合模型分类准确率

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中线性模型的基础，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，粗略地说，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遇到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 6.1-6.3 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。

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。

行列式的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

$$x_1 = \frac{b_1a_{22} - a_{12}b_2}{a_{11}a_{22} - a_{12}a_{21}}, x_2 = \frac{a_{11}b_2 - a_{21}b_1}{a_{11}a_{22} - a_{12}a_{21}}.$$

我们称 $a_{11}a_{22} - a_{12}a_{21}$ 为二阶行列式，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, 首先要给出 n 阶行列式的定义及性质, 这是本节的主要内容。

定义 2.4.1. n 个不同的元素排成一列, 称为这 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.

- (1) $\tau(3, 2, 1, 4) = 3$, $3, 2, 1, 4$ 为奇排列
- (2) $\tau(1, 3, 2, 4) = 1$, $1, 3, 2, 4$ 为奇排列
- (3) $\tau(3, 1, 2, 4) = 2$, $3, 1, 2, 4$ 为偶排列

定义 2.4.2. $\det(A)$ 叫做矩阵 A 的行列式, 是从 $\mathbb{R}^{n \times n}$ 映射到 \mathbb{R} 的一个函数, 其中 $A \in \mathbb{R}^{n \times n}$ 。

$$\det(A) = \sum_{j_1, j_2, \dots, j_n} (-1)^{r(j_1, j_2, \dots, j_n)} a_{1j_1} a_{2j_2} \cdots a_{nj_n}$$

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 $1 \leq i_1 < i_2 \leq n$ 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, 这两个行列式的该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上三角形行列式（下三角形行列式）。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

$$\begin{aligned} & \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & a_{nn} \end{vmatrix} = a_{11}a_{22} \cdots a_{nn}, \\ & \begin{vmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} = a_{11}a_{22} \cdots a_{nn}. \end{aligned}$$

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 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：

$$\sum_{s=1}^n a_{ks} A_{is} = \begin{cases} D & \text{当 } k = i \\ 0 & \text{当 } k \neq i \end{cases} \quad (2.5)$$

$$\sum_{s=1}^n a_{sl} A_{sj} = \begin{cases} D & \text{当 } l = j \\ 0 & \text{当 } l \neq j \end{cases} \quad (2.6)$$

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 $A = \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix}$ 的行列式是平面直角坐标系 xoy 平面上以行向量 $\mathbf{a} = (a_1, a_2)$ ， $\mathbf{b} = (b_1, b_2)$ 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 \mathbf{a} 沿逆时针方向转到 \mathbf{b} 而得到的，面积取正值；若这个平行四边形是由向量 \mathbf{a} 沿顺时针方向转到 \mathbf{b} 而得到的，面积取负值，如图2.18所示。

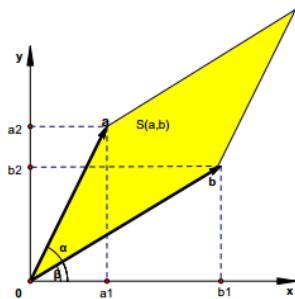


图 2.18: 二阶行列式的几何意义

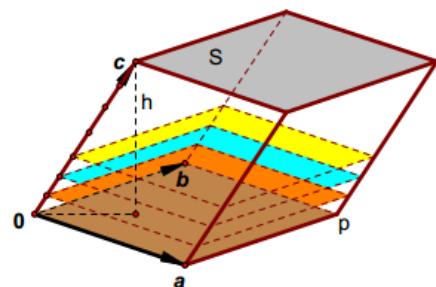


图 2.19: 三阶行列式的几何意义

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2.19，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} 则将其变成一个椭圆。

同样地，在 3 维的情形下（如2.19）， \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} 将其变成一个椭球。

一般地，一个 $n \times n$ 矩阵 \mathbf{A} 将 \mathbb{R}^n 中 n 维单位立方体变成 \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，可得矩阵是伴随矩阵的行列式的关系。

定理 2.4.4.

$$\mathbf{A}\mathbf{A}^* = \mathbf{A}^*\mathbf{A} = \begin{pmatrix} |\mathbf{A}| & 0 & \cdots & 0 \\ 0 & |\mathbf{A}| & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & |\mathbf{A}|\end{pmatrix} = |\mathbf{A}|\mathbf{I}$$

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。

性质 2.4.1. 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)

$$\text{Tr}(AB) = \sum_{i=1}^n \sum_{j=1}^m a_{ij}b_{ji} = \sum_{j=1}^m \sum_{i=1}^n b_{ji}a_{ij} = \text{Tr}(BA)$$

(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 2.4.2. 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 (逆时针方向转轴). 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。

$$\begin{cases} x = x' \cos \theta - y' \sin \theta, \\ y = x' \sin \theta + y' \cos \theta, \end{cases} \quad (2.9)$$

(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。本节介绍它的一些最基本的性质。

定义 2.4.7. 对称矩阵 (symmetric) 是其转置和自己相等的矩阵, 即

$$A = A^T$$

定义 2.4.8. 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。二次型 (2.10) 的系数排成的对称矩阵

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

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:

$$f(x_1, x_2, \dots, x_n) = \mathbf{x}^T A \mathbf{x}$$

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 2.4.9. 设 $x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_n$ 是两组向量, 系数在数域 \mathbb{K} 中的一组关系式

$$x_i = \sum_{j=1}^n c_{ij} y_j (i = 1, 2, \dots, n) \quad (2.11)$$

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 (2.11) 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

$$f(x_1, x_2, \dots, x_n) = \mathbf{x}^T A \mathbf{x}, A = A^T$$

是一个二次型, 作非退化线性替换

$$\mathbf{x} = \mathbf{C} \mathbf{y},$$

我们有

$$f(x_1, x_2, \dots, x_n) = \mathbf{x}^T A \mathbf{x} = (\mathbf{C} \mathbf{y})^T A (\mathbf{C} \mathbf{y}) = \mathbf{y}^T \mathbf{C}^T A \mathbf{C} \mathbf{y} = \mathbf{y}^T (\mathbf{C}^T A \mathbf{C}) \mathbf{y} = \mathbf{y}^T \mathbf{B} \mathbf{y},$$

即得到一个 y_1, y_2, \dots, y_n 的二次型

$$\mathbf{y}^T \mathbf{B} \mathbf{y},$$

其中

$$\mathbf{B} = \mathbf{C}^T \mathbf{A} \mathbf{C}.$$

这就是前后两个二次型的矩阵的关系。

定义 2.4.10. 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 $\mathbf{A}_2 = (\mathbf{C}_1 \mathbf{C}_2)^T \mathbf{A} (\mathbf{C}_1 \mathbf{C}_2)$.

因此我们可知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 2.4.5. 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

$$d_1 x_1^2 + d_2 x_2^2 + \dots + d_n x_n^2$$

的形式, 它称为所给二次型的标准型。

定理 2.4.5 也可等价地叙述为如下的定理 2.4.6。

定理 2.4.6. 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型。

设二次型 $f = f(x_1, x_2, \dots, x_n)$ 的矩阵为 \mathbf{A} , 作初等变换

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{I} \end{pmatrix} \xrightarrow[\text{对 } \mathbf{I} \text{ 只作其中的初等列变换}]{\text{对 } \mathbf{A} \text{ 作成对的初等行、列变换}} \begin{pmatrix} \mathbf{D} \\ \mathbf{C} \end{pmatrix}$$

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, $f = d_1 y_1^2 + d_2 y_2^2 + \dots + d_n y_n^2$ 。

例 2.4.4. 用初等变换法化二次型 $f(x_1, x_2, x_3) = x_1 x_2 + x_1 x_3 - 3x_2 x_3$ 为标准型。

解. $f(x_1, x_2, x_3)$ 的矩阵为

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 0 & -3/2 \\ 1/2 & -3/2 & 0 \end{pmatrix}, \\ \begin{pmatrix} \mathbf{A} \\ \mathbf{I} \end{pmatrix} &= \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1/2 & 0 & -3/2 \\ 1/2 & -3/2 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \xrightarrow{\text{阶梯形}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1/4 & 0 \\ 0 & 0 & 3 \\ 1 & -1/2 & 3 \\ 1 & 1/2 & -1 \\ 0 & 0 & 1 \end{pmatrix}. \\ \mathbf{D} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1/4 & 0 \\ 0 & 0 & 3 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 1 & -1/2 & 3 \\ 1 & 1/2 & -1 \\ 0 & 0 & 1 \end{pmatrix}, \end{aligned}$$

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 2.4.7. 在二次型的标准型中, 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 2.4.11. 设 $f(x_1, x_2, \dots, x_n)$ 是一实二次型, 其矩阵的秩为 r , p 代表正平方项的个数, 且标准型为

$$d_1y_1^2 + d_2y_2^2 + \dots + d_py_p^2 - d_{p+1}y_{p+1}^2 - \dots - d_r y_r^2 \quad (2.12)$$

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

$$z_1^2 + z_2^2 + \dots + z_p^2 - z_{p+1}^2 - \dots - z_r^2 \quad (2.13)$$

(2.13) 式称为实二次型 $f(x_1, x_2, \dots, x_n)$ 的规范型。

若 $f(x_1, x_2, \dots, x_n)$ 是一复二次型, 其矩阵的秩为 r , 则其规范型为

$$z_1^2 + z_2^2 + \dots + z_r^2 \quad (r \text{ 为二次型的秩})$$

定理 2.4.8. 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型, 且规范型是唯一的。

定理 2.4.8 换个说法就是, 任意复数的对称矩阵与一个形式为

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

的对角矩阵合同, 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, 其规范型完全被 r, p 这两个数所决定。

定义 2.4.12. 在实二次型 $f(x_1, x_2, \dots, x_n)$ 的规范型中, 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 $f(x_1, x_2, \dots, x_n)$ 的符号差。

正(负)定二次型

定义 2.4.13. 设 $f(x_1, x_2, \dots, x_n)$ 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 2.4.14. 设 $A = (a_{ij})_{n \times n}$, 则称 $k (k \leq n)$ 阶子式

$$P_k = \begin{vmatrix} a_{i_1 i_1} & a_{i_1 i_2} & \cdots & a_{i_1 i_k} \\ a_{i_2 i_1} & a_{i_2 i_2} & \cdots & a_{i_2 i_k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{i_k i_1} & a_{i_k i_2} & \cdots & a_{i_k i_k} \end{vmatrix}$$

为 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

$$Q_k = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{vmatrix}$$

称为 A 的 k 阶顺序主子式。

定理 2.4.9. 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) 它的规范型为 $z_1^2 + z_2^2 + \dots + z_p^2 - z_{p+1}^2 - \dots - z_r^2$ 。
- (4) \mathbf{A} 的所有顺序主子式全大于零 (\mathbf{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 2.4.15. 对于一个 $n \times n$ 矩阵 \mathbf{A} , 如果存在数 λ 和向量 \mathbf{x} 使得

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \tag{2.14}$$

则称 λ 是矩阵 \mathbf{A} 的一个特征值, 而 \mathbf{x} 是 λ 对于的矩阵 \mathbf{A} 的一个特征向量。

我们容易知道矩阵 \mathbf{A} 的特征值就是变元 λ 的 n 次多项式 $\det((\lambda\mathbf{I} - \mathbf{A}))$ 的 n 个根, 所以特征值也称特征根。而 λ 对应的特征向量 \mathbf{x} 就是齐次线性方程组 $(\lambda\mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{0}$ 的非零解向量。因此进一步引入如下术语。

定义 2.4.16. 矩阵 $\lambda\mathbf{I} - \mathbf{A}$ 称为 \mathbf{A} 的特征矩阵。多项式 $\Delta_{\mathbf{A}}(\lambda) = \det(\lambda\mathbf{I} - \mathbf{A})$ 称为 \mathbf{A} 的特征多项式。所有特征值的集合 $\lambda(\mathbf{A})$ 称为 \mathbf{A} 的谱。对于 $\lambda_0 \in \lambda(\mathbf{A})$, 线性方程组 $(\lambda_0\mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{0}$ 的非零解子空间称为 \mathbf{A} 的属于特征根 λ_0 的特征子空间, 记做 $E_{\lambda_0}(\mathbf{A})$, 其中的非零向量就是特征向量。这些概念统称为矩阵 \mathbf{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

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

的特征值和特征向量。

矩阵 \mathbf{A} 的特征方程为

$$\det(\lambda\mathbf{I} - \mathbf{A}) = \begin{vmatrix} \lambda - 1 & -2 & -2 \\ -2 & \lambda - 4 & -2 \\ -2 & -2 & \lambda - 1 \end{vmatrix} = (\lambda + 1)^2(\lambda - 5) = 0,$$

故 \mathbf{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 $(\lambda\mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{0}$, 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 $\mathbf{x} = [-1, 1, 0]^T$, 记为 \mathbf{x}_1 ,
- 设 $x_2 = 0, x_3 = 1$, 求出解为 $\mathbf{x} = [-1, 0, 1]^T$, 记为 \mathbf{x}_2 ,
- 则 \mathbf{x}_1 和 \mathbf{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 $k_1\mathbf{x}_1 + k_2\mathbf{x}_2, k_1, k_2 \in \mathbb{R}$ 。

同理, $\lambda_3 = 5$ 的一个特征向量为 $\mathbf{x}_3 = [1, 1, 1]^T$, 属于 5 的全部特征向量为 $k\mathbf{x}_3, k \in \mathbb{R}$ 。

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 2.4.10. 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 2.4.11. 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 \mathbf{y} 为 B 的特征向量, 则相应地, $X\mathbf{y}$ 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 2.4.17. 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 2.4.12. 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, 它们是全空间 \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. 求矩阵 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 2.4.18. 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, 那么它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。

矩阵的对角化和特征分解

下面我们将关注如何将一个矩阵变成对角的形式, 这是第 4 讲基变换和特征值的一个应用。对角矩阵具有如下形式

$$D = \begin{pmatrix} c_1 & & \\ & \ddots & \\ & & c_n \end{pmatrix}$$

对角矩阵的行列式、幂和逆有以下关系:

- (1) 行列式是他对角元素的乘积。
- (2) 幂矩阵 \mathbf{D}^k 就是将每一个对角元素变成 k 的幂次。
- (3) 逆 \mathbf{D}^{-1} 就是将对角元素变成倒数。

若两个矩阵 \mathbf{D}, \mathbf{A} 相似, 则存在一个可逆矩阵使得 $\mathbf{D} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$ 成立。更具体地, 可以让 \mathbf{A} 相似于一个对角矩阵 \mathbf{D} , 而且 \mathbf{D} 的对角线上包含了矩阵 \mathbf{A} 的特征值, 则对角化定义为:

定义 2.4.19. 一个矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 是可对角化的, 如果它相似于一个对角矩阵, 即存在一个可逆矩阵 $\mathbf{P} \in \mathbb{R}^{n \times n}$ 使得 $\mathbf{D} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$ 成立。

下面将一个矩阵 \mathbf{A} 对角化是一种在不同基底下表达相同线性映射的方式。即找到由 \mathbf{A} 的特征向量构成的一个新基底把矩阵 \mathbf{A} 对角化。首先考虑如何计算 \mathbf{P} 来对角化矩阵 \mathbf{A} 。

例 2.4.8. 令 $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\lambda_1, \dots, \lambda_n$ 是标量, $\mathbf{p}_1, \dots, \mathbf{p}_n$ 是 \mathbb{R}^n 中的一组向量; $\mathbf{P} = (\mathbf{p}_1, \dots, \mathbf{p}_n)$, $\mathbf{D} \in \mathbb{R}^{n \times n}$ 是对角矩阵, 并且其对角元为 $\lambda_1, \dots, \lambda_n$; 证明

$$\mathbf{AP} = \mathbf{PD}$$

当且仅当 $\lambda_1, \dots, \lambda_n$ 是 \mathbf{A} 的特征值而 \mathbf{p}_i 是对应的特征向量。

解. (1) 根据上面的假设, 有下面等式成立

$$\mathbf{AP} = \mathbf{A}(\mathbf{p}_1, \dots, \mathbf{p}_n) = (\mathbf{Ap}_1, \dots, \mathbf{Ap}_n)$$

以及

$$\mathbf{PD} = (\mathbf{p}_1, \dots, \mathbf{p}_n) \begin{pmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \lambda_n & \end{pmatrix} = (\lambda_1 \mathbf{p}_1, \dots, \lambda_n \mathbf{p}_n)$$

(2) 因为要使 $\mathbf{AP} = \mathbf{PD}$ 成立, 这就意味着有

$$\mathbf{Ap}_1 = \lambda_1 \mathbf{p}_1$$

$$\vdots \quad ,$$

$$\mathbf{Ap}_n = \lambda_n \mathbf{p}_n$$

反之亦成立。

(3) 由此可知, 矩阵 \mathbf{P} 必须由特征列构成。但是这并不足以让我们知道我们是否可以对角化 \mathbf{A} , 因为在我们的定义之中 \mathbf{P} 是可逆的。我们知道对于方阵, 当且仅当 \mathbf{P} 是满秩的, 则它是可逆的。这意味着特征向量 $\mathbf{p}_1, \dots, \mathbf{p}_n$ 线性无关的时候, \mathbf{P} 才是可逆的。

(4) 根据结论: 方阵 \mathbf{A} 若有 n 个不同的特征值及其对应的 n 个特征向量, 则这 n 个特征向量线性无关。

定理 2.4.13. 一个矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 可以分解为

$$\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1},$$

其中 \mathbf{P} 是由特征向量构成的可逆矩阵, \mathbf{D} 是对角矩阵且对角元是 \mathbf{A} 的特征值, 当且仅当 \mathbf{A} 有 n 个线性无关的特征向量。

特征分解的几何意义

如图2.20所示, 令 A 为标准基下的线性映射的变换矩阵, P^{-1} 将标准基变换到特征基下, 特征向量 p_i (图2.20红色和绿色箭头) 映射到坐标轴轴 e_i 上, 然后对角矩阵 D 通过特征值沿这些轴缩放特征向量, 即 $\lambda_i e_i$, 最后, P 将这些缩放后的矢量变换回标准基下。

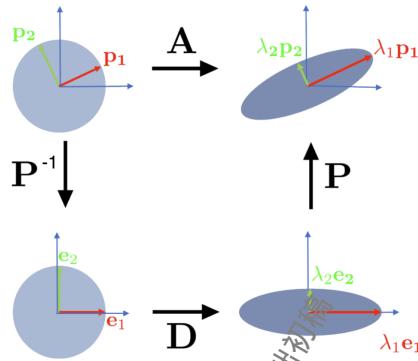


图 2.20: 特征分解示意图

求解方阵 $A \in \mathbb{R}^{n \times n}$ 的特征分解步骤如下:

(1) 计算矩阵 A 的特征值 $\lambda_1, \dots, \lambda_n$ 。即求特征方程

$$|A - \lambda I| = 0$$

的 n 个根, 并根据这些根是否都是不同的来判断矩阵 A 是否可对角化。

(2) 求特征值对应的 n 个线性无关的特征向量 p_1, \dots, p_n 。即求解方程组

$$Ap_i = \lambda_i p_i, i = 1, \dots, n$$

若不能找到 n 个线性无关的特征向量 p_1, \dots, p_n , 则说明矩阵 A 不能进行特征分解。

(3) 记矩阵 $P = (p_1, \dots, p_n)$ 并计算 P^{-1} 。

(4) 最终得到矩阵 A 的特征分解为

$$A = P \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} P^{-1}$$

例 2.4.9. 计算 $A = \begin{pmatrix} 2 & 0 \\ 4 & 4 \end{pmatrix}$ 的特征分解

解. 首先计算特征值

$$|A - \lambda I| = (\lambda - 4)(\lambda - 2) = 0$$

所以特征值为 $\lambda_1 = 4, \lambda_2 = 2$ 对应的特征向量通过

$$A\mathbf{p}_1 = 4\mathbf{p}_1, A\mathbf{p}_2 = 2\mathbf{p}_2,$$

得到, 所以有

$$\mathbf{p}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \mathbf{p}_2 = \begin{pmatrix} 1 \\ -2 \end{pmatrix}$$

因为 A 有 2 个不同的特征值, 所以一定可以进行对角化。

然后将特征向量合并起来得到 \mathbf{P} , 并计算 \mathbf{P}^{-1} , 有

$$\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2) = \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix}, \quad \mathbf{P}^{-1} = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}$$

$$A\mathbf{P} = \begin{pmatrix} 2 & 0 \\ 4 & 4 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix} = \begin{pmatrix} 0 & 2 \\ 4 & -4 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix} = \mathbf{P}\mathbf{D}$$

因此

$$A = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$$

即

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

矩阵的特征值分解有以下性质:

(1) 我们通过对角矩阵良好的性质来计算矩阵的幂 A^k , 即

$$A^k = \mathbf{P}\mathbf{D}^k\mathbf{P}^{-1}$$

(2) 对角矩阵的另一个特性是, 它们可以用于解耦变量。这在概率论中解释随机变量非常重要, 例如我们在降维中碰到的高斯分布等。

(3) 特征分解需要具有 n 个不同特征值的方阵才能做, 对于不具有 n 个不同特征值的方阵则不存在特征分解。

定义 2.4.20. 设矩阵 $A \in \mathbb{R}^{n \times n}$, 如果他没有 n 个线性无关的特征向量或特征子空间的维数之和小于 n , 则我们则称其为亏损矩阵。

注记 10. 亏损矩阵一定少于 n 个不同的特征值, 因为不同特征值对应的特征向量是线性无关的。具体而言, 亏损矩阵至少有一个代数重数为 $m > 1$ 的特征值 λ , 其有少于 m 个对应的线性无关的特征向量。亏损矩阵是不可对角化的, 也即不存在特征分解, 但是可以通过若当标准型 (Jordan Normal Form) 来对其做若当分解。

注记 11. 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 10 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，系统地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。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 $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \mathbb{R}^3$ ，求方程式 $Ax = 12x$ 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 $A = \begin{pmatrix} 3 & 1 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 1 & -1 \\ 2 & -1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$

计算 $AB, AB - BA$

习题 2.5. 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 2.6. 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 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 $\ker(f) \cap \text{Im}(g) = \{0_E\}$ 。

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:

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

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:

$$(1) \quad x_1^2 + x_2^2 + 5x_3^2 + 2tx_1x_2 - 2x_1x_3 + 4x_2x_3$$

$$(2) \quad x_1^2 + 4x_2^2 + x_3^2 + 2tx_1x_2 + 10x_1x_3 + 6x_2x_3$$

习题 2.21. 设 $A = \begin{pmatrix} 1 & 4 & 2 \\ 0 & -3 & 4 \\ 0 & 4 & 3 \end{pmatrix}$ 求 A^k

习题 2.22. 证明: 如果 A 可逆, 证明: AB 与 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 A , 对每一个 $\mathbf{x} \in R^n$ 都使 $f(\mathbf{x}) = A\mathbf{x}$ 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 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 2.25. 证明: 在 \mathbb{R}^n 上, 当且仅当对称矩阵 A 是正定矩阵时, 函数 $f(\mathbf{x}) = (\mathbf{x}^T \mathbf{A} \mathbf{x})^{\frac{1}{2}}$ 是一个向量范数。

习题 2.26. 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 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，比如如何获得两段文本或两幅图像的类别信息。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标。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的几何结构：度量和投影，用来刻画向量空间的几何特征，包括向量的长度，两个向量之间的距离、角度等度量特征，以及高维空间到低维空间的投影特征等。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。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：

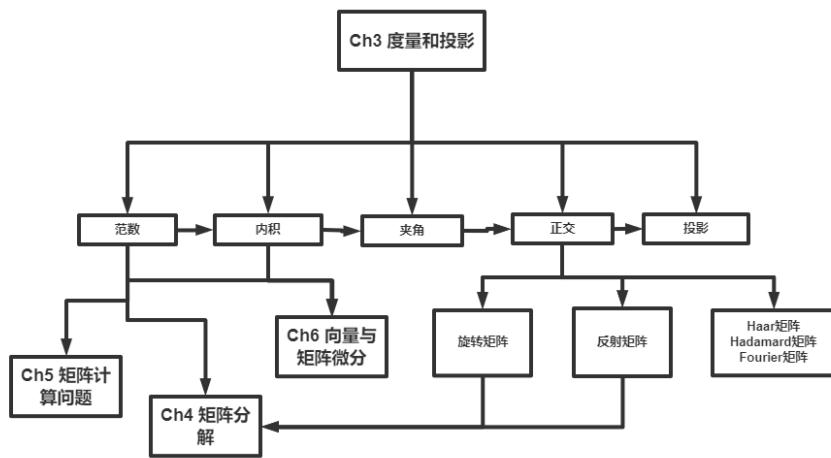


图 3.1: 本章导图

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)引入一种度量作为它们的“大小”, 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?”, 可以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。

$$sim_{cos}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}| |\mathbf{y}|},$$

其中 \mathbf{x}, \mathbf{y} 是文本向量, $sim_{cos}(\mathbf{x}, \mathbf{y})$ 表示 \mathbf{x}, \mathbf{y} 的余弦相似度。

如图3.2所示, 在MINST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。当我们对测试样本进行预测时, 根据最近邻算法, 只需要去找到和这个数据最相似的训练数据所属的类别。另外, 我们也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



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)

间的差异, 从而可以使用下面的公式计算图片间的差异:

$$d(\mathbf{A}, \mathbf{T}) = \sum_{jk} |A_{jk} - T_{jk}|$$

其中 d 是手写数字训练图片的表示矩阵 \mathbf{A} 和测试图片的表示矩阵 \mathbf{T} 之间的距离 (两个矩阵同等大小), A_{jk}, T_{jk} 分别表示矩阵 A 和 T 的第 j 行第 k 列的元素, j, k 取遍矩阵所有元素。距离越大, 则图片越不相似; 距离越小, 图片越相似。

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。这就需要引入向量与向量之间的相似度量方法, 下面先介绍向量的范数, 长度与距离, 用以学习向量与向量之间的相似度量方法。

3.1.1 向量范数

向量范数的定义

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 $\mathbf{x} = (a, b) = a + ib$ 的长度或者模指的是

$$\|\mathbf{x}\| = \sqrt{a^2 + b^2}$$

显然复数 \mathbf{x} 的模 $\|\mathbf{x}\|$ 具有下列三条性质:

- (1) 非负性: $\|\mathbf{x}\| \geq 0$, 当且仅当 $\mathbf{x} = 0$ 时等号成立;
- (2) 齐次性: $\|\lambda\mathbf{x}\| = |\lambda|\|\mathbf{x}\|$, ($\forall \lambda \in \mathbb{R}$);
- (3) 三角不等式: $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$, ($\mathbf{x}, \mathbf{y} \in \mathbb{C}$).

例 3.1.2. n 维向量 $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ 的模或长度定义为

$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

显然向量 \mathbf{x} 的模 $\|\mathbf{x}\|$ 也具有下列三条性质:

- (1) 非负性: $\|\mathbf{x}\| \geq 0$, 当且仅当 $\mathbf{x} = 0$ 时等号成立;
- (2) 齐次性: $\|\lambda\mathbf{x}\| = |\lambda|\|\mathbf{x}\|$, ($\forall \lambda \in \mathbb{R}$);
- (3) 三角不等式: $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$, ($\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$).

向量的模又称为欧氏长度，代表了从原点 0 到点 \mathbf{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。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，其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 $\mathbf{x} = (x_1, x_2, x_3)^T \in \mathbb{C}^3$ ，试问如右所示实值函数是否构成向量范数？

- (1) $|x_1| + |2x_2 + x_3|$,
- (2) $|x_1| + |2x_2| - 5|x_3|$,
- (3) $|x_1| + 3|x_2| + 2|x_3|$.

解. 我们只需要验证实值函数是否满足三条性质。

- (1) 取 $\mathbf{x} = (0, 1, -1)$ ， $|0| + |1 - 1| = 0$ 即存在非零 \mathbf{x} 使得此实值函数为 0，不满足非负性。
所以 $|x_1| + |2x_2 + x_3|$ 不是一个向量范数。
- (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) 非负性： $|x_1| + 3|x_2| + 2|x_3| \geq 0$
齐次性：令 $c \in \mathbb{C}$, $|cx_1| + 3|cx_2| + 2|cx_3| = |c||x_1| + 3|c||x_2| + 2|c||x_3|$
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, $|x_1+y_1|+3|x_2+y_2|+2|x_3+y_3| \leq |x_1| + 3|x_2| + 2|x_3| + |y_1| + 3|y_2| + 2|y_3|$
所以， $|x_1| + 3|x_2| + 2|x_3|$ 是向量范数。

常用范数

这里以 \mathbb{R}^n 空间为例， \mathbb{C}^n 空间类似，最常用的范数就是 p 范数。

例 3.1.4. 对于任意 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ ，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

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$$

其中 l_2 范数就是通常意义上的距离。

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 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$, 由

$$\|\mathbf{x}\|_\infty = \lim_{p \rightarrow \infty} \|\mathbf{x}\|_p = \max_{i=1, \dots, n} |x_i|,$$

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. 易证 $\|\mathbf{x}\|_\infty \equiv \max_i |x_i|$ 。下证 $\max_i |x_i| = \lim_{p \rightarrow +\infty} \|\mathbf{x}\|_p$ 。令 $\|\mathbf{x}_j\| = \max_i |x_i|$, 则有

$$\begin{aligned} \|\mathbf{x}\|_\infty &= |\mathbf{x}_j| \leq \left(\sum_{i=1}^n |\mathbf{x}_i|^p \right)^{\frac{1}{p}} = \|\mathbf{x}\|_p \\ &\quad (n|\mathbf{x}_j|^p)^{\frac{1}{p}} = n^{\frac{1}{p}} \|\mathbf{x}\|_\infty \end{aligned}$$

由极限的夹逼准则, 并注意到 $\lim_{p \rightarrow +\infty} n^{\frac{1}{p}} = 1$, 证毕。 □

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

$$\begin{aligned} \|\boldsymbol{\alpha}\|_{\frac{1}{2}} &= \|\boldsymbol{\beta}\|_{\frac{1}{2}} = 1, \|\boldsymbol{\alpha} + \boldsymbol{\beta}\|_{\frac{1}{2}} = 4 \\ \|\boldsymbol{\alpha} + \boldsymbol{\beta}\|_{\frac{1}{2}} &\geq \|\boldsymbol{\alpha}\|_{\frac{1}{2}} + \|\boldsymbol{\beta}\|_{\frac{1}{2}} \end{aligned}$$

不满足三角不等式。 □

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

$$\text{card}(\mathbf{x}) = \sum_{i=1}^n \mathcal{I}(x_i \neq 0)$$

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.

$$\|\mathbf{x}\|_0 = 3$$

$$\|\mathbf{x}\|_1 = |-1| + 2 + 4 = 7$$

$$\|\mathbf{x}\|_2 = \sqrt{|-1|^2 + 2^2 + 4^2} = \sqrt{21}$$

$$\|\mathbf{x}\|_\infty = \max\{|-1|, 2, 4\} = 4$$

例 3.1.8. 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

$$\|\mathbf{x}\|_\infty \|\mathbf{x}\|_1 \leq n \|\mathbf{x}\|_\infty$$

$$\frac{1}{\sqrt{n}} \|\mathbf{x}\|_1 \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1$$

$$\frac{1}{\sqrt{n}} \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2$$

或者

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{n} \|\mathbf{x}\|_2 \leq n \|\mathbf{x}\|_\infty$$

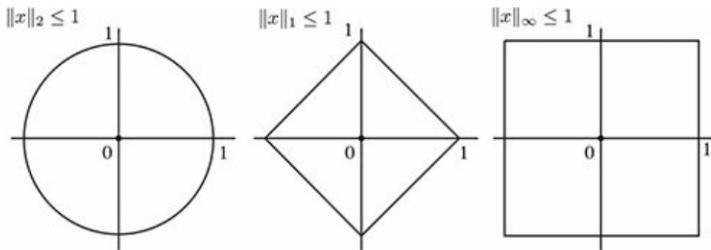
范数的几何意义

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 3.1.9. 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

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} \rightarrow \mathbf{x}^*$$

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

$$c = \sum_{i=1}^n \|\mathbf{e}_i\|.$$

□

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 $S = \{\mathbf{x} \mid \|\mathbf{x}\|_\infty = 1, \mathbf{x} \in \mathbb{R}^n\}$, 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

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}^* \iff \lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\| = 0$$

其中 $\|\cdot\|$ 为向量的任一种范数. 若 $\lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\| = 0$, 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

$$\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}^* \iff \left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\|_\infty \rightarrow 0 \quad (\text{当 } k \rightarrow \infty \text{ 时}),$$

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

$$\left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\|_\infty \rightarrow 0 \iff \left\| \mathbf{x}^{(k)} - \mathbf{x}^* \right\|_t \rightarrow 0 \quad (\text{当 } k \rightarrow \infty \text{ 时})$$

□

这就说明向量列依坐标收敛等价于依范数收敛.

定义 3.1.5. (柯西序列) 一向量序列 $\{\mathbf{x}^{(k)}\}$ 被称为柯西序列, 如果对于任何正实数 $r > 0$, 存在一个正整数 N 使得对于所有的整数 $m, n \geq N$, 都有

$$\|\mathbf{x}_m - \mathbf{x}_n\| \leq r$$

定义 3.1.6. (完备性) 如果任何柯西序列都收敛, 则称一个度量空间是完备的.

3.1.2 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.

内积的定义

定义 3.1.7. 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

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y} = \sum_{i=1}^n x_i y_i$$

通常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 3.1.8. 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 $\langle \mathbf{x}, \mathbf{y} \rangle$ ，若满足

- (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 $\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$ 。

则 $\langle \mathbf{x}, \mathbf{y} \rangle$ 是向量 \mathbf{x}, \mathbf{y} 的内积，且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。

例 3.1.11. 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。

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示，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。

- 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。

- 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 3.1.12. 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

$$\mathbf{x}^T \mathbf{A}_1 \mathbf{x} = \begin{pmatrix} x_1, x_2 \end{pmatrix} \begin{pmatrix} 9 & 6 \\ 6 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 9x_1^2 + 12x_1x_2 + 5x_2^2 = (3x_1 + 2x_2)^2 + x_2^2 > 0$$

\mathbf{A}_2 是对称的但不正定。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 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 是对称、正定的，则

$$\langle \mathbf{x}, \mathbf{y} \rangle = \hat{\mathbf{x}}^T \mathbf{A} \hat{\mathbf{y}}$$

定义了一个关于有序基底 \mathbf{B} 的内积，其中 $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ 是 \mathbb{V} 中向量 \mathbf{x}, \mathbf{y} 关于 \mathbf{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 \mathbf{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 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 满足

$$\langle \mathbf{x}, \mathbf{y} \rangle = \hat{\mathbf{x}}^T \mathbf{A} \hat{\mathbf{y}}$$

则

- \mathbf{A} 的核（零空间）只包含 $\mathbf{0}$ 因为 $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$ 对于任意 $\mathbf{x} \neq 0$ 成立，即如果 $\mathbf{x} \neq 0$ 则 $\mathbf{A} \mathbf{x} \neq 0$ ；
- \mathbf{A} 的对角元是正的，因为 $a_{ii} = \mathbf{e}_i^T \mathbf{A} \mathbf{e}_i > 0$ ，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。如果我们将 $\|\mathbf{x}\|$ 定义为 $\sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ ，容易验证 $\sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ 满足范数定义要求的非负性、齐次性和三角不等式。

从这个角度看，一个内积空间包含有一个赋范线性空间。

定义 3.1.9. 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。

标准内积与 l_2 范数之间存在联系：

$$\|\mathbf{x}\|_2^2 = \mathbf{x}^T \mathbf{x}$$

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。同样，范数不一定可以推出内积，当范数满足平行四边形公式 $\|\mathbf{x} + \mathbf{y}\|^2 + \|\mathbf{x} - \mathbf{y}\|^2 = 2(\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2)$ 时，这个范数一定可以诱导内积；完备的内积空间称为希尔伯特空间。

柯西施瓦兹不等式

定理 3.1.5. 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 $\lambda \in \mathbb{R}$ ，

$$\begin{aligned} 0 &\leq \langle \mathbf{x} - \lambda \mathbf{y}, \mathbf{x} - \lambda \mathbf{y} \rangle \\ &= \langle \mathbf{x} - \lambda \mathbf{y}, \mathbf{x} \rangle - \lambda \langle \mathbf{x} - \lambda \mathbf{y}, \mathbf{y} \rangle \\ &= \langle \mathbf{x}, \mathbf{x} \rangle - 2\lambda \langle \mathbf{x}, \mathbf{y} \rangle + \lambda^2 \langle \mathbf{y}, \mathbf{y} \rangle \\ &= \|\mathbf{x}\|^2 - 2\lambda \langle \mathbf{x}, \mathbf{y} \rangle + \lambda^2 \|\mathbf{y}\|^2 \end{aligned}$$

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

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\| \|\mathbf{y}\|.$$

□

距离的度量空间

利用范数或者内积，我们可以定义两个向量间的距离。

定义 3.1.10. 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。若使用内积导出的范数，则距离会依赖于内积的选择。下面给出度量的数学定义。

定义 3.1.11. 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 3.1.12. 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 \mathbb{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。

定义 3.1.13. (Banach 空间) 如果赋范线性空间作为(其范数自然诱导度量 $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$) 的原点空间是完备的，即柯西序列收敛，则称这个赋范线性空间为巴拿赫 (Banach) 空间。

向量之间的夹角

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 yxz 运用勾股定理，有

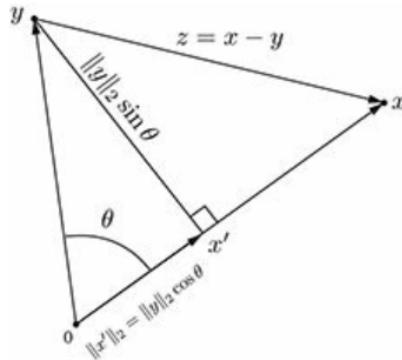
$$\begin{aligned} \|\mathbf{z}\|_2^2 &= (\|\mathbf{y}\|_2 \sin \theta)^2 + (\|\mathbf{x}\|_2 - \|\mathbf{y}\|_2 \cos \theta)^2 \\ &= \|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 - 2 \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \cos \theta. \end{aligned}$$

由于

$$\|\mathbf{z}\|_2^2 = \|\mathbf{x} - \mathbf{y}\|_2^2 = (\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y}) = \mathbf{x}^T \mathbf{x} + \mathbf{y}^T \mathbf{y} - 2 \mathbf{x}^T \mathbf{y},$$

则有

$$\mathbf{x}^T \mathbf{y} = \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \cos \theta.$$

图 3.4: 向量 x, y 之间的夹角 θ

则向量 x, y 之间的夹角为

$$\cos \theta = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2} \quad (3.1)$$

当 $\mathbf{x}^T \mathbf{y} = 0$ 时, 向量 x, y 之间的角度为 90° , 称为正交。当 θ 为 0° 或者 180° 时, x, y 成一直线, 即 $\mathbf{y} = k\mathbf{x}, k \in \mathbb{K}$, 称为平行。

向量的正交

定义 3.1.14. 设向量 $\mathbf{x}, \mathbf{y} \in \mathbb{X}$, 如果 $\langle \mathbf{x}, \mathbf{y} \rangle = 0$, 则称 \mathbf{x}, \mathbf{y} 正交, 记作 $\mathbf{x} \perp \mathbf{y}$ 。特别地, 如果 $\|\mathbf{x}\| = 1 = \|\mathbf{y}\|$, x 和 y 即是单位向量时, 称 \mathbf{x}, \mathbf{y} 标准正交。

零向量与任何向量正交。

对于非零向量组 $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_d\}$, 如果对于 $\forall i \neq j$, 有 $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = 0$, 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 $\mathbf{x} = (1, 1)^T, \mathbf{y} = (-1, 1)^T \in \mathbb{R}^2$ 。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 $\mathbf{x} \perp \mathbf{y}$ 。

而我们选择内积

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{y}$$

计算 \mathbf{x}, \mathbf{y} 之间的角度 ω 时,

$$\cos \omega = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|} = -\frac{1}{3} \implies \omega \approx 109.5^\circ$$

所以 \mathbf{x}, \mathbf{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 3.1.15. 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。事实上，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

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

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。

3.1.3 数据科学中常用的相似性度量

聚类和分类是数据分析的重要运算。聚类是把大数据集聚为 N 类子集，并且每个子集（目标类）的数据都具有共同或者相似的特征。分类则是将一个数据映射到某个已知目标类别中。

相似性度量是聚类与分类算法中一个很重要的数学工具。本小节主要讨论非概率相关的相似性度量。

距离作为相似性度量

在一个向量空间中，两点之间是否相似最直观的就是距离相近的相似。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 n 个样本，每个样本由 m 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 3.1.16. 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 $\mathbf{x}_i, \mathbf{x}_j \in X$, $\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{mi})^T$, $\mathbf{x}_j = (x_{1j}, x_{2j}, \dots, x_{mj})^T$, 样本 \mathbf{x}_i 与样本 \mathbf{x}_j 的闵可夫斯基距离, 简称闵氏距离, 定义为

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt[p]{\sum_{k=1}^m |x_{ki} - x_{kj}|^p} = \|\mathbf{x}_i - \mathbf{x}_j\|_p,$$

其中 $1 \leq p < \infty$ 。

- (1) 当 $p = 2$ 时, 对应欧氏距离, 是多维空间中各个点之间的直线距离。
- (2) 当 $p = 1$ 时, 对应曼哈顿距离, 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。
- (3) 当 $p \rightarrow \infty$ 时, 对应切比雪夫距离, 是将两个点其各坐标数值差绝对值的最大值作为距离。

定义 3.1.17. 欧氏距离计算公式如下:

$$dist(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} = \|\mathbf{x} - \mathbf{y}\|_2$$

定义 3.1.18. 曼哈顿距离计算公式如下:

$$dist(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n |x_i - y_i| = \|\mathbf{x} - \mathbf{y}\|_1$$

曼哈顿距离是在 1 范数意义下的距离。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。

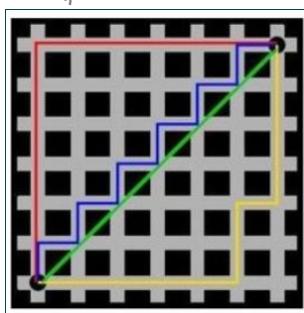


图 3.5: 曼哈顿距离

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 3.1.19. 切比雪夫距离计算公式如下:

$$dist(\mathbf{x}, \mathbf{y}) = \max |x_i - y_i| = \|\mathbf{x} - \mathbf{y}\|_\infty$$

切比雪夫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。如图3.6所示。

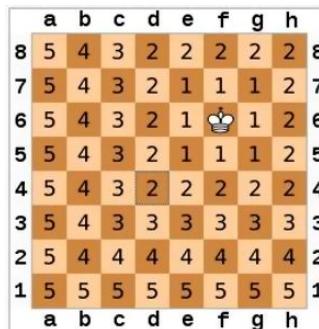


图 3.6: 国际象棋中的切比雪夫距离

下面以闵氏距离为例, 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时, 只需比较距离该数据最近的 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3.7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。

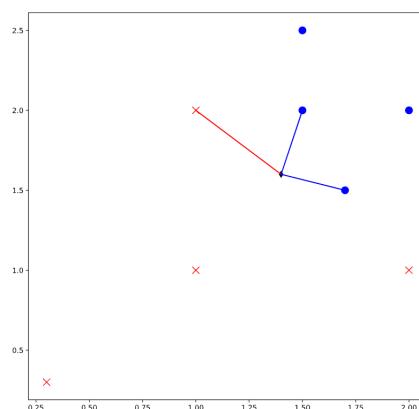


图 3.7: k -NN, 待预测样本为正类。

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。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。

若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。若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。

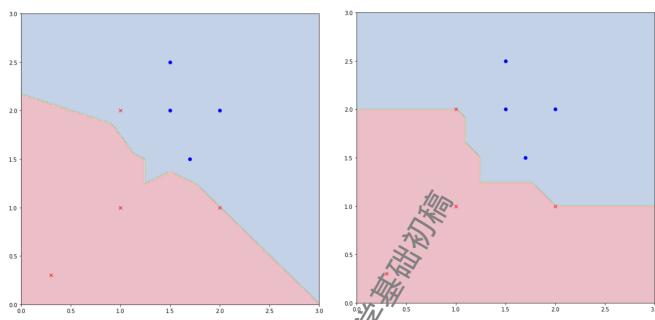


图 3.8: 左图采用欧氏距离，右图采用曼哈顿距离

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。

例如考虑：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存在缺点。

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 3.1.20. 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

$$sim_{cos}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i||\mathbf{x}_j|} = \frac{\sum_{k=1}^m x_{ki}x_{kj}}{[\sum_{k=1}^m x_{ki}^2 \sum_{k=1}^m x_{kj}^2]^{\frac{1}{2}}}$$

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，需要调整余弦相似度来修正这种不合理性，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，那么调整后评分分别为 $(-2, -1)$ 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。

其它相似性度量

定义 3.1.21. 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. 计算如下字符串的汉明距离

- (1) 1011101 与 1001001 之间的汉明距离是 2。
- (2) 2143896 与 2233796 之间的汉明距离是 3。
- (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，也可以将其拓展应用到向量上。

3.1.4 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。

矩阵范数

定义 3.1.22. 令 $m \times n$ 实矩阵 $A = [a_1, \dots, a_n]$ ，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 3.1.23. 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 3.1.24. 对于任意的 $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}$ 。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) $\|\mathbf{A}\|_{m_1} \geq 0$, ($\|\mathbf{A}\|_{m_1} = 0 \iff \mathbf{A} = \mathbf{0}$);
- (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

$$\|\mathbf{A}\|_F := \left(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2 \right)^{\frac{1}{2}} = (\text{Tr}(\mathbf{A}^T \mathbf{A}))^{\frac{1}{2}}$$

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。

例 3.1.20. 在数据科学中, 有时还用到 p, q -矩阵范数。

$$\begin{aligned} \|\mathbf{A}\|_{1,2} &= \left(\sum_{j=1}^n \|\mathbf{a}_j\|_1^2 \right)^{\frac{1}{2}} = \left(\sum_{j=1}^n \left(\sum_{i=1}^m |A_{ij}| \right)^2 \right)^{\frac{1}{2}} \\ \|\mathbf{A}\|_{2,1} &= \sum_{j=1}^n \|\mathbf{a}_j\|_2 = \left(\sum_{j=1}^n \left(\sum_{i=1}^m |A_{ij}|^2 \right) \right)^{\frac{1}{2}} \\ \|\mathbf{A}\|_{p,q} &= \left(\sum_{j=1}^n \left(\sum_{i=1}^m |A_{ij}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}} \\ \|\mathbf{A}\|_* &= \text{Tr}(\sqrt{\mathbf{A}^T \mathbf{A}}) \end{aligned}$$

范数的相容性

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 3.1.25. 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

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

那么

$$\|A^2\|_{m_\infty} = \|2A\|_{m_\infty} = 2 \not\leq \|A\|_{m_\infty}^2 = 1$$

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:

$$\|A\|_{m_\infty} := n \max_{1 \leq i \leq m, 1 \leq j \leq n} |a_{ij}|$$

□

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 3.1.26. 若矩阵范数 $\|\cdot\|_M$ 和向量范数 $\|\cdot\|_v$ 满足

$$\|Ax\|_v \leq \|A\|_M \|x\|_v, \quad A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n,$$

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, 我们都可以构造一个与该向量范数相容的矩阵范数。

定义 3.1.27. $m \times n$ 矩阵空间上如下定义的范数 $\|\cdot\|$ 称为从属于向量范数 $\|\cdot\|_v$ 的矩阵范数, 也称其为由向量范数 $\|\cdot\|_v$ 诱导出的算子范数

$$\begin{aligned} \|A\| &= \max\{\|Ax\|_v : x \in \mathbb{R}^n, \|x\|_v = 1\} \\ &= \max\left\{\frac{\|Ax\|_v}{\|x\|_v} : x \in \mathbb{R}^n, x \neq 0\right\} \end{aligned}$$

显然, 该矩阵范数和向量范数 $\|\cdot\|_v$ 是相容的。

因为, 对任意 $x \in \mathbb{R}^n, x \neq 0$,

$$\frac{\|Ax\|_v}{\|x\|_v} \leq \max\left\{\frac{\|Ax\|_v}{\|x\|_v} : x \in \mathbb{R}^n, x \neq 0\right\} = \|A\|$$

所以 $\|Ax\|_v \leq \|A\|_m \|x\|_v$ 。

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

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|$$

$$\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|$$

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$$

证明：当 $A = \mathbf{O}$ 时, 以上三式显然成立。假定 $A \neq \mathbf{O}$, 对以上的三个范数进行证明。

1 范数证明 对于 1 范数, 将给定的 $A \in \mathbb{R}^{m \times n}$ 按列分块 $A = [\mathbf{a}_1, \dots, \mathbf{a}_n]$, 并记 $\delta = \|\mathbf{a}_{j_0}\|_1 = \max_{1 \leq j \leq n} \|\mathbf{a}_j\|_1$, 则对任意满足 $\|x\|_1 = \sum_{i=1}^n |x_i| = 1$ 的 $x \in \mathbb{R}^n$, 有

$$\begin{aligned} \|Ax\|_1 &= \left\| \sum_{j=1}^n x_j \mathbf{a}_j \right\| \leq \sum_{j=1}^n |x_j| \|\mathbf{a}_j\|_1 \\ &\leq \sum_{j=1}^n |x_j| \max_{1 \leq j \leq n} \|\mathbf{a}_j\|_1 = \|\mathbf{a}_{j_0}\|_1 = \delta \end{aligned}$$

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

$$\|Ae_{j_0}\|_1 = \|\mathbf{a}_{j_0}\|_1 = \delta$$

这样我们证明了存在满足 $\|x\|_1 = 1$ 的 x , 使得 $\|Ax\|_1 = \delta$ 。

因此有

$$\|A\|_1 = \max_{\|x\|_1=1} \|Ax\|_1 = \delta = \max_{1 \leq j \leq n} \|\mathbf{a}_j\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|.$$

∞ 范数证明 对于 ∞ 范数, 记

$$\eta = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|,$$

则对任意满足 $\|x\|_\infty = 1$ 的 $x \in \mathbb{R}^n$, 有

$$\|Ax\|_\infty = \max_{1 \leq i \leq m} \left| \sum_{j=1}^n a_{ij} x_j \right| \leq \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}| |x_j| \leq \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}| = \eta$$

此处我们证明了 $\|A\|_\infty := \max_{\|x\|_\infty=1} \|Ax\|_\infty \leq \eta$ 。

设 A 的第 k 行元素的绝对值之和最大, 即 $\eta = \sum_{j=1}^n |a_{kj}|$ 。令

$$\tilde{x} = (sgn(a_{k1}), \dots, sgn(a_{kn}))^T$$

则 $A \neq \mathbf{0}$ 蕴含 $\|\tilde{x}\|_\infty = 1$, 有 $\|A\tilde{x}\|_\infty = \sum_{j=1}^n |a_{kj}| = \eta$ 。

这里证明了存在满足 $\|\mathbf{x}\|_\infty = 1$ 的 \mathbf{x} , 使得 $\|A\mathbf{x}\|_\infty = \eta$,

则

$$\|A\|_\infty = \eta = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}| \quad \square$$

2 范数证明 对于 2 范数, 应有

$$\begin{aligned} \|A\|_2 &= \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2 = \max_{\|\mathbf{x}\|_2=1} [(A\mathbf{x})^T A\mathbf{x}]^{\frac{1}{2}} \\ &= \max_{\|\mathbf{x}\|_2=1} [\mathbf{x}^T (A^T A) \mathbf{x}]^{\frac{1}{2}} \end{aligned}$$

注意, $A^T A$ 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

$$\mathbf{x}^T A^T A \mathbf{x} = \sum_{i=1}^n \lambda_i \alpha_i^2 \leq \lambda_1$$

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, $\|U A V\|_2 = \|A\|_2$ 。

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 $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^2 \mid \|\mathbf{x}\|_p = 1\}$ 在矩阵 $A = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix}$ 作用下的效果分别为

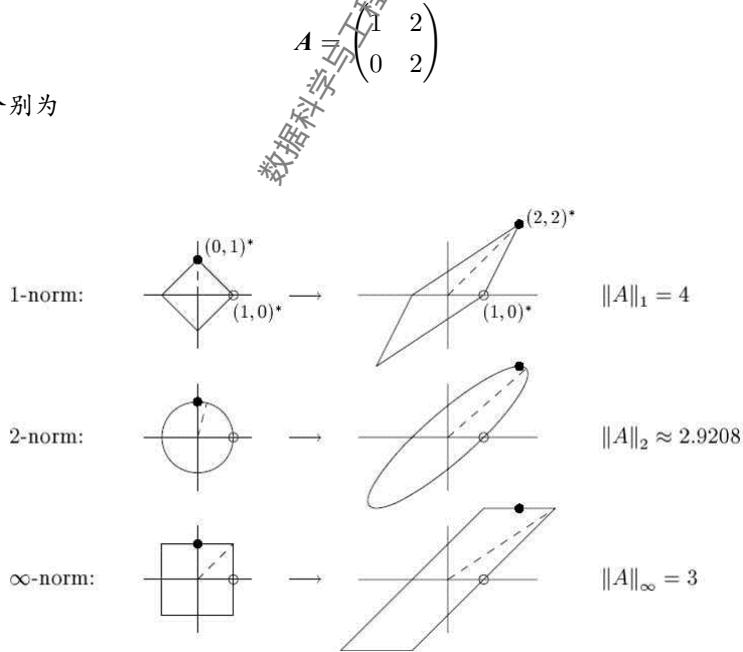


图 3.9: 不同向量范数下, 单位球面在矩阵作用下的变换。

3.1.5 范数在机器学习中的应用

在 1.3 节介绍了对于监督学习问题，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； $J(f)$ 是正则化项，正则化项主要用于防止模型过拟合， λ 是用于调节经验风险和正则化项关系的超参数。

在监督学习中损失函数 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和非距离度量形式的损失。

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，它们是由带有噪音的四次曲线产生的。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。

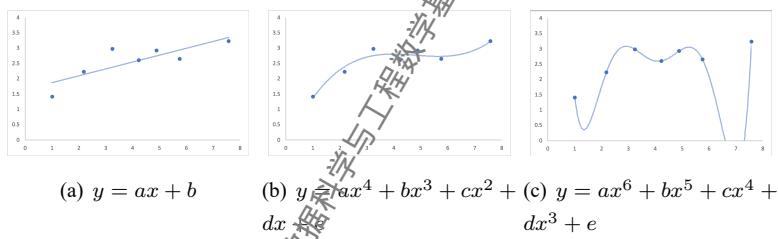


图 3.10: 欠拟合、正常拟合、过拟合

- 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。

正则化：范数的选择 在 3.1.26 中若想要避免过拟合，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音的数据的情况，高次函数拟合效果虽好但也拟合了数据噪音。

正则化在多种模型中的应用都非常广泛，如果数据集的特征数量大于样本的总数时，问题常常会有多个解，这时需要借助正则项来选出性质不同的解。这里我们记 $\mathbf{r} = (\mathbf{w}^T, b)^T$

如果想要平衡模型的拟合性质和解的光滑性,我们可以使用 Tikhonov 正则化(也叫岭回归),把 l_2 范数的平方作为正则项。那么我们问题的目标函数为:

$$\min \sum_{i=1}^N L(y_i, f(x_i)) + \lambda \|\mathbf{r}\|_2^2$$

而如果希望得到的解 \mathbf{x} 是稀疏的,那么可以考虑添加 l_1 范数为正则项,对应的正则化问题叫做 LASSO 问题:

$$\min \sum_{i=1}^N L(y_i, f(x_i)) + \lambda \|\mathbf{r}\|_1$$

对于某些实际问题,权重 \mathbf{r} 本身不是稀疏的,但其在某种变换下是稀疏,因此我们也需要调整对应的正则项:

$$\min \sum_{i=1}^N L(y_i, f(x_i)) + \lambda \|\mathbf{Fr}\|_1$$

如当 \mathbf{F} 取

$$\mathbf{F} = \begin{pmatrix} 1 & -1 \\ 1 & -1 \\ \vdots & \vdots \\ 1 & -1 \end{pmatrix}$$

时,它实际上要求 \mathbf{r} 相邻点之间的变化是稀疏的。实际上不同的正则化方法可以结合起来,同时提出多种要求。例如融合 LASSO 模型(fused-LASSO)可表示为

$$\min \sum_{i=1}^N L(y_i, f(x_i)) + \lambda_1 \|\mathbf{r}\|_1 + \lambda_2 \|\mathbf{Fr}\|_1$$

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。

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: $\text{Null}(\mathbf{A}^T)$

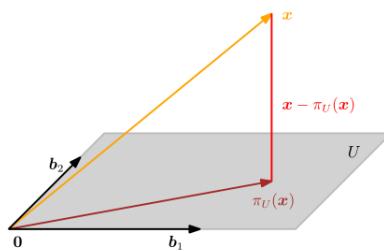


图 3.11: 将三维空间中的向量投影到二维平面上

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 $A \in \mathbb{R}^{m \times n}$, 其 m 个行向量、 n 个列向量分别记作

$$\begin{aligned} \mathbf{r}_1 &= [a_{11}, a_{12}, \dots, a_{1n}]^T & \mathbf{a}_1 &= \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix}, \mathbf{a}_2 = \begin{pmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{pmatrix}, \dots, \mathbf{a}_n = \begin{pmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{pmatrix} \\ \mathbf{r}_2 &= [a_{21}, a_{22}, \dots, a_{2n}]^T \\ \dots \\ \mathbf{r}_m &= [a_{m1}, a_{m2}, \dots, a_{mn}]^T \end{aligned}$$

即 $A = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m)^T = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)$

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 $Col(A)$ 表示, 即有

$$Col(A) = \left\{ \mathbf{y} \in \mathbb{R}^m \mid \mathbf{y} = \sum_{j=1}^n \alpha_j \mathbf{a}_j, \alpha_j \in \mathbb{R} \right\} = \text{span}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\} \quad (3.4)$$

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 $Row(A)$ 表示, 也可以用 $Col(A^T)$ 表示, 有

$$Row(A) = Col(A^T) = \left\{ \mathbf{y} \in \mathbb{R}^n \mid \mathbf{y} = \sum_{i=1}^m \beta_i \mathbf{r}_i, \beta_i \in \mathbb{R} \right\} = \text{span}\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m\} \quad (3.5)$$

定义 3.2.3. 零空间是所有满足齐次线性方程组 $A\mathbf{x} = \mathbf{0}$ 的解向量集合, 它是 \mathbb{R}^n 的一个子空间, 用符号 $Null(A)$ 表示, 即有

$$Null(A) = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{0}\} \quad (3.6)$$

定义 3.2.4. 左零空间是所有满足齐次线性方程组 $A^T\mathbf{y} = \mathbf{0}$ 的解向量集合, 它是 \mathbb{R}^n 的一个子空间, 用符号 $Null(A^T)$ 表示, 即有

$$Null(A^T) = \{\mathbf{y} \in \mathbb{R}^n \mid A^T\mathbf{y} = \mathbf{0}\} \quad (3.7)$$

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:

命题 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. 下面仅对 (1)(2) 进行证明, (3)(4) 也可以用类似的方法证明。(1)容易验证, 任何一种初等行变换都不改变行空间。事实上,

- 对于 I 型初等行变换 (用非零常数乘某一行), 有

$$\text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_i, \dots, \mathbf{r}_m\} = \text{span}\{\mathbf{r}_1, \dots, c\mathbf{r}_i, \dots, \mathbf{r}_m\}$$

- 对于 II 型初等行变换 (某一行的 c 倍加到另一行), 有

$$\text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_i + c\mathbf{r}_j, \dots, \mathbf{r}_j, \dots, \mathbf{r}_m\} = \text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_i, \dots, \mathbf{r}_j, \dots, \mathbf{r}_m\}$$

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

$$\begin{aligned} \mathbf{y} &= \beta_1\mathbf{r}_1 + \dots + \beta_i\mathbf{r}_i + \dots + \beta_j\mathbf{r}_j + \dots + \beta_m\mathbf{m} \\ &= \beta_1\mathbf{r}_1 + \dots + \beta_i(\mathbf{r}_i + c\mathbf{r}_j) + \dots + (\beta_j - c\beta_i)\mathbf{r}_j + \dots + \beta_m\mathbf{m} \end{aligned}$$

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), 有

$$\text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_i, \dots, \mathbf{r}_j, \dots, \mathbf{r}_m\} = \text{span}\{\mathbf{r}_1, \dots, \mathbf{r}_j, \dots, \mathbf{r}_i, \dots, \mathbf{r}_m\}$$

(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, 初等行变换不改变矩阵的零空间。□

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:

$$\begin{pmatrix} 1 & 2 & 1 \\ -1 & -1 & 1 \\ 1 & 4 & 5 \end{pmatrix} \xrightarrow[C_2-2C_1]{C_3-C_1} \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 2 \\ 1 & 2 & 4 \end{pmatrix} \xrightarrow[C_1+C_2]{C_3-2C_2} \mathbf{A}_C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 2 & 0 \end{pmatrix}$$

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 $-3\mathbf{r}_1 - 2\mathbf{r}_2 + \mathbf{r}_3 = 0$ 。

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

$$\begin{pmatrix} 1 & 2 & 1 \\ -1 & -1 & 1 \\ 1 & 4 & 5 \end{pmatrix} \xrightarrow[R_3-R_1]{R_2+R_1} \begin{pmatrix} 1 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 2 & 4 \end{pmatrix} \xrightarrow[R_3-2R_2]{R_1-2R_2} \mathbf{A}_R = \begin{pmatrix} 1 & 0 & -3 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{pmatrix}$$

\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 3.2.1. 设 $\mathbf{A} \in \mathbb{R}^{m \times n}$, 则 $\dim(Col(\mathbf{A})) = \dim(Row(\mathbf{A})) = \text{rank}(\mathbf{A})$

定理 3.2.2. 设 $\mathbf{A} \in \mathbb{R}^{m \times n}$ 则 $\dim(Null(\mathbf{A})) = n - \text{rank}(\mathbf{A})$

证明. 令 $r = \text{rank}(\mathbf{A})$, 根据定义 $Null(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{A}\mathbf{x} = \mathbf{0}\}$, 对 \mathbf{A} 做行初等变换并交换其中的一些列, \mathbf{A} 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。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 $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}'\mathbf{x} = \mathbf{0}\}$ 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 $\mathbf{A}\mathbf{x} = \mathbf{0}$ 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是求解矩阵 \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 3.2.3. 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。下面将继续讨论四个基本子空间之间的关系。设 $\mathbf{A} \in \mathbb{R}^{m \times n}$ ， \mathbf{A} 的四个基本子空间中， $\text{Col}(\mathbf{A}), \text{Null}(\mathbf{A}^T)$ 都是 \mathbb{R}^m 的子空间，它们是否有交集？ $\text{Col}(\mathbf{A}^T), \text{Null}(\mathbf{A})$ 都是 \mathbb{R}^n 的子空间，它们是否有交集？

定理 3.2.4. 设 $\mathbf{A} \in \mathbb{R}^{m \times n}$ ，

$$\text{Col}(\mathbf{A}) \cap \text{Null}(\mathbf{A}^T) = \{\mathbf{0}\}$$

$$\text{Col}(\mathbf{A}^T) \cap \text{Null}(\mathbf{A}) = \{\mathbf{0}\}$$

证明. 设 $\mathbf{v} \in \text{Col}(\mathbf{A}^T) \cap \text{Null}(\mathbf{A})$, 即 \mathbf{v} 在 $\mathbf{A} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m)^T$ 的行空间中且 $\mathbf{A}\mathbf{v} = \mathbf{0}$ 。设 $\mathbf{v} = a_1\mathbf{r}_1 + a_2\mathbf{r}_2 + \dots + a_m\mathbf{r}_m$, 则

$$\mathbf{A}\mathbf{v} = \mathbf{0} \implies \mathbf{r}_1^T \mathbf{v} = 0, \dots, \mathbf{r}_m^T \mathbf{v} = 0 \implies \mathbf{v}^T \mathbf{r}_1 = 0, \dots, \mathbf{v}^T \mathbf{r}_m = 0 \implies \mathbf{v} = \mathbf{0}$$

即

$$\text{Col}(\mathbf{A}^T) \cap \text{Null}(\mathbf{A}) = \{\mathbf{0}\}.$$

同理 $\text{Col}(\mathbf{A}) \cap \text{Null}(\mathbf{A}^T) = \{\mathbf{0}\}$ 。 \square

定义 3.2.5. 设 \mathbb{S} 和 \mathbb{T} 是 \mathbb{R}^n 的两个子空间。如果

$$\mathbb{S} \cap \mathbb{T} = \{\mathbf{0}\}$$

称 \mathbb{S} 和 \mathbb{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 \mathbb{S} 和 \mathbb{T} 是 \mathbb{R}^n 的两个子空间。如果对 $\forall \mathbf{v} \in \mathbb{S}, \forall \mathbf{w} \in \mathbb{T}$, 均有

$$\mathbf{v}^T \mathbf{w} = 0$$

则称 \mathbb{S} 垂直于 \mathbb{T} , \mathbb{T} 垂直于 \mathbb{S} , 记做 $\mathbb{S} \perp \mathbb{T}, \mathbb{T} \perp \mathbb{S}$ 。

或者说, 子空间 \mathbb{S} 和子空间 \mathbb{T} 是正交的。

定理 3.2.5. 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 \mathbb{S}, \mathbb{T} 不是无交连的, 则 $\exists \mathbf{v} \neq \mathbf{0}, \mathbf{v} \in \mathbb{S} \cap \mathbb{T}$, 而 $\mathbf{v}^T \mathbf{v} \neq 0$, 因而 \mathbb{S} 和 \mathbb{T} 不正交。从而正交的两个子空间必是无交连的。 \square

显然, 无交连的子空间不一定是正交的。如 $\text{span}\{(1, 1)^T\}$ 和 $\text{span}\{(1, 0)^T\}$ 。

例 3.2.2. 设 \mathbf{A} 是 $m \times n$ 阶阵, 则 $\text{Col}(\mathbf{A})$ 和 $\text{Null}(\mathbf{A}^T)$ 正交, $\text{Col}(\mathbf{A}^T)$ 和 $\text{Null}(\mathbf{A})$ 正交。

证明. 对 $\forall \mathbf{v} \in \text{Null}(\mathbf{A}^T)$, 则

$$\mathbf{v}^T \mathbf{A} = \mathbf{0} \implies \mathbf{v}^T \mathbf{a}_1 = 0, \mathbf{v}^T \mathbf{a}_2 = 0, \dots, \mathbf{v}^T \mathbf{a}_n = 0$$

对 $\forall \mathbf{w} \in \text{Col}(\mathbf{A})$, 有 $\mathbf{w} = \alpha_1 \mathbf{a}_1 + \alpha_2 \mathbf{a}_2 + \dots + \alpha_n \mathbf{a}_n$:

$$\mathbf{v}^T \mathbf{w} = \alpha_1 \mathbf{v}^T \mathbf{a}_1 + \alpha_2 \mathbf{v}^T \mathbf{a}_2 + \dots + \alpha_n \mathbf{v}^T \mathbf{a}_n = 0$$

因此, $\text{Null}(\mathbf{A}^T) \perp \text{Col}(\mathbf{A})$, $\text{Col}(\mathbf{A})$ 和 $\text{Null}(\mathbf{A}^T)$ 正交。将 \mathbf{A} 换成 \mathbf{A}^T , 我们得到 $\text{Col}(\mathbf{A}^T) \perp \text{Null}(\mathbf{A})$, $\text{Col}(\mathbf{A}^T)$ 和 $\text{Null}(\mathbf{A})$ 正交。 \square

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 $\mathbb{V} \subset \mathbb{R}^n$ 是一个子空间, \mathbb{V} 在 \mathbb{R}^n 中的正交补定义为集合

$$\{\mathbf{w} \in \mathbb{R}^n \mid \mathbf{v}^T \mathbf{w} = 0, \forall \mathbf{v} \in \mathbb{V}\}$$

记作 \mathbb{V}^\perp 。

也就是说 \mathbb{V} 的正交补空间是 \mathbb{R}^n 中所有和 \mathbb{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 $\mathbb{V} \perp \mathbb{V}^\perp$ 。同时 \mathbb{V} 与 \mathbb{V}^\perp 的和是直和, 因此, 对于 \mathbb{R}^n 中的任意向量 \mathbf{x} 可以唯一的分解成如下形式:

$$\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$$

其中 $\mathbf{x}_1 \in \mathbb{V}$, $\mathbf{x}_2 \in \mathbb{V}^\perp$, 并且 $\mathbf{x}_1^\top \mathbf{x}_2 = 0$ 。这种分解形式叫做向量的正交分解。

定理 3.2.6. 证明: $\text{Col}(\mathbf{A}^\top)^\perp = \text{Null}(\mathbf{A})$, $\text{Col}(\mathbf{A})^\perp = \text{Null}(\mathbf{A}^\top)$ 。

证明. 已知 $\text{Col}(\mathbf{A}^\top)$ 和 $\text{Null}(\mathbf{A})$ 是正交的, 也就是说

$$\text{Null}(\mathbf{A}) \subseteq \text{Col}(\mathbf{A}^\top)^\perp$$

对 $\forall \mathbf{x} \in \text{Col}(\mathbf{A}^\top)^\perp$, \mathbf{x} 和 $\text{Col}(\mathbf{A}^\top)$ 中的任意向量正交, 那么:

$$\mathbf{x}^\top \mathbf{r}_1 = 0, \mathbf{x}^\top \mathbf{r}_2 = 0, \dots, \mathbf{x}^\top \mathbf{r}_m = 0$$

即 $\mathbf{A}\mathbf{x} = \mathbf{0}$ 。说明 $\mathbf{x} \in \text{Null}(\mathbf{A})$, 也即

$$\text{Col}(\mathbf{A}^\top)^\perp \subseteq \text{Null}(\mathbf{A})$$

因此 $\text{Col}(\mathbf{A}^\top)^\perp = \text{Null}(\mathbf{A})$ 。同样可以证明 $\text{Col}(\mathbf{A})^\perp = \text{Null}(\mathbf{A}^\top)$ 。 \square

顾名思义, 子空间 \mathbb{V} 在向量空间 \mathbb{R}^n 的正交补空间 \mathbb{V}^\perp 含有正交和补充双重含义:

1. 子空间 \mathbb{V}^\perp 与 \mathbb{V} 正交;
2. 向量空间 \mathbb{R}^n 是子空间 \mathbb{V} 与 \mathbb{V}^\perp 的直和, 即 $\mathbb{R}^n = \mathbb{V} \oplus \mathbb{V}^\perp$ 。这表明, 向量空间 \mathbb{R}^n 是由子空间 \mathbb{V} 补充 \mathbb{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 \mathbb{V} 给定之后, 和 \mathbb{V} 正交的空间不一定是唯一的, 但是 \mathbb{V} 的正交补 \mathbb{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 3.2.7. (线性代数基本定理) 若 \mathbf{A} 是 $m \times n$ 矩阵,

- 1 [正交角度] $\text{Col}(\mathbf{A}^\top) \perp \text{Null}(\mathbf{A})$, $\text{Col}(\mathbf{A}) \perp \text{Null}(\mathbf{A}^\top)$,
- 2 [扩张角度] $\text{Col}(\mathbf{A}^\top) \oplus \text{Null}(\mathbf{A}) = \mathbb{R}^n$, $\text{Col}(\mathbf{A}) \oplus \text{Null}(\mathbf{A}^\top) = \mathbb{R}^m$,
- 3 [维数角度] $\dim \text{Col}(\mathbf{A}^\top) + \dim \text{Null}(\mathbf{A}) = n$, $\dim \text{Col}(\mathbf{A}) + \dim \text{Null}(\mathbf{A}^\top) = m$ 。

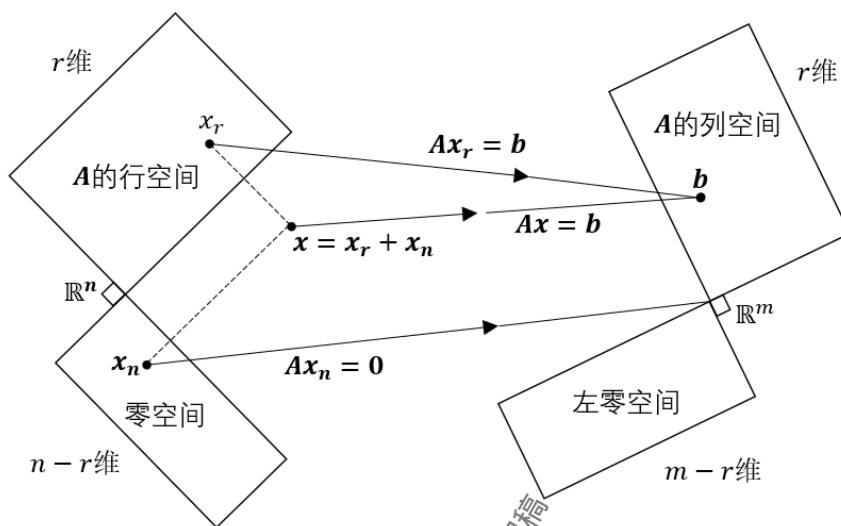


图 3.12: 四个子空间

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将丢失信息。为了最大程度地减少这种压缩损失，我们希望在数据中找到最有用的信息维度。然后，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。

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 \mathbf{P}_π 使得向量投影到某个 1 维子空间上。如图3.13所示。

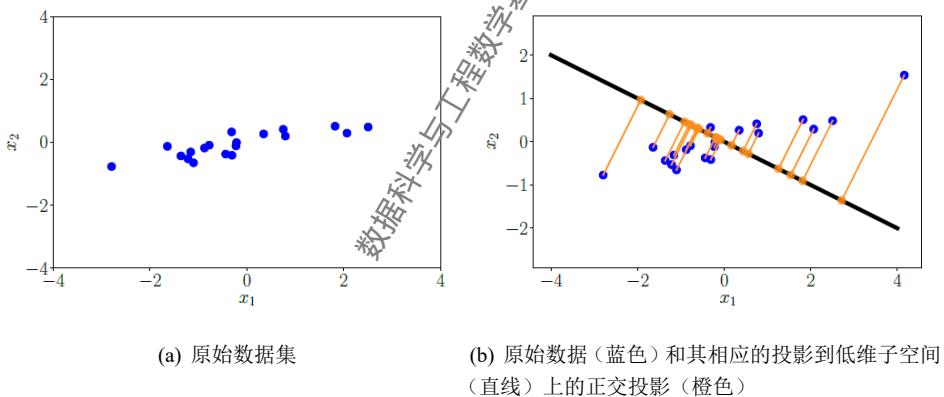


图 3.13: 将 2 维空间的点投影到 1 维子空间上。

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 \mathbf{P}_π 。

1. 确定 λ 因为 $\pi_U(x) \in \text{Col}(\mathbf{B})$ 是 x 的投影, 所以 $x - \pi_U(x) \in \text{Col}(\mathbf{B})^\perp = \text{Null}(\mathbf{B}^T)$, 有

$$\mathbf{b}^T(x - \pi_U(x)) = 0 \iff \mathbf{b}^T x - \lambda \mathbf{b}^T \mathbf{b} = 0$$

从而

$$\lambda = \frac{\mathbf{b}^T x}{\mathbf{b}^T \mathbf{b}}$$

或者利用内积和范数表示可得

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

2. 确定 $\pi_U(x)$ 因为 $\pi_U(x) = \lambda \mathbf{b}$, 由上面的结论可得:

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

我们可以给出 $\pi_U(x)$ 的长度

$$\begin{aligned} \|\pi_U(x)\| &= \|\lambda \mathbf{b}\| = |\lambda| \|\mathbf{b}\| \\ &= |\cos \omega| \frac{\|x\| \|\mathbf{b}\|}{\|\mathbf{b}\|} \frac{\|\mathbf{b}\|}{\|\mathbf{b}\|^2} \\ &= |\cos \omega| \|x\| \end{aligned}$$

其中 ω 是 x 和 \mathbf{b} 之间的夹角, $\cos \omega = \frac{\mathbf{b}^T x}{\|\mathbf{b}\| \|x\|}$

3. 确定投影矩阵 \mathbf{P}_π 投影矩阵 \mathbf{P}_π 是投影 $\pi_U(x)$ 对应的变换矩阵, 那么就有 $\pi_U(x) = \mathbf{P}_\pi x$, 则有

$$\pi_U(x) = \lambda \mathbf{b} = \mathbf{b} \lambda = \mathbf{b} \frac{\mathbf{b}^T x}{\|\mathbf{b}\|^2} = \frac{\mathbf{b} \mathbf{b}^T}{\|\mathbf{b}\|^2} x$$

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

$$\mathbf{P}_\pi = \frac{\mathbf{b} \mathbf{b}^T}{\mathbf{b}^T \mathbf{b}} = \frac{1}{9} \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 2 \end{pmatrix} = \frac{1}{9} \begin{pmatrix} 1 & 2 & 2 \\ 2 & 4 & 4 \\ 2 & 4 & 4 \end{pmatrix}$$

给定向量 $x = (1, 1, 1)^T$ 其投影为

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

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 $\mathbf{x} \in \mathbb{R}^m$ 投影到更低维的子空间 $\mathbb{U} \subseteq \mathbb{R}^m$ 中, 其中 $\dim(\mathbb{U}) = n \geq 1$ 。

设 $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ 是子空间 \mathbb{U} 的一个有序基底。 \mathbb{U} 上的任何投影 $\pi_{\mathbb{U}}(\mathbf{x})$ 必须是 \mathbb{U} 中的一个元素。故有

$$\pi_{\mathbb{U}}(\mathbf{x}) = \sum_{i=1}^n \lambda_i \mathbf{b}_i$$

和一维情况一样, 我们将逐步确定 $\lambda_1, \dots, \lambda_n$, $\pi_{\mathbb{U}}(\mathbf{x})$ 和投影矩阵 \mathbf{P}_{π} 。

1. 确定 $\lambda_1, \dots, \lambda_n$ 设

$$\pi_{\mathbb{U}}(\mathbf{x}) = \sum_{i=1}^n \lambda_i \mathbf{b}_i = \mathbf{B}\lambda \in \text{Col}(\mathbf{B})$$

最接近 $\mathbf{x} \in \mathbb{R}^m$, 其中 $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_n] \in \mathbb{R}^{m \times n}$, $\lambda = [\lambda_1, \dots, \lambda_n]^T \in \mathbb{R}^n$ 。

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

$$\begin{pmatrix} \mathbf{b}_1^T \\ \vdots \\ \mathbf{b}_n^T \end{pmatrix} (\mathbf{x} - \mathbf{B}\lambda) = \mathbf{0} \iff \mathbf{B}^T(\mathbf{x} - \mathbf{B}\lambda) = \mathbf{0} \iff \mathbf{B}^T\mathbf{B}\lambda = \mathbf{B}^T\mathbf{x}$$

最终的方程称为正规方程。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 P_π 由上面的讨论容易看出

$$P_\pi = B(B^T B)^{-1} B^T$$

例 3.2.4. 已知 \mathbb{R}^3 中的子空间 $U = \text{span} \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \right\}$ 和向量 $x = \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix}$, 确定 x 投影到 U 上的坐标 λ 、投影点 $\pi_U(x)$ 和投影矩阵 P_π

解. 首先确定 $B = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}$

其次计算

$$B^T B = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 3 & 3 \\ 3 & 5 \end{pmatrix}, \quad B^T x = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix}$$

然后只需要解方程 $B^T B \lambda = B^T x$ 得到 λ ,

$$\begin{pmatrix} 3 & 3 \\ 3 & 5 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \end{pmatrix} \Leftrightarrow \lambda = \begin{pmatrix} 5 \\ -3 \end{pmatrix}$$

故投影点 $\pi_U(x) = B\lambda = \begin{pmatrix} 5 \\ 2 \\ -1 \end{pmatrix}$ 。最后

$$P_\pi = B(B^T B)^{-1} B^T = \frac{1}{6} \begin{pmatrix} 5 & 2 & -1 \\ 2 & 2 & 2 \\ -1 & 2 & 5 \end{pmatrix}$$

我们还可以验证 $P_\pi^2 = P_\pi$

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 U 上。下面, 我们将讨论如何将向量投影到仿射子空间上。

考虑图3.14(a)。给定一个仿射空间 $L = x_0 + U$, 其中 b_1, b_2 是 U 的基向量。为了确定 x 在 L 上的正交投影 $\pi_L(x)$ 。我们将问题转化为知道如何解决的问题: 投影到向量子空间上。首先我们从 x 和 L 中减去支撑点 x_0 , 所以 $L - x_0 = U$ 。

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 $\pi_U(x - x_0)$, 如图3.14(b) 所示。

最后我们通过添加 x_0 将该投影转换回 L , 这样我们就可以得出仿射空间 L 上的正交投影为

$$\pi_L(x) = x_0 + \pi_U(x - x_0)$$

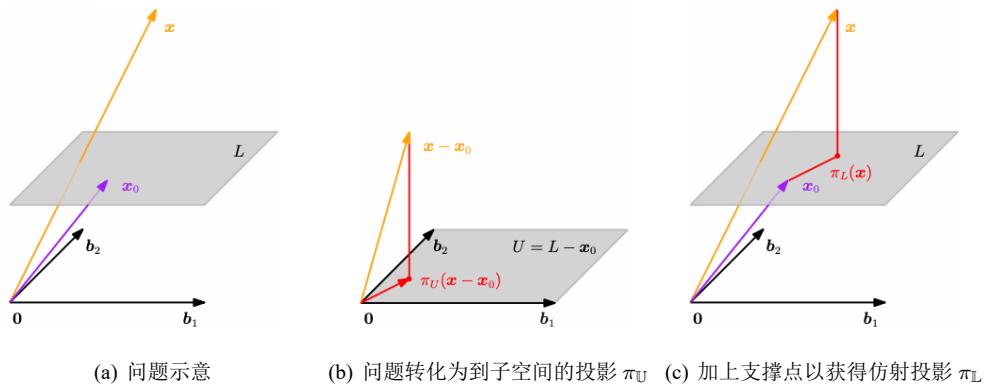


图 3.14: 投影到仿射空间。

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

$$\text{span}\{\mathbf{e}_1, \dots, \mathbf{e}_r\} = \text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_r\}$$

首先取 \mathbf{a}_1 作为一个基, 记为 \mathbf{b}_1 。那么 \mathbf{a}_2 可以正交分解为

$$\mathbf{a}_2 = \mathbf{a}_2^{(1)} + \mathbf{a}_2^{(2)},$$

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:

$$\begin{aligned} \mathbf{a}_{k+1}^{(1)} &= \pi_{\text{Col}(\mathbf{B}_k)}(\mathbf{a}_{k+1}) = \mathbf{B}_k (\mathbf{B}_k^T \mathbf{B}_k)^{-1} \mathbf{B}_k^T \mathbf{a}_{k+1} \\ &= (\mathbf{b}_1, \dots, \mathbf{b}_k) \begin{pmatrix} \langle \mathbf{b}_1, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_1, \mathbf{b}_k \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{b}_k, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_k, \mathbf{b}_k \rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle \mathbf{b}_1, \mathbf{a}_{k+1} \rangle \\ \vdots \\ \langle \mathbf{b}_k, \mathbf{a}_{k+1} \rangle \end{pmatrix} \end{aligned}$$

且 $\{\mathbf{b}_1, \dots, \mathbf{b}_k\}$ 是相互正交的。

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

$$\text{span}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r\} = \text{span}\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r\}.$$

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

$$\begin{aligned}\mathbf{b}_1 &= \mathbf{a}_1; \\ \mathbf{b}_2 &= \mathbf{a}_2 - \frac{\langle \mathbf{b}_1, \mathbf{a}_2 \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} \mathbf{b}_1; \\ &\vdots \\ \mathbf{b}_r &= \mathbf{a}_r - \frac{\langle \mathbf{b}_1, \mathbf{a}_r \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} \mathbf{b}_1 - \frac{\langle \mathbf{b}_2, \mathbf{a}_r \rangle}{\langle \mathbf{b}_2, \mathbf{b}_2 \rangle} \mathbf{b}_2 - \cdots - \frac{\langle \mathbf{b}_{r-1}, \mathbf{a}_r \rangle}{\langle \mathbf{b}_{r-1}, \mathbf{b}_{r-1} \rangle} \mathbf{b}_{r-1}\end{aligned}$$

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。

解. 取

$$\begin{aligned}\mathbf{b}_1 &= (3, 1, 1)^T \\ \mathbf{b}_2 &= \mathbf{a}_2 - \frac{\mathbf{b}_1^T \mathbf{a}_2}{\mathbf{b}_1^T \mathbf{b}_1} \mathbf{b}_1 = (2, 2, 0)^T - \frac{8}{11} (3, 1, 1)^T = \frac{-2}{11} (1, -7, 4)^T \\ \mathbf{e}_1 &= \frac{1}{\sqrt{11}} (3, 1, 1)^T \\ \mathbf{e}_2 &= \frac{1}{\sqrt{66}} (1, -7, 4)^T\end{aligned}$$

故标准正交基为 $\mathbf{e}_1, \mathbf{e}_2$ 。

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 $\theta > 0$ 则保持以下约定：

- 原点是固定的点
- 一般，旋转方向规定为逆时针

例 3.4.1. 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：

$$\Phi_\theta(\mathbf{x}) = x_1 \Phi_\theta(\mathbf{e}_1) + x_2 \Phi_\theta(\mathbf{e}_2) = \left[\Phi_\theta(\mathbf{e}_1), \Phi_\theta(\mathbf{e}_2) \right] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

所以平面上旋转 θ 的变换矩阵 \mathbf{R}_θ 为：

$$\mathbf{R}_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

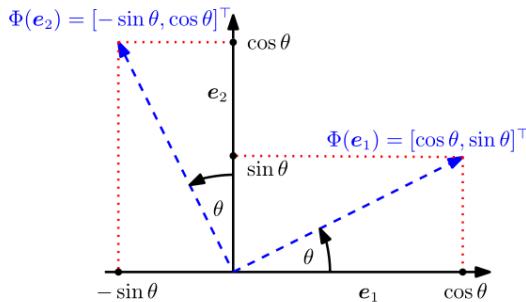


图 3.15: 平面中的旋转

三维空间中的旋转

例 3.4.2. 对于 \mathbb{R}^3 中的向量 x , 设 \mathbb{R}^3 的三个基底分别为 e_1, e_2, e_3 。若 x 绕 e_3 旋转 θ , 记为 Φ_θ^3 , 类似 2 维空间的做法

$$\Phi_\theta^3(e_1) = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix}, \quad \Phi_\theta^3(e_2) = \begin{bmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{bmatrix}, \quad \Phi_\theta^3(e_3) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

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:

$$R_\theta^2 = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}$$

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。

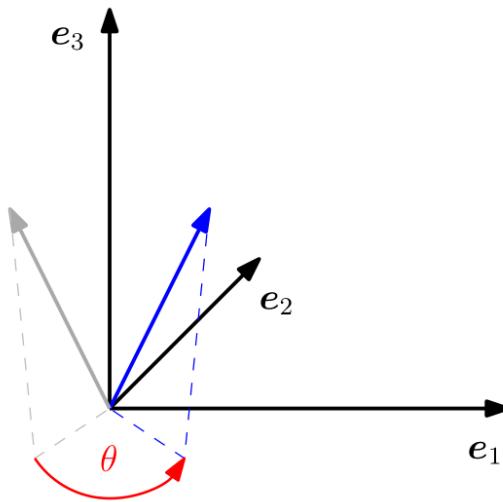


图 3.16: 三维空间中的旋转

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

$$R_{i,j}(\theta) := \begin{bmatrix} I_{i-1} & & & & & \\ & \cos \theta & & & & -\sin \theta \\ & \sin \theta & & I_{j-i-1} & & \\ & & & & \cos \theta & \\ & & & & & I_{n-j} \end{bmatrix}$$

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。

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 $R \in \mathbb{R}^{n \times n}$, R 是旋转矩阵当且仅当它是正交矩阵并且 $\det(R) = 1$ 。

性质 3.4.2. 保距性: 设 $R_\theta \in \mathbb{R}^{n \times n}$ 是旋转矩阵, $\forall x, y \in \mathbb{R}^n$, 有 $\|x - y\|_2 = \|R_\theta(x) - R_\theta(y)\|_2$ 。

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 $R_\theta \in \mathbb{R}^{n \times n}$ 是旋转矩阵, $\forall x, y \in \mathbb{R}^n \setminus \{\mathbf{0}\}$, 有 $\frac{\langle x, y \rangle}{\|x\| \|y\|} = \frac{\langle R_\theta(x), R_\theta(y) \rangle}{\|R_\theta(x)\| \|R_\theta(y)\|}$ 。

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 \mathbf{e}_3 绕 \mathbf{e}_3 旋转 $\pi/2$ 仍是 \mathbf{e}_3 , 再绕 \mathbf{e}_2 旋转 $\pi/2$ 会变换到 \mathbf{e}_1 。如果 \mathbf{e}_3 先绕 \mathbf{e}_2 旋转 $\pi/2$ 到 \mathbf{e}_1 , 再绕 \mathbf{e}_3 旋转 $\pi/2$ 会变换到 \mathbf{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 $\mathcal{O}(n)$ 。

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 $\mathcal{SO}(n)$ 。当 $n = 2$ 时, $\mathcal{SO}(2)$ 是阿贝尔群。

例 3.4.8. $\mathcal{SO}(n)$ 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。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。

由题可得:

$$\begin{aligned} \arg \min_{\mathbf{Q} \in \mathcal{O}(n)} \sum_{t=1}^T \frac{1}{2} \|\mathbf{Q}\mathbf{x}_t - \mathbf{y}_t\|^2 &= \arg \min_{\mathbf{Q} \in \mathcal{O}(n)} T - \left(\sum_{t=1}^T \mathbf{y}_t^T \mathbf{Q} \mathbf{x}_t \right) \\ &= \arg \max_{\mathbf{Q} \in \mathcal{O}(n)} \text{Tr} \left(\left(\sum_{t=1}^T \mathbf{x}_t \mathbf{y}_t^T \right) \mathbf{Q} \right) \end{aligned}$$

记 $\mathbf{S} := \sum_{t=1}^T \mathbf{x}_t \mathbf{y}_t^T$, 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。

例 3.4.10. 考虑二维平面中的一个向量 $\mathbf{x} = (x_1, x_2)^T$, $\mathbf{b} = (\cos \theta, \sin \theta)^T$, 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

$$\mathbf{x}' = \mathbf{u} - \mathbf{v} = \mathbf{x} - 2\mathbf{v} = 2\mathbf{u} - \mathbf{x}$$

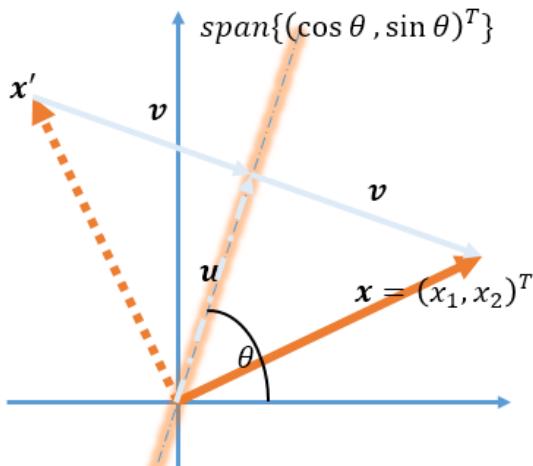


图 3.17: 平面上的镜像反射

根据投影公式 $u = b b^T x$, 那么

$$\begin{aligned} u &= \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta \end{pmatrix} x \\ &= \begin{pmatrix} \cos^2 \theta & \cos \theta \sin \theta \\ \cos \theta \sin \theta & \sin^2 \theta \end{pmatrix} x \end{aligned}$$

则

$$\begin{aligned} x' &= 2u - x = \begin{pmatrix} 2\cos^2 \theta - 1 & 2\cos \theta \sin \theta \\ 2\cos \theta \sin \theta & 2\sin^2 \theta - 1 \end{pmatrix} x \\ &= \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix} x \end{aligned}$$

令 $\phi = 2\theta$,

$$x' = \begin{pmatrix} \cos \phi & \sin \phi \\ \sin \phi & -\cos \phi \end{pmatrix} x$$

或者我们记 $(\text{span}\{b\})^\perp$ 中的单位向量为 w , 容易求得 $w = (\sin \theta, -\cos \theta)^T$ 。 v 是 x 在 $\text{span}\{w\}$ 上的投影, 即 $v = w w^T x$ 。利用

$$\begin{aligned} x' &= x - 2v = (I - 2ww^T)x \\ x' &= \begin{pmatrix} 1 - 2\sin^2 \theta & 2\cos \theta \sin \theta \\ 2\cos \theta \sin \theta & 1 - 2\cos^2 \theta \end{pmatrix} x = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix} x \end{aligned}$$

我们可以得到同样的结论。

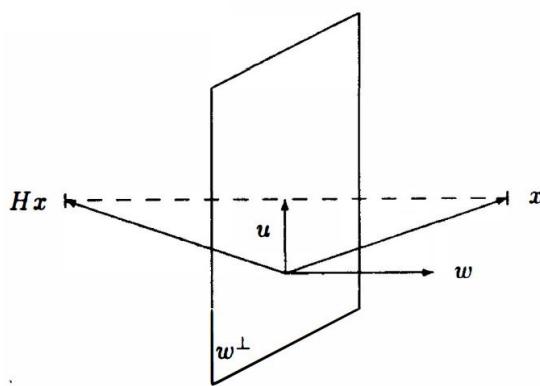


图 3.18: Householder 变换是关于 w 的垂直超平面的镜面反射。

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 w 可以很容易的求出关于 w 垂直的超平面的镜像反射。

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 $x \in \mathbb{R}^n$, 如图 3.18 所示, Hx 是 x 关于 w 的垂直超平面的镜像反射。

证明. (1) 显然成立。(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 $\mathbf{x} \in \mathbb{R}^n$, 则 \mathbf{x} 可表示为 $\mathbf{x} = \mathbf{u} + \alpha \mathbf{w}$

其中 $\mathbf{u} \in \text{span}\{\mathbf{w}\}^\perp$, $\alpha \in \mathbb{R}$ 。利用 $\mathbf{u}^T \mathbf{w} = 0$ 和 $\mathbf{w}^T \mathbf{w} = 1$, 可得

$$\begin{aligned}\mathbf{Hx} &= (\mathbf{I} - 2\mathbf{w}\mathbf{w}^T)(\mathbf{u} + \alpha\mathbf{w}) \\ &= \mathbf{u} + \alpha\mathbf{w} - 2\mathbf{w}\mathbf{w}^T\mathbf{u} - 2\alpha\mathbf{w}\mathbf{w}^T\mathbf{w} \\ &= \mathbf{u} - \alpha\mathbf{w}\end{aligned}$$

□

这就说明了 \mathbf{Hx} 为 \mathbf{x} 关于 $\text{span}\{\mathbf{w}\}^\perp$ 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 。

$\mathbb{R}^{n \times n}$ 中的 Householder 矩阵 $\mathbf{H} = \mathbf{I} - 2\mathbf{w}\mathbf{w}^T$ 将 \mathbf{x} 变换到关于 \mathbf{w} 的垂直超平面 $\text{span}\{\mathbf{w}\}^\perp$ 的镜像上, 这里 \mathbf{w} 仍是单位向量。而在 $\text{span}\{\mathbf{w}\}^\perp$ 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 $\mathbf{Hx} = \mathbf{x}$, 若 $\mathbf{x} \in \text{span}\{\mathbf{w}\}^\perp$ 。而 $\text{span}\{\mathbf{w}\}^\perp$ 是 $n-1$ 维的, 也就是说 \mathbf{H} 至少有 $n-1$ 个特征值是 1 。而在 $\text{span}\{\mathbf{w}\}$ 上的每个向量 $\mathbf{x} = \alpha\mathbf{w}$, 它们关于这个超平面的镜像是 $-\mathbf{x}$ 。因为

$$\mathbf{Hx} = (\mathbf{I} - 2\mathbf{w}\mathbf{w}^T)\mathbf{x} = \alpha\mathbf{w} - 2\alpha\mathbf{w}\mathbf{w}^T - \alpha\mathbf{w} = -\mathbf{x}, \text{若 } \mathbf{x} \in \text{span}\{\mathbf{w}\}.$$

那么 \mathbf{H} 至少有 1 个特征值是 -1 。

\mathbf{H} 一共有 n 个特征值, 所以 \mathbf{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 $\det(\mathbf{H}) = -1$ 。

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

小波模型

小波分析是图像重构的一种重要方法。它通过不同尺度变化对失真图像进行多尺度分析, 进而保留想要的尺度信息, 去掉噪声等对应的干扰信息。小波分析的一个最重要的概念是小波框架, 它是空间中基函数的推广。具体地, 将图像理解成一个向量 $\mathbf{x} \in \mathbb{R}^n$, 令 $\mathbf{W} \in \mathbb{R}^{m \times n}$ 为小波框架。需要注意的是, 这里 m 可以比 n 大, 但是有 $\text{rank}(\mathbf{W}) = n$, 也就意味着 \mathbf{W} 带有一些冗余信息。在小波框架下, 可以对图像 \mathbf{x} 做分解得到小波系数 $\boldsymbol{\alpha} \in \mathbb{R}^m$, 即 $\boldsymbol{\alpha} = \mathbf{Wx}$ 。反之, 给定小波系数 $\boldsymbol{\alpha}$, 可以重构出图像 $\mathbf{x} = \mathbf{W}^T \boldsymbol{\alpha}$ 。为了保证重构的完整性, 我们要求 $\mathbf{W}^T \mathbf{W} = \mathbf{I}$ 。因为冗余性, 所以 $\mathbf{W} \mathbf{W}^T \neq \mathbf{I}$ 。

对于一张图像而言, 只有少数的小波系数对原始图像起到决定作用. 我们考虑基于小波框架的重构模型. 常用的有

- 分解模型: 直接求解重构图像, 其通过惩罚图像的小波系数的 l_1 范数来去除图像中不必要的噪声信息. 问题形式为

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\lambda \odot (\mathbf{Wx})\|_1 + \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2,$$

其中 \mathbf{b} 为实际观测的图像数据, $\lambda \in \mathbb{R}^m$ 是给定的非负向量, \odot 表示逐个分量相乘.

- 合成模型: 求解图像对应的小波系数来重构图像, 其通过小波系数的 l_1 范数来去除图像中不必要的噪声信息. 问题形式为

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\lambda \odot \boldsymbol{\alpha}\|_1 + \frac{1}{2} \|\mathbf{AW}\boldsymbol{\alpha} - \mathbf{b}\|_2^2,$$

- 平衡模型: 求解图像对应的小波系数来重构图像. 在合成模型中, $\boldsymbol{\alpha}$ 不一定对应于真实图像的小波系数. 因此, 平衡模型添加 $(\mathbf{I} - \mathbf{WW}^T)\boldsymbol{\alpha}$ 的二次罚项来保证 $\boldsymbol{\alpha}$ 更接近真实图像的小波系数. 问题形式为

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\lambda \odot \boldsymbol{\alpha}\|_1 + \frac{1}{2} \|\mathbf{AW}\boldsymbol{\alpha} - \mathbf{b}\|_2^2 + \frac{\kappa}{2} \|(\mathbf{I} - \mathbf{WW}^T)\boldsymbol{\alpha}\|_2^2,$$

其中 κ 为给定常数.

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,

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Haar 矩阵的构造

当 $n = 4$ 时, 我们可以构造 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 \mathbf{H}_2 的第一行 $[1, 1]$ 中的每个 1.

(2) 我们用 $[1, 1]$ 替换掉 \mathbf{H}_2 第二行中的每个 1, 得到 \mathbf{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, 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 $2j-1, 2j$ ($j = 1, 2, 3, 4$) 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。

$$\mathbf{H}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

Haar 矩阵变换的特点

哈尔小波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。 \square

例 3.4.11. 当 $n = 2^2 = 4$ 时, Hadamard 矩阵为

$$\mathbf{H}_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

例 3.4.12.

$$\mathbf{H}_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

(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 H_2 线性变换得到的向量为

$$(7, -1)^T$$

Hadamar 在信号处理中的优势

当 H 为 Hadamar 矩阵时, 若 H 作为线性空间的一组基, 由于 Hadamar 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 H 的阶数。

线性变换 $y = Hx$ 称为 Hadamar 变换。同样由于 Hadamar 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 Hadamar 矩阵的紧凑表示

矩阵的克罗内克 (Kronecker) 积

定义 3.4.5. 设 $A = (a_{ik})_{m \times n}$, $B = (b_{rl})_{r \times s}$ 是域 \mathbb{K} 中的两个矩阵, 则矩阵 $C = (c_{\lambda\mu})_{mr \times ns}$ 称为 A 与 B 的克罗内克积, 记作 $C = A \otimes B$, 即

$$C = A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}$$

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:

- $A \otimes (B_1 + B_2) = A \otimes B_1 + A \otimes B_2$, $A \in \mathbb{R}^{m \times n}$, $B_1, B_2 \in \mathbb{R}^{p \times q}$
- $(A_1 + A_2) \otimes B = A_1 \otimes B + A_2 \otimes B$, $A_1, A_2 \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$
- $A \otimes (B \otimes C) = (A \otimes B) \otimes C = A \otimes B \otimes C$, $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$, $C \in \mathbb{R}^{k \times l}$
- $(A \otimes B)^T = A^T \otimes B^T$, $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$
- $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$, $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$, $C \in \mathbb{R}^{n \times r}$, $D \in \mathbb{R}^{q \times s}$

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

$$X^{\otimes n} = \underbrace{X \otimes X \otimes \cdots \otimes X}_{n \text{ 次}}, \bigotimes_{i=1}^n X_i = X_1 \otimes X_2 \otimes \cdots \otimes X_n$$

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，这里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

$$\bar{\mathbf{A}} = (\bar{a}_{ij})_{n \times n}.$$

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

- $\overline{\mathbf{A} + \mathbf{B}} = \bar{\mathbf{A}} + \bar{\mathbf{B}}$;
- $\overline{(k\mathbf{A})} = \bar{k}\bar{\mathbf{A}}$;
- $\overline{\mathbf{AB}} = \bar{\mathbf{A}}\bar{\mathbf{B}}$;
- $\overline{(\mathbf{A}^{-1})} = (\bar{\mathbf{A}})^{-1}$ 。

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

$$\mathbf{A}^H = \overline{(\mathbf{A}^T)} = (\bar{\mathbf{A}})^T = (\bar{a}_{ji})_{n \times n}.$$

例 3.4.13. 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 $A \in \mathbb{C}^{n \times m}, B \in \mathbb{C}^{m \times n}$, 则 $(A + B)^H = A^H + B^H$
- 设 $A \in \mathbb{C}^{n \times m}, B \in \mathbb{C}^{m \times k}$, 则 $(AB)^H = B^H A^H$
- 设 $A \in \mathbb{C}^{n \times m}, k \in \mathbb{C}$, 则有 $(kA)^H = \bar{k}A^H$
- 设 $A \in \mathbb{C}^{n \times m}$, 则有 $(A^H)^H = A$
- 设 $A \in \mathbb{C}^{n \times n}$ 可逆, 则有 $(A^{-1})^H = (A^H)^{-1}$

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 3.4.14. 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
- $\|\mathbf{Ux}\|_2 = \|\mathbf{x}\|_2$

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

$$\mathbf{F}_n = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \omega_n^3 & \omega_n^4 & \dots & \omega_n^{(n-1)} \\ 1 & \omega_n^2 & \omega_n^4 & \omega_n^6 & \omega_n^8 & \dots & \omega_n^{2(n-1)} \\ 1 & \omega_n^3 & \omega_n^6 & \omega_n^9 & \omega_n^{12} & \dots & \omega_n^{3(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega_n^{(n-1)} & \omega_n^{2(n-1)} & \omega_n^{3(n-1)} & \omega_n^{4(n-1)} & \dots & \omega_n^{(n-1)^2} \end{pmatrix}$$

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 $\mathbf{F}_{jk} = \frac{1}{\sqrt{n}} \omega_n^{jk} = \frac{1}{\sqrt{n}} e^{\frac{2\pi i}{n} jk}$ 。

例 3.4.15. 二阶的傅里叶矩阵为

$$\mathbf{F}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

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

$$\mathbf{f}_i^H \mathbf{f}_i = \left(\frac{1}{\sqrt{n}} \right)^2 \sum_{j=0}^{n-1} \omega_n^{ij} (\overline{\omega_n^{ij}}) = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{ij} \omega_n^{n-ij} = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^n = 1$$

而当 $i \neq k$ 时,

$$\mathbf{f}_i^H \mathbf{f}_k = \left(\frac{1}{\sqrt{n}} \right)^2 \sum_{j=0}^{n-1} \omega_n^{ij} (\overline{\omega_n^{kj}}) = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{ij} \omega_n^{n-kj} = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{n+(i-k)j} = \frac{1}{n} \sum_{j=0}^{n-1} \omega_n^{(i-k)j}$$

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

$$(x - 1)(x^{n-1} + x^{n-2} + \dots + x^2 + x + 1) = 0.$$

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。

相位恢复问题

相位恢复是信号处理中的一个重要问题, 它是从信号在某个变换域的幅度测量值来恢复该信号。其问题背景如下: 将待测物体(信号)放置在指定位置, 用透射光照射, 经过衍射成像, 可以由探测器得到其振幅分布。我们需要从该振幅分布中恢复出原始信号的信息。由 Fraunhofer 衍射方程可知, 探测器处的光场可以被观测物体的傅里叶变换很好地逼近。但是因为实际中的探测器只能测量光的强度, 因此我们只能得到振幅信息。

信号的相位通常包含丰富的信息。我们可以对一张图片 Y 做二维离散傅里叶变换 \mathcal{F} 得到 $\mathcal{F}(Y)$ 。由于变换后的图片 $\mathcal{F}(Y)$ 是复数矩阵, 它可以由模长 $|\mathcal{F}(Y)|$ 和相位 $\text{phase}(\mathcal{F}(Y))$ 来表示, 即

$$\mathcal{F}(Y) = |\mathcal{F}(Y)| \odot \text{phase}(\mathcal{F}(Y)),$$

○ 表示矩阵对应元素相乘。如果我们将另外一张图片 S 的相位信息代替原图片的相位信息则

$$\hat{S} = \mathcal{F}^{-1}(|\mathcal{F}(Y)| \odot \text{phase}(\mathcal{F}(S))),$$

那么得到的 \hat{S} 将会与 S 差不多, 可见相位信息比模长信息更加重要。

在实际应用中, 我们不一定使用傅里叶变换对原始信号进行采样处理。给定复信号 $\mathbf{x} = (x_0, x_1, x_2, \dots, x_{n-1})^T \in \mathbb{C}^n$ 以及采样数 m , 我们可以逐分量定义如下线性变换:

$$(\mathcal{A}(\mathbf{x}))_k = \langle a_k, \mathbf{x} \rangle, k = 1, 2, \dots, m$$

如果将其对应的振幅观测记为 b_k , 那么相位恢复问题本质上是求解如下的二次方程组:

$$b_k^2 = |\langle a_k, \mathbf{x} \rangle|^2, k = 1, 2, \dots, m.$$

虽然求解线性方程组很简单, 但是求解二次方程组问题却是 NP 难的。

通常可以将次问题转化为非线性最小二乘问题:

$$\min_{\mathbf{x} \in \mathbb{C}^n} \sum_{i=1}^m (|\langle a_i, \mathbf{x} \rangle|^2 - b_i^2)^2$$

这个模型的目标函数是可微 (Wirtinger 导数) 的四次函数, 是非凸优化问题。

另一种方法是相位提升 (phase lift)。相位恢复问题本质的困难在于处理二次方程组。注意到

$$|\langle a_i, x \rangle|^2 = \text{Tr}(\mathbf{x} \bar{\mathbf{x}}^T \mathbf{a}_i \bar{\mathbf{a}}_i^T),$$

令 $\mathbf{X} = \mathbf{x} \bar{\mathbf{x}}^T$, 方程组可以转化为

$$\text{Tr}(\mathbf{X} \mathbf{a}_i \bar{\mathbf{a}}_i^T) = b_i^2, i = 1, 2, \dots, m, \mathbf{X} \succcurlyeq 0, \text{rank}(\mathbf{X}) = 1$$

所以得到优化问题

$$\begin{aligned} \min_{\mathbf{X}} \quad & \text{rank}(\mathbf{X}), \\ \text{s.t.} \quad & \text{Tr}(\mathbf{X} \mathbf{a}_i \bar{\mathbf{a}}_i^T) = b_i^2, i = 1, 2, \dots, m \\ & \mathbf{X} \succcurlyeq 0 \end{aligned}$$

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, 内积不仅可以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”, 此外, 概率回归中的高斯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,。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. 证明: 假设向量 β 可以经向量组 $\alpha_1, \alpha_2, \dots, \alpha_r$ 线性表出, 证明: 表示方式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 $A\mathbf{x} = 12\mathbf{x}$ 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 $A\mathbf{x} = \mathbf{b}$ 中所有解的集合 \mathbb{S} , 其中 A 和 \mathbf{b} 定义如下:

(1)

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

(2)

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

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 $AB, AB - BA$

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个 α_i ($1 < i \leq s$) 可被 $\alpha_1, \alpha_2, \dots, \alpha_{i-1}$ 线性表示。

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) $\beta = (1, 2, 1, 1)$, $\alpha_1 = (1, 1, 1, 1)$, $\alpha_2 = (1, 1, -1, -1)$, $\alpha_3 = (1, -1, 1, -1)$, $\alpha_4 = (1, -1, -1, 1)$;

(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 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)

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

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

(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

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

(1) 求 $\ker(\Phi)$, $\text{Im}(\Phi)$ 。

(2) 确定关于基 B 的变换矩阵 \tilde{A}_Φ 。

$$B = \left(\left[\begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ 2 \\ 1 \end{array} \right], \left[\begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right] \right)$$

习题 3.16. 已知 \mathbb{R}^3 标准基下向量 $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$

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

令 $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ 。

(1) 证明 \mathbf{C} 是 \mathbb{R}^3 的基。

(2) $\mathbf{C}' = (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ 是 \mathbb{R}^3 的标准基。计算从 \mathbf{C}' 到 \mathbf{C} 的过渡矩阵 \mathbf{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 $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2)$ 并且 $\mathbf{B}' = (\mathbf{b}'_1, \mathbf{b}'_2)$

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

求 \mathbf{B}' 到 \mathbf{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 $\mathbf{x}^T \mathbf{A} \mathbf{x} = \text{Tr}(\mathbf{x}^T \mathbf{A} \mathbf{x})$ 和 $\mathbf{x}^T \mathbf{A} \mathbf{x} = \text{Tr}(\mathbf{A} \mathbf{x} \mathbf{x}^T)$

习题 3.20. t 取什么值时, 下列二次型是正定的:

$$(1) \quad x_1^2 + x_2^2 + 5x_3^2 + 2tx_1x_2 - 2x_1x_3 + 4x_2x_3$$

$$(2) \quad x_1^2 + 4x_2^2 + x_3^2 + 2tx_1x_2 + 10x_1x_3 + 6x_2x_3$$

$$\text{习题 3.21. 设 } \mathbf{A} = \begin{pmatrix} 1 & 4 & 2 \\ 0 & -3 & 4 \\ 0 & 4 & 3 \end{pmatrix} \quad \text{求 } \mathbf{A}^k$$

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 $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$, 如何计算 (唯一) 矩阵 \mathbf{A} , 对每一个 $\mathbf{x} \in \mathbb{R}^n$ 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 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 \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 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 Calafiori 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.

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一分解，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：

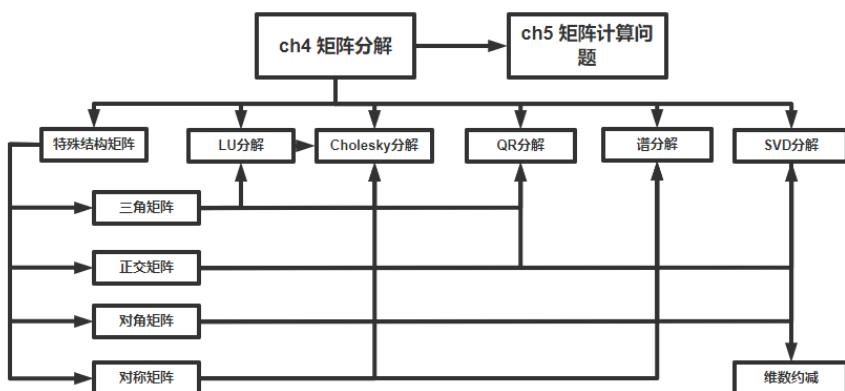


图 4.1: 本章内容概览

4.1 数学中常见的具有特殊结构的矩阵

方阵

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。很多概念只有方阵才有。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可得一些特殊的方阵, 如对称矩阵和正半定矩阵。

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

$$x^T A x \geq 0$$

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

$$x^T A x > 0$$

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具有重要地位, 它们是很多数据表示和建模的矩阵, 很多机器学习模型可以用它们来进行表示。上面介绍的几个矩阵连同 $n \times n$ 的非方阵是我们要分解的主要对象。接下来介绍几个用于表示分解的具有简单结构和性质的矩阵。

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 $A = \text{diag}(\mathbf{a}) = \text{diag}(a_1, a_2, \dots, a_n)$ 。这里 \mathbf{a} 是 n 维向量, 包含了矩阵 A 的全部对角元素。

$$A = \text{diag}(a_1, a_2, \dots, a_n) = \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{pmatrix}$$

容易验证, 对于对角矩阵:

$$A^k = \begin{pmatrix} a_1^k & & \\ & \ddots & \\ & & a_n^k \end{pmatrix}$$

如果 A 是可逆矩阵, 即矩阵 A 的对角元都不为零, 则有:

$$A^{-1} = \begin{pmatrix} \frac{1}{a_1} & & \\ & \ddots & \\ & & \frac{1}{a_n} \end{pmatrix}$$

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 A 的所有元素满足 $i > j$ 时, $a_{ij} = 0$, 则称 A 为上三角矩阵

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$$

定义 4.1.2. 若矩阵 A 的所有元素满足 $i < j$ 时, $a_{ij} = 0$, 则称 A 为下三角矩阵

$$A = \begin{bmatrix} a_{11} & & \\ \vdots & \ddots & \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$$

性质 4.1.1. 三角矩阵性质

- 三角矩阵主对角线上元素均非零 \Leftrightarrow 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 $A \in \mathbb{R}^{n \times n}$ 为三角矩阵, $k \in \mathbb{Z}$, 那么矩阵 A^k 主对角线上的元素 $(A^k)_{ii} = (A_{ii})^k, i = 1, 2, \dots, n$

正交矩阵

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

$$A^T A = A A^T = I$$

从定义上可以看出

$$A^{-1} = A^T$$

正交矩阵求逆，只需对矩阵转置即求得矩阵的逆。

性质 4.1.2. 正交矩阵的正交性 设 $A = [a_1, \dots, a_n]$ ，并且 A 是一个正交矩阵，那么

$$a_i^T a_j = \begin{cases} 1 & \text{如果 } i = j \\ 0 & \text{如果 } i \neq j \end{cases}$$

正交矩阵因为有列正交性，以其作为基底，则可以大大减少我们的计算量。

性质 4.1.3. 和范数有关的性质 如果矩阵 $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ 是正交矩阵， $M \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^m$

- $\|U\|_2 = 1$, $\|U\|_F = \sqrt{m}$
- $\|Ux\|_2 = \|x\|_2$, $\|Ux\|_F = \|x\|_F$
- $\|UMV\|_2 = \|M\|_2$, $\|UMV\|_F = \|M\|_F$

Dyads (并向量或单纯矩阵或秩一矩阵)

定义 4.1.3. 矩阵 $A \in \mathbb{R}^{m \times n}$ 如果具有如下形式：

$$A = \mathbf{u}\mathbf{v}^T$$

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，则我们称其为秩一矩阵。

如果 \mathbf{v}, \mathbf{u} 具有相同维度，则 dyad $A = \mathbf{u}\mathbf{v}^T$ 就是一个方阵。

一个 dyad $A = \mathbf{u}\mathbf{v}^T$ 对于输入向量 $x \in \mathbb{R}^n$ 有如下作用：

$$Ax = (\mathbf{u}\mathbf{v}^T)x = (\mathbf{v}^T x)\mathbf{u}$$

- 因为 $A_{ij} = \mathbf{u}_i \mathbf{v}_j$ ；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 $A = \mathbf{u}\mathbf{v}^T$ ，由对应的线性映射 $x \rightarrow Ax$ 可知，无论输入 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 $\mathbf{v}^T x$ ，即取决于向量 \mathbf{v} 。

- 对于一个 dyad $A = \mathbf{u}\mathbf{v}^T$, 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 $A = \mathbf{u}\mathbf{v}^T$ 称为秩一矩阵。

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} 。

对于一个 dyad $A = \mathbf{u}\mathbf{v}^T$, 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:

$$A = \mathbf{u}\mathbf{v}^T = (\|\mathbf{u}\|_2 \cdot \|\mathbf{v}\|_2) \frac{\mathbf{u}}{\|\mathbf{u}\|_2} \frac{\mathbf{v}^T}{\|\mathbf{v}\|_2} = \sigma \tilde{\mathbf{u}} \tilde{\mathbf{v}}^T$$

其中 $\sigma > 0$, 并且 $\|\tilde{\mathbf{u}}\| = \|\tilde{\mathbf{v}}\| = 1$ 。

例 4.1.1. 一般在推荐系统中, 数据往往使用“用户——物品”矩阵来表示。表 4.2 表示了 m 个矩阵, n 个物品的矩阵。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, 因此产生矩阵补全问题。

	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					2		
用户3		1		2			5			
用户4			3					3		5
用户5	5				2					

图 4.2: 用户-物品表

矩阵补全问题一般可表示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

$$\begin{aligned} \min_X \quad & \text{rank}(X) \\ \text{s.t.} \quad & X_{ij} = M_{ij} \quad \forall i, j \in \mathbb{E} \end{aligned}$$

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

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

利用秩一分解, 将 X 看作 $\sum_{i=k}^r \mathbf{f}_k \mathbf{g}_k^T$, 其中 \mathbf{f} 是列向量, \mathbf{g} 是行向量。 $X_{ij} = \sum_k \mathbf{f}_k[i] \mathbf{g}_k[j]$, 优化问题可以进一步写作:

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

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：

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

所谓相容维就是指 A_{11}, A_{12} 行数一样, A_{11}, A_{21} 列数一样。当 A 是方阵, 并且 $A_{12} = \mathbf{0}, A_{21} = \mathbf{0}$, 那么称 A 为块对角矩阵:

$$A = \begin{bmatrix} A_{11} & \mathbf{0} \\ \mathbf{0} & A_{22} \end{bmatrix}$$

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 A 为块对角矩阵, 用 $\lambda(A)$ 表示 A 的特征值集合, 显然, 它是 A_{11} 和 A_{22} 特征值集合 $\lambda(A_{11})$ 和 $\lambda(A_{22})$ 的并集, 也即

$$\lambda(A) = \lambda(A_{11}) \cup \lambda(A_{22})$$

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

$$\begin{bmatrix} A_{11} & \mathbf{0} \\ \mathbf{0} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} A_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & A_{22}^{-1} \end{bmatrix}$$

除了块对角矩阵, 还有分块三角矩阵。分块方阵 A , 如果 $A_{21} = \mathbf{0}$, 称之为分块上三角矩阵; 如果 $A_{12} = \mathbf{0}$, 称之为分块下三角矩阵。

若 A 为分块三角矩阵, 用 $\lambda(A)$ 表示 A 的特征值集合, 同样有:

$$\lambda(A) = \lambda(A_{11}) \cup \lambda(A_{22})$$

下面我们给出分块三角矩阵的逆和分块矩阵的逆。

命题 4.1.1. 非退化的分块三角矩阵的逆可以表示为:

$$\begin{bmatrix} A_{11} & \mathbf{0} \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} A_{11}^{-1} & \mathbf{0} \\ -A_{22}^{-1} A_{21} A_{11}^{-1} & A_{22}^{-1} \end{bmatrix}$$

$$\begin{bmatrix} A_{11} & A_{12} \\ \mathbf{0} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} A_{11}^{-1} & -A_{11}^{-1} A_{12} A_{22}^{-1} \\ \mathbf{0} & A_{22}^{-1} \end{bmatrix}$$

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解。

命题 4.1.2. 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

$$\begin{aligned} \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix}^{-1} &= \begin{pmatrix} \mathbf{S}_1^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{S}_2^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{S}_1^{-1} & \mathbf{S}_2^{-1} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{S}_1^{-1} & -\mathbf{S}_1^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{S}_2^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{S}_2^{-1} \end{pmatrix} \end{aligned}$$

引理 4.1.1. 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。

推论 4.1.1. *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:

$$(\mathbf{E} + \mathbf{F}\mathbf{G}\mathbf{H})^{-1} = \mathbf{E}^{-1} - \mathbf{E}^{-1}\mathbf{F}(\mathbf{G}^{-1} + \mathbf{H}\mathbf{E}^{-1}\mathbf{F})^{-1}\mathbf{H}\mathbf{E}^{-1}$$

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: Sherman-Morrison 公式。

推论 4.1.2. *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

$$\mathbf{A}_{11} = \mathbf{A}, \mathbf{A}_{12} = \mathbf{u}, \mathbf{A}_{22} = -1, \mathbf{A}_{21} = \mathbf{v}^T$$

则我们可以得到如下等式:

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}$$

这个式子让我们能够计算矩阵 \mathbf{A} 的秩 1 扰动的逆, 并且计算仅仅依赖于 \mathbf{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 \mathbf{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 $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{q} \in \mathbb{R}^m$, $\mathbf{p} \in \mathbb{R}^n$ 则有

$$|\text{rank}(\mathbf{A}) - \text{rank}(\mathbf{A} + \mathbf{q}\mathbf{p}^T)| \leq 1$$

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

现实世界的对象，除了用数值型向量表示数据之外，也可以用网络和图进行表达。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。

4.2.1 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 $\mathbb{X} = \{(1, 3), (1, 4), (2, 4), (3, 2), (2, 1), (3, 1)\}$ ，如图 4.3(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，如图 4.3(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 4.3(c)。从而将问题转化为研究图上顶点聚类。

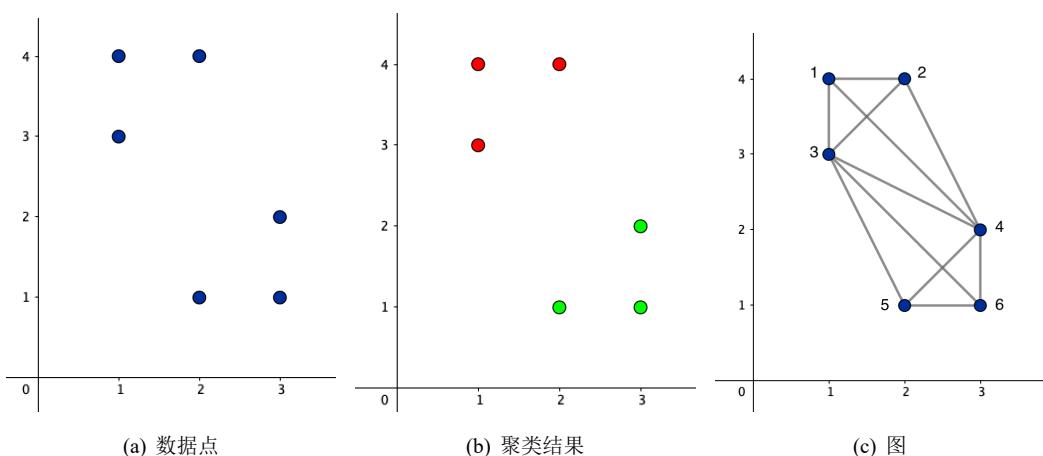


图 4.3: 谱聚类

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。
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尽可能的低，而子图内的边权重尽可能的高，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。而点与边之间，点与点之间可以通过关联矩阵，邻接矩阵，度矩阵和拉普拉斯矩阵来描述。

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。下面我们介绍图以及相关矩阵的概念。

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 $\mathbb{V} = \{v_1, v_2, \dots, v_n\}$ 和边集 \mathbb{E} 组成的，其中 \mathbb{E} 中的元素是一个二元对 $\{x, y\}$ ， $x, y \in \mathbb{V}$ 。

- 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 $\{v_i, v_j\}$ ， $1 \leq i, j, \text{len}$ 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，我们可以用顶点序列 v_0, v_1, \dots, v_n 来表示这条通路，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 ， $e_n = \{v_{n-1}, v_n\}$ 。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。

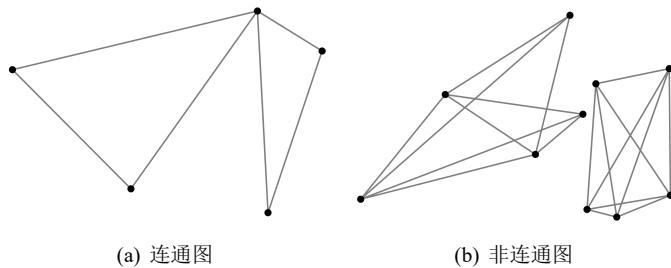


图 4.4: 连通图和非连通图

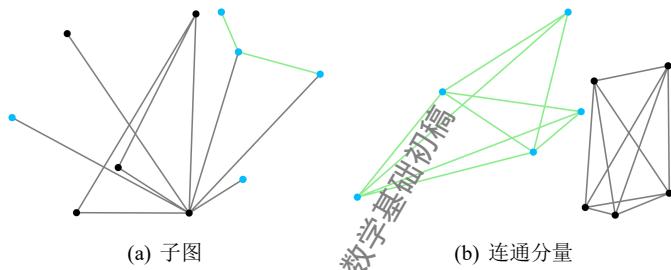


图 4.5: 子图

定理 4.2.1. 如果图 $G = (\mathbb{V}, \mathbb{E})$ 是连通图, 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。

有向图的矩阵

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 , 所有边的排列为 e_1, e_2, \dots, e_n , 其中 $m = |\mathbb{V}|$, $n = |\mathbb{E}|$, 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。

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 4.6 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：

$$\mathbf{B} = \begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & 0 & -1 & -1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

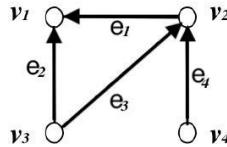


图 4.6: 一个有 4 个顶点的图

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. 图 4.3(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 4.2.1. 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：

- A 是对称矩阵, 即 $A = A^T$ 。
- 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 为图 G 节点的另一种排列, 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 , 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

$$\lambda_1 = -1.82842712$$

$$\lambda_2 = \lambda_3 = \lambda_4 = -1$$

$$\lambda_5 = 1$$

$$\lambda_6 = 3.82842712$$

所以图 4.3(c) 的特征值为 $\lambda_i, i = 1, \dots, 6$, 其谱为 λ_6

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. 图 4.3(c) 对应的拉普拉斯矩阵为

$$L = D - A = \begin{pmatrix} 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ -1 & -1 & 5 & -1 & -1 & -1 \\ -1 & -1 & -1 & 5 & -1 & -1 \\ 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & -1 & 3 \end{pmatrix}$$

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 。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。

性质 4.2.2. 设有向无权图 G 的关联矩阵为 B , 其对应的无向图的拉普拉斯矩阵为 L , 则 L 和 B 满足以下关系:

$$L = BB^T.$$

例 4.2.6. 图4.6对应的拉普拉斯矩阵为

$$\mathbf{L} = \mathbf{D} - \mathbf{A} = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

易验证, $\mathbf{L} = \mathbf{B}\mathbf{B}^T$ 。

加权图相关的矩阵

对于例4.2.1中的聚类例子, 目的是要把它聚成两类。从例4.2.5的拉普拉斯矩阵中可以发现, 第3个节点和第4个节点是对称的。也就是说, 如果仅仅根据这个无向图的拉普拉斯矩阵, 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 $G = \langle \mathbb{V}, \mathbb{E} \rangle$, 把图 G 的顶点排列成 $v_1, v_2, \dots, v_n, n = |\mathbb{V}|$, 将图 G 的邻接矩阵 $\mathbf{A} = (a_{ij})_{n \times n}$ 定义为

$$a_{ij} = \begin{cases} w_{ij} & \{v_i, v_j\} \in \mathbb{E} \\ 0 & \{v_i, v_j\} \notin \mathbb{E} \end{cases}$$

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。

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 $v_1, v_2, \dots, v_n, n = |\mathbb{V}|$, 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, 加权图的度矩阵 $\mathbf{D} = (d_{ij})_{n \times n}$ 定义为

$$d_{ij} = \begin{cases} d(v_i) & i = j \\ 0 & i \neq j \end{cases}$$

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 \mathbf{A} 和 \mathbf{D} , 我们称 $\mathbf{L} = \mathbf{D} - \mathbf{A}$ 为加权图的拉普拉斯矩阵。

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 \mathbf{A} 和 \mathbf{D} , 我们称矩阵

$$\tilde{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$$

为加权图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 \mathbf{L} 将变为 $k\mathbf{L}$ 。而正规化的拉普拉斯矩阵 $\tilde{\mathbf{L}}$ 则不会发生变化。这是因为

$$\tilde{\mathbf{L}}_{ij} = \frac{\mathbf{L}_{ij}}{\sqrt{d(v_i)d(v_j)}} = \frac{k\mathbf{L}_{ij}}{\sqrt{kd(v_i)kd(v_j)}}$$

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 4.2.7. 如果使用高斯核 ($\sigma = 1$) 来给例 4.2.1 中图 4.3(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. 设权重为正的加权图 $G = (\mathbb{V}, \mathbb{E})$ 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. 第 1 条性质是显然的。

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

$$\begin{aligned} \mathbf{x}^T \mathbf{L} \mathbf{x} &= \mathbf{x}^T \mathbf{D} \mathbf{x} - \mathbf{x}^T \mathbf{A} \mathbf{x} = \sum_{i=1}^n d_i x_i^2 - \sum_{i,j=1}^n w_{ij} x_i x_j \\ &= \frac{1}{2} \left(\sum_{i,j=1}^n w_{ij} x_i^2 - 2 \sum_{i,j=1}^n w_{ij} x_i x_j + \sum_{i,j=1}^n w_{ij} x_j^2 \right) \\ &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (x_i - x_j)^2 \\ &= \sum_{\{v_i, v_j\} \in \mathbb{E}} w_{ij} (x_i - x_j)^2 \end{aligned}$$

那么对于一个权重为正的加权图来说，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 $\mathbf{D}^{-\frac{1}{2}} \mathbf{1}$ 。 \square

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. 我们已经得到图 4.3(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

$$\mathbf{x} = (-0.442, -0.421, -0.358, 0.358, 0.421, 0.442)^T$$

我们可以得到前 3 个节点作为一类，后 3 个节点作为一类。

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。

反过来，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。

稀疏矩阵零元素分布常常是没有规律的。邻接矩阵经常是稀疏矩阵。

例 4.2.9. 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 4.7 对应的邻接矩阵。

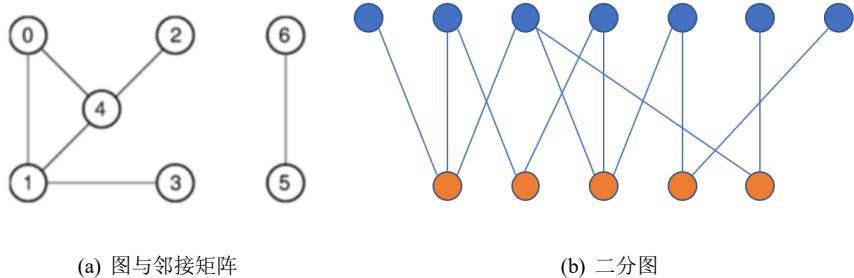


图 4.7: 对应矩阵为稀疏矩阵的图

例 4.2.10. 形如图 4.7 (b) 中的图是一种特殊图, 称为二分图, 他可以转换为如下矩阵, 是一个稀疏矩阵

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

二分图有丰富的应用场景, 比如

- 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: 一个多层次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。

字典学习

正如各种各样的知识都可以用一本字典里的字通过排列组合来表达一样，字典学习的目的就是将已有的(超)大规模的数据集进行压缩，找到蕴藏在这些数据点背后的最基本的原理。考虑一个 m 维空间中的数据集 $\{\mathbf{a}_i\}_{i=1}^n, \mathbf{a}_i \in \mathbb{R}^m$ ，假定每个 \mathbf{a}_i 都是由同一个字典生成的，且生成之后的数据带有噪声，因此字典学习的线性模型可以表示为 $\mathbf{a} = \mathbf{D}\mathbf{x} + \mathbf{e}$ ，这里 $\mathbf{D} \in \mathbb{R}^{m \times k}$ 是某个未知的字典，它的每一列 d_i 是字典的一个基向量； \mathbf{x} 是字典中基的系数，同样是未知的； \mathbf{e} 是某种噪声。字典学习模型中我们需要同时解出字典 \mathbf{D} 和系数 \mathbf{x} 。一般来说，数据的维数 m 和字典里基向量的数量 k 是远小于观测数 n 的，例如观测的数据是 10×10 的图像，则 $m = 100$ ，但采集的数据量 n 可以非常大(例如 $n \geq 100000$)。如果 $k < m$ ，我们称字典 \mathbf{D} 是不完备的；如果 $k > m$ ，我们称字典 \mathbf{D} 是超完备的； $k = m$ 对应的字典不能对表示带来任何的提高，因此实际中不予考虑。当 \mathbf{e} 是高斯白噪声时，可以定义损失函数

$$f(\mathbf{D}, \mathbf{X}) = \frac{1}{2n} \|\mathbf{D}\mathbf{X} - \mathbf{A}\|_F^2,$$

其中 $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n) \in \mathbb{R}^{m \times n}$ 为所有观测数据全体， $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \in \mathbb{R}^{k \times n}$ 是所有基系数全体。

在实际计算中，我们并不要求 \mathbf{D} 的列是正交的，因此一个样本点 \mathbf{a}_i 可能存在着多种不同的表示。这种冗余性给表示引入了稀疏性，这意味着字典 \mathbf{D} 是超完备的($k > m$)。稀疏性还可以帮助我们快速确定样本点是由哪几个基向量(而不是所有基向量)表示的，进而提高计算速度。具体地，在 \mathbf{e} 为高斯白噪声的条件下，我们定义稀疏编码损失函数

$$f(\mathbf{D}, \mathbf{X}) = \frac{1}{2n} \|\mathbf{D}\mathbf{X} - \mathbf{A}\|_F^2 + \lambda \|\mathbf{X}\|_1,$$

这里 λ 为正则化参数，其大小用来控制 \mathbf{X} 的稀疏度。我们注意到在 $f(\mathbf{D}, \mathbf{X})$ 中有乘积项 $\mathbf{D}\mathbf{X}$ ，显然 f 的极小值点处必有 $\|\mathbf{X}\|_1 \rightarrow 0$ 。因为假设 (\mathbf{D}, \mathbf{X}) 为问题的最小值点，那么 $f(c\mathbf{D}, \frac{1}{c}\mathbf{X}) < f(\mathbf{D}, \mathbf{X})$ ， $\forall c > 1$ 。因此，这里的保稀疏的正则项并没有意义。一个改进的做法是要求字典中的基向量模长不能太大，即 $\|\mathbf{D}\|_F \leq 1$ 。最终得到的优化问题为

$$\min_{\mathbf{D}, \mathbf{X}} \frac{1}{2n} \|\mathbf{D}\mathbf{X} - \mathbf{A}\|_F^2 + \lambda \|\mathbf{X}\|_1,$$

$$s.t. \|\mathbf{D}\|_F \leq 1$$

4.2.2 低秩矩阵

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 4.2.16. 设矩阵 $\mathbf{A} \in \mathbb{R}^{n \times m}$ ，如果矩阵 \mathbf{A} 的秩 $\text{rank}(\mathbf{A})$ 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 $\text{rank}(\mathbf{A})$ 和 $\max\{n, m\}$ 的关系，是因为很多时候，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。

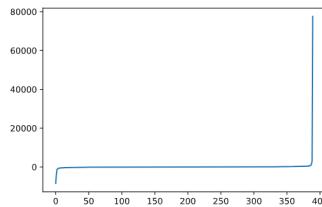
例 4.2.11. 考虑矩阵

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

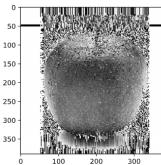
也是一个低秩矩阵。



(a) 图像



(b) 图像的特征值



(c) 新图像

图 4.8: 图像以及图像的特征值

例 4.2.12. 图 4.8(a) 是一个 390×390 的图像, 对应矩阵的特征值从小到大展示在图 4.8(b) 上。可以注意到很多特征值都集中在 0 附近。记图 4.8(a) 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 $\text{rank}(A') = 112$, 对应的图像为图 4.8(c)。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。

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。

$$\begin{aligned} & \min_{\mathbf{A}, \mathbf{E}} \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_1 \\ & \text{s.t. } \mathbf{D} = \mathbf{A} + \mathbf{E} \end{aligned}$$

其中 \mathbf{A} 代表了低秩结构信息， \mathbf{E} 是稀疏噪声。

现在考虑视频分割的问题。它是指把人们感兴趣的对象从视频场景中提取出来，例如分割出一段视频中的静止部分。视频的每一帧实际上是一个静态图片，虽然每幅图片中的静止对象可能受到光照变化、遮挡、平移、噪声等影响，造成不同图片之间有细微差别，但是不可否认的是它们彼此之间具有高度的相似性。如果把所有图片中的静止部分表示成一个矩阵，显然它们是相似的，并且由于静止对象具有一定的内部结构，由静止对象构成的矩阵一定是低秩的（各行或各列线性相关）。类似地，视频中的动态部分以及其他背景因素可以看作噪声。那么我们的任务就变成将视频含有的信息矩阵分解为含有内部结构的低秩矩阵和稀疏噪声矩阵之和。

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 \mathbf{D} 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

$$P_{\Omega}(\mathbf{D}_{ij}) = \begin{cases} D_{ij} & (i, j) \in \Omega \\ 0 & (i, j) \notin \Omega \end{cases}$$

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。

$$\begin{aligned} & \min_{\mathbf{Z}} \text{rank}(\mathbf{Z}) \\ & \text{s.t. } \mathbf{D} = \mathbf{BZ} \end{aligned}$$

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。从秩一分解的角度分析 $A = LU$, 可以将 A 写成若干个秩一矩阵和的形式:

$$A = l_1 u_1^T + l_2 u_2^T + \cdots + l_r u_r^T = \sum_{i=1}^r l_i u_i^T$$

其中, r 为矩阵 A 的秩。

若 A 是满秩, 也即 $r = n$, 则令

$$L = (l_1, l_2, \dots, l_n), U = (u_1^T, u_2^T, \dots, u_n^T)^T$$

那么就有

$$A = \sum_{i=1}^r l_i u_i^T = (l_1, l_2, \dots, l_n) \begin{pmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{pmatrix} = LU$$

如果我们进一步假设 l_i 和 u_i 的前 $i-1$ 个元素均为 0, 并且 l_i 的第 i 个元素为 1, 那么我们就得到了 A 的 LU 分解。

我们现在来考虑

$$l_1 u_1^T = A - (0, l_2, \dots, l_n) \begin{pmatrix} 0 \\ u_2^T \\ \vdots \\ u_n^T \end{pmatrix}.$$

我们知道 l_1 的第一个元素为 1, 所以 u_1 就是 $l_1 u_1^T$ 的第一行的行向量。另外一方面, 矩阵

$$(0, l_2, \dots, l_n) \begin{pmatrix} 0 \\ u_2^T \\ \vdots \\ u_n^T \end{pmatrix} = \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & * \end{pmatrix} \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & * \end{pmatrix} = \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & * \end{pmatrix}$$

的第一行和第一列均为 0, 即有

$$\begin{aligned}
 \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & * & * & \cdots & * \\ a_{31} & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & * & \cdots & \cdots & * \end{pmatrix} &= \begin{pmatrix} 1 \\ l_{21} \\ l_{31} \\ \vdots \\ l_{n1} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & * & \cdots & \cdots & * \end{pmatrix} \\
 &= \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ l_{21}u_{11} & * & * & \cdots & * \\ l_{31}u_{11} & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1}u_{11} & * & \cdots & \cdots & * \end{pmatrix}
 \end{aligned}$$

所以 \mathbf{u}_1 就是 \mathbf{A} 的第一行。而 \mathbf{l}_1 则是 \mathbf{A} 的第一列除以 u_{11} 也就是 a_{11} 得到的。

我们记 $\tilde{\mathbf{A}}^{(0)} = \mathbf{A}$, 当 $i \geq 1$ 时, 记

$$\tilde{\mathbf{A}}^{(i)} = \mathbf{A} - \sum_{j=1}^i \mathbf{l}_j \mathbf{u}_j^T$$

根据上面对于 \mathbf{u}_1 和 \mathbf{l}_1 的推导, 我们很容易将其应用到 \mathbf{u}_i 和 \mathbf{l}_i 上。也就是说, \mathbf{u}_i 就是 $\tilde{\mathbf{A}}^{(i-1)}$ 的第 i 行。而 \mathbf{l}_i 则是 $\tilde{\mathbf{A}}^{(i-1)}$ 的第 i 列除以 $\tilde{a}_{ii}^{(i-1)}$ 得到的。

例 4.3.1. 求矩阵 \mathbf{A}

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix}$$

的 LU 分解。

解. 记 $\tilde{\mathbf{A}}^{(0)} = \mathbf{A}$, 令 \mathbf{u}_1 是 \mathbf{A} 的第 1 行, \mathbf{l}_1 是 \mathbf{A} 的第 1 列除以 u_{11} 。则

$$\mathbf{l}_1 \mathbf{u}_1^T = \begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 8 & 12 \\ 7 & 14 & 21 \end{pmatrix}$$

$$\tilde{\mathbf{A}}^{(1)} = \mathbf{A} - \mathbf{l}_1 \mathbf{u}_1^T = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix} - \begin{pmatrix} 1 & 2 & 3 \\ 4 & 8 & 12 \\ 7 & 14 & 21 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{pmatrix}$$

容易看出 $\tilde{\mathbf{A}}^{(1)}$ 的第 2 行是 \mathbf{A} 的第 2 行减去其第一行的 4 倍, $\tilde{\mathbf{A}}^{(1)}$ 的第 3 行是 \mathbf{A} 的第 3 行减去其第一行的 7 倍。

$$\tilde{\mathbf{A}}^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{pmatrix}$$

令 \mathbf{u}_2 是 $\tilde{\mathbf{A}}^{(1)}$ 的第 2 行, \mathbf{l}_2 是 $\tilde{\mathbf{A}}^{(1)}$ 的第 2 列除以 u_{22} 。则

$$\mathbf{l}_2 \mathbf{u}_2^T = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \begin{pmatrix} 0 & -3 & -6 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -3 & -6 \\ 0 & -6 & -12 \end{pmatrix}$$

$$\tilde{\mathbf{A}}^{(2)} = \mathbf{A} - \mathbf{l}_1 \mathbf{u}_1^T - \mathbf{l}_2 \mathbf{u}_2^T = \tilde{\mathbf{A}}^{(1)} - \mathbf{l}_2 \mathbf{u}_2^T = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -3 & -6 \\ 0 & -6 & -12 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & -3 & -6 \\ 0 & -6 & -12 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

容易看出 $\tilde{\mathbf{A}}^{(2)}$ 的第 3 行就是 $\tilde{\mathbf{A}}^{(1)}$ 的第 3 行减去其第 2 行的 2 倍。

$$\tilde{\mathbf{A}}^{(2)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

令 \mathbf{u}_3 是 $\tilde{\mathbf{A}}^{(2)}$ 的第 3 行, \mathbf{l}_3 是 $\tilde{\mathbf{A}}^{(2)}$ 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

$$\mathbf{A} = \mathbf{L}\mathbf{U} = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{pmatrix}$$

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{L_1} [\] \xrightarrow{L_2} \cdots \xrightarrow{L_{k-1}} [\] \cdots \xrightarrow{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 $\mathbf{A} = \mathbf{L}\mathbf{U}$ 给出。

在上述步骤 1 中, \mathbf{L}_k 的一般形式可以表示为:

$$\mathbf{L}_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & & \ddots & \ddots & \\ & & & & -l_{n,k} & 1 \end{bmatrix} = \mathbf{I} - \mathbf{l}_k \mathbf{e}_k^T,$$

其中

$$\mathbf{l}_k = (0, \dots, 0, l_{k+1,k}, \dots, l_{n,k})^T.$$

我们把这种类型的初等下三角矩阵称作 **Gauss 变换**, 而称向量 \mathbf{l}_k 为 **Gauss 向量**。基于 Gauss 变换的 LU 分解计算方法也称为 **Gauss 消去法**。

Gauss 变换 \mathbf{L}_k 具有许多良好的性质:

- 对于一个给定的向量 $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, 我们有

$$\mathbf{L}_k \mathbf{x} = (x_1, \dots, x_k, x_{k+1} - x_k l_{k+1,k}, \dots, x_n - x_k l_{n,k})^T$$

由此立即可知, 只要取

$$l_{ik} = \frac{x_i}{x_k}, i = k+1, \dots, n$$

便有 $\mathbf{L}_k \mathbf{x} = (x_1, \dots, x_k, 0, \dots, 0)^T$, 这里我们要求 $x_k \neq 0$ 。

- Gauss 变换 \mathbf{L}_k 的逆易求解。因为 $\mathbf{e}_k^T \mathbf{l}_k = 0$, 所以

$$(\mathbf{I} - \mathbf{l}_k \mathbf{e}_k^T)(\mathbf{I} + \mathbf{l}_k \mathbf{e}_k^T) = \mathbf{I} - \mathbf{l}_k \mathbf{e}_k^T \mathbf{l}_k \mathbf{e}_k^T = \mathbf{I}$$

即

$$\mathbf{L}_k^{-1} = \mathbf{I} + \mathbf{l}_k \mathbf{e}_k^T$$

- Gauss 变换作用于矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 就相当于对该矩阵进行秩 1 修正, 也即

$$\mathbf{L}_k \mathbf{A} = (\mathbf{I} - \mathbf{l}_k \mathbf{e}_k^T) \mathbf{A} = \mathbf{A} - \mathbf{l}_k (\mathbf{e}_k^T \mathbf{A})$$

因此有

$$\begin{aligned} \mathbf{L} &= \mathbf{L}_1^{-1} \cdots \mathbf{L}_{n-1}^{-1} \\ &= (\mathbf{I} + \mathbf{l}_1 \mathbf{e}_1^T) (\mathbf{I} + \mathbf{l}_2 \mathbf{e}_2^T) \cdots (\mathbf{I} + \mathbf{l}_{n-1} \mathbf{e}_{n-1}^T) \\ &= \mathbf{I} + \mathbf{l}_1 \mathbf{e}_1^T + \cdots + \mathbf{l}_{n-1} \mathbf{e}_{n-1}^T \end{aligned}$$

即 L 有如下形式

$$L = I + (l_1, l_2, \dots, l_{n-1}, 0) = \begin{pmatrix} 1 & & & & & \\ l_{21} & 1 & & & & \\ l_{31} & l_{32} & 1 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix}$$

例 4.3.2. 求矩阵 A 的 LU 分解。

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix}$$

解. $A \rightarrow U$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \end{pmatrix} \xrightarrow{\substack{R_2 - (\frac{4}{1})R_1 \\ R_3 - (\frac{7}{1})R_1}} \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -11 \end{pmatrix} \xrightarrow{R_3 - (\frac{-6}{-3})R_2} \begin{pmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & 0 & 1 \end{pmatrix}$$

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

$$L = L_1^{-1} L_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{pmatrix}$$

根据利用秩一分解求解 LU 分解的方法，我们可以归纳得到算法 4.1。

根据以上算法，我们会得到唯一形式的 LU 分解。我们可以证明如下定理。

算法 4.1 LU 分解

```

1:  $L = I, U = O$ 
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  $A(k + 1 : n, k + 1 : n)$ 
12:    end for
13:   end for
14: end for

```

定理 4.3.1. [LU 分解的唯一性] 如果 $A \in \mathbb{R}^{n \times n}$ 非奇异，并且其 LU 分解存在，则 A 的 LU 分解是唯一的，且 $\det(A) = u_{11}u_{22} \cdots u_{nn}$ 。

证明. 令 $A = L_1U_1$ 和 $A = L_2U_2$ 是非奇异矩阵 A 的两个 LU 分解，则 $L_1U_1 = L_2U_2$ 。

由于 $L_2^{-1}L_1$ 是下三角矩阵，并且 $U_2U_1^{-1}$ 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， $L_1 = L_2, U_1 = U_2$ ，即 LU 分解是唯一的。

若 $A = LU$ ，则 $\det(A) = \det(LU) = \det(L)\det(U) = \det(U) = u_{11}u_{22} \cdots u_{nn}$ 。 \square

4.3.2 选主元的 LU 分解

然而，LU 分解并不一定总是存在的。我们来看一个例子。

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。

定理 4.3.2. 矩阵 $A \in \mathbb{R}^{n \times n}$ 能够进行 LU 分解的充分必要条件是 A 的前 $n-1$ 个主元均不为 0。

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 4.3.3. 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. 在例 4.3.3 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。

$$A \rightarrow U$$

$$\begin{pmatrix} 1 & 2 & 0 \\ 1 & 2 & 1 \\ 0 & 2 & 0 \end{pmatrix} \xrightarrow{R_2 - (\frac{1}{1})R_1} \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 2 & 0 \end{pmatrix} \xrightarrow{R_2 \leftrightarrow R_3} \begin{pmatrix} 1 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

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。

所以为了避免在 LU 分解过程中主元为零，在每次对 $A^{(i-1)}$ 做初等变换 L_i 前需要判断主元是否为零。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

$$L_n P_n L_{n-1} P_{n-1} \cdots L_2 P_2 L_1 P_1 A = U.$$

这样我们得到了 A 的分解：

$$A = (L_n P_n L_{n-1} P_{n-1} \cdots L_2 P_2 L_1 P_1)^{-1} U.$$

虽然 $(L_n P_n L_{n-1} P_{n-1} \cdots L_2 P_2 L_1 P_1)^{-1}$ 不是一个下三角矩阵，但是如果我们将 A 按如下方式重新排列各行，有

$$P_n P_{n-1} \cdots P_2 P_1 A = P_n P_{n-1} \cdots P_2 P_1 (L_n P_n L_{n-1} P_{n-1} \cdots L_2 P_2 L_1 P_1)^{-1} U$$

可以证明

$$P_n P_{n-1} \cdots P_2 P_1 (L_n P_n L_{n-1} P_{n-1} \cdots L_2 P_2 L_1 P_1)^{-1}$$

是一个下三角矩阵。

上述过程可归纳为算法4.2。

算法 4.2 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

矩阵的 QR 分解也称正交三角分解，是一种特殊的三角分解。QR 分解在解决最小二乘问题、矩阵特征值的计算等问题中起到重要作用，也是目前计算一般矩阵的全部特征值和特征向量的最有效方法之一。矩阵 A 的 QR 分解可以通过 Gram-Schmidt 正交化、Householder 变换和 Givens 变换等方法实现。

定义 4.4.1. 设矩阵 $A \in \mathbb{R}^{m \times n}$ ($m \geq n$)，如果存在 m 阶正交矩阵 Q 和 n 阶上三角矩阵 R ，使得

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix},$$

则称之为 A 的 QR 分解或正交三角分解。

在上述定义中，当 $A \in \mathbb{C}^{m \times n}$ ($m \geq n$) 且 Q 为 m 阶酉矩阵，则称之为 A 的酉三角分解。

4.4.1 基于 Gram-Schmidt 正交化的 QR 分解

定理 4.4.1. 对任意一个列满秩的实矩阵 $A \in \mathbb{R}^{m \times n}$ ($m \geq n$)，都存在正交三角分解

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix},$$

其中 Q 为 m 阶正交矩阵， R 具有正的对角元的上三角矩阵；而且当 $m = n$ 且 A 非奇异时，上述分解还是唯一的。

上述定理对于复矩阵也成立，此时 Q 为酉矩阵。

证明. 设 A 是一个列满秩的实矩阵， A 的 n 个列向量为 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ ，由于 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ 线性无关，将它们用 Schmidt 正交化方法得标准正交向量 $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$ 即

$$\begin{cases} \mathbf{a}_1 = r_{11} \mathbf{q}_1 \\ \mathbf{a}_2 = r_{12} \mathbf{q}_1 + r_{22} \mathbf{q}_2 \\ \dots \\ \mathbf{a}_n = r_{1n} \mathbf{q}_1 + r_{2n} \mathbf{q}_2 + \dots + r_{nn} \mathbf{q}_n \end{cases}$$

其中 $r_{ii} > 0, i = 1, 2, \dots, n$ ，从而有

$$(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n) = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n) \begin{pmatrix} r_{11} & r_{12} & r_{13} & \dots & r_{1n} \\ 0 & r_{22} & r_{23} & \dots & r_{2n} \\ 0 & 0 & r_{33} & \dots & r_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & r_{nn} \end{pmatrix}$$

如果给 $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$ 补上 $m - n$ 个标准正交的向量 $\mathbf{q}_{n+1}, \mathbf{q}_{n+2}, \dots, \mathbf{q}_m$ 就有

$$(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n) = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m) \begin{pmatrix} r_{11} & r_{12} & r_{13} & \dots & r_{1n} \\ 0 & r_{22} & r_{23} & \dots & r_{2n} \\ 0 & 0 & r_{33} & \dots & r_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & r_{nn} \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

令 $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m)$, $\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & \dots & r_{1n} \\ 0 & r_{22} & r_{23} & \dots & r_{2n} \\ 0 & 0 & r_{33} & \dots & r_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & r_{nn} \end{pmatrix}$, 则 $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$

再证唯一性。

如果

$$\mathbf{A} = \mathbf{Q}\mathbf{R} = \mathbf{Q}_1\mathbf{R}_1,$$

由此得 $\mathbf{Q} = \mathbf{Q}_1\mathbf{R}_1\mathbf{R}^{-1}$, 令 $\mathbf{D} = \mathbf{R}_1\mathbf{R}^{-1}$, 那么 \mathbf{D} 仍为具有正对角元的上三角矩阵。

由于

$$\mathbf{I} = \mathbf{Q}^T \mathbf{Q} = (\mathbf{Q}_1 \mathbf{D})^T (\mathbf{Q}_1 \mathbf{D}) = \mathbf{D}^T \mathbf{D}$$

即 \mathbf{D} 为正交矩阵, 因此 \mathbf{D} 为单位矩阵 (正交上三角矩阵为对角矩阵)

故

$$\mathbf{Q} = \mathbf{Q}_1 \mathbf{D} = \mathbf{Q}_1, \mathbf{R}_1 = \mathbf{D} \mathbf{R} = \mathbf{R}$$

□

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:

$$\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R}_{n \times n} \\ \mathbf{0}_{m-n} \end{pmatrix}$$

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 $A^T A = (QR)^T (QR) = R^T R$, 因此可以得出结论: $G = R^T$ 是 $A^T A$ 的下三角 Cholesky 因子。由于这个原因, 在关于估计的文献中, 矩阵 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。

引理 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

$$Q A = B \quad (4.3)$$

下面我们通过一个例子来说明, 如何通过 Gram-Schmidt 正交化方法求得矩阵的 QR 分解。

例 4.4.1. 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 $a_1 = (0, 1, 1)^T$, $a_2 = (1, 1, 0)^T$, $a_3 = (1, 0, 1)^T$, 由 Gram-Schmidt 正交化方法。先正交化得

$$\left\{ \begin{array}{l} b_1 = a_1 = (0, 1, 1)^T \\ b_2 = a_2 - \frac{\langle a_2, b_1 \rangle}{\langle b_1, b_1 \rangle} b_1 = (1, \frac{1}{2}, -\frac{1}{2})^T \\ b_3 = a_3 - \frac{\langle a_3, b_1 \rangle}{\langle b_1, b_1 \rangle} b_1 - \frac{\langle a_3, b_2 \rangle}{\langle b_2, b_2 \rangle} b_2 = (\frac{2}{3}, -\frac{2}{3}, \frac{2}{3})^T \end{array} \right.$$

然后单位化

$$\left\{ \begin{array}{l} q_1 = \frac{1}{\sqrt{2}}(0, 1, 1)^T \\ q_2 = \frac{1}{\sqrt{6}}(2, 1, -1)^T \\ q_3 = \frac{1}{\sqrt{3}}(1, -1, 1)^T \end{array} \right.$$

整理得

$$\left\{ \begin{array}{l} a_1 = |b_1|q_1 \\ a_2 = \langle a_2, q_1 \rangle q_1 + |b_2|q_2 \\ a_3 = \langle a_3, q_1 \rangle q_1 + \langle a_3, q_2 \rangle q_2 + |b_3|q_3 \end{array} \right.$$

于是

$$Q = (q_1, q_2, q_3) = \begin{bmatrix} 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} |\mathbf{b}_1| & \langle \mathbf{a}_2, \mathbf{q}_1 \rangle & \langle \mathbf{a}_3, \mathbf{q}_1 \rangle \\ 0 & |\mathbf{b}_2| & \langle \mathbf{a}_3, \mathbf{q}_2 \rangle \\ 0 & 0 & |\mathbf{b}_3| \end{bmatrix} = \begin{bmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{\sqrt{6}}{2} & \frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{2}{\sqrt{3}} \end{bmatrix}$$

那么 $\mathbf{A} = \mathbf{QR}$ 即为所求表达式。

在实际数值计算中, Gram-Schmidt 正交化是数值不稳定的, 计算中累积的舍入误差会使最终结果的正交性变得很差。因此常用一种修正的 Gram-Schmidt 正交化方法, 它是对经典 Gram-Schmidt 正交化法的修正, 使上三角矩阵 \mathbf{R} 的元素不是按列, 而是按行计算, 这时舍入误差将变小。

4.4.2 基于 Householder 变换的 QR 分解

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 Householder 变换 \mathbf{H} 满足

$$\mathbf{Hx} = \alpha \mathbf{e}_1 \quad (4.4)$$

其中 $\alpha = \pm \|\mathbf{x}\|_2$ 。

证明. 由于

$$\begin{aligned} \mathbf{Hx} &= (\mathbf{I} - 2\mathbf{w}\mathbf{w}^T)\mathbf{x} = \mathbf{x} - 2(\mathbf{w}^T\mathbf{x})\mathbf{w} \\ 2(\mathbf{w}^T\mathbf{x})\mathbf{w} &= \mathbf{x} - \mathbf{Hx} \end{aligned}$$

因此 \mathbf{w} 为与 $\mathbf{x} - \mathbf{Hx}$ 同方向的单位向量, 故欲使 $\mathbf{Hx} = \alpha \mathbf{e}_1$, 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

$$\|\mathbf{x}\|_2 = \|\mathbf{Hx}\|_2 = \|\alpha \mathbf{e}_1\|_2 = |\alpha| \cdot \|\mathbf{e}_1\|_2 = |\alpha|$$

即 $\alpha = \pm \|\mathbf{x}\|_2$ 。容易验证, 如上选取的 \mathbf{H} 确实满足式(4.4)。 \square

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 $\mathbf{w} = \mathbf{v} / \|\mathbf{v}\|_2$ 。

此外, 在实际计算中, α 取正还是取负根据具体情况来决定。

例 4.4.2. 用 Householder 变换将向量 $\mathbf{x} = (0, 3, 4)^T$ 化为与 $\mathbf{e} = (1, 0, 0)^T$ 平行的向量。

解. 由于 $\|x\|_2 = 5$, 不妨取 $\alpha = \|x\|_2 = 5$ 。令

$$\omega = \frac{x - \alpha e}{\|x - \alpha e\|_2} = \frac{1}{5\sqrt{2}} \begin{pmatrix} -5 \\ 3 \\ 4 \end{pmatrix},$$

则

$$H = I - 2\omega\omega^T = \frac{1}{25} \begin{pmatrix} 0 & 15 & 20 \\ 15 & 16 & -12 \\ 20 & -12 & 91 \end{pmatrix}$$

因此 $Hx = 5e$ 。

利用 Householder 变换求矩阵的 QR 分解的步骤:

[1] 将矩阵 $A = (\alpha_1, \alpha_2, \dots, \alpha_n)$, 取 $\omega_1 = \frac{\alpha_1 - a_1 e}{\|\alpha_1 - a_1 e\|_2}$, $a_1 = \|\alpha_1\|_2$ 则

$$H_1 = I - 2\omega_1\omega_1^T$$

那么

$$H_1 A = (H_1 \alpha_1, H_1 \alpha_2, \dots, H_1 \alpha_n) = \begin{pmatrix} a_1 & * & \dots & * \\ 0 & & & \\ \vdots & & & \mathbf{B}_1 \\ 0 & & & \end{pmatrix}$$

[2] 将矩阵 \mathbf{B}_1 按列分块, $\mathbf{B}_1 = (\beta_2, \beta_3, \dots, \beta_n)$ 取 $\omega_2 = \frac{\beta_2 - b_2 e}{\|\beta_2 - b_2 e\|_2}$, $b_2 = \|\beta_2\|_2$ 则

$$\tilde{H}_2 = I - 2\omega_2\omega_2^T$$

并且令

$$H_2 = \begin{pmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \tilde{H}_2 \end{pmatrix}$$

故有

$$H_2(H_1 A) = \begin{pmatrix} a_1 & * & * & \dots & * \\ 0 & a_2 & * & \dots & * \\ 0 & 0 & & & \\ \vdots & \vdots & & & \mathbf{C}_1 \\ 0 & 0 & & & \end{pmatrix}$$

依次进行下去, 得到第 $n - 1$ 个 n 阶的 Householder 矩阵 H_{n-1} , 使得

$$\mathbf{H}_{n-1} \dots \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} = \begin{pmatrix} a_1 & * & \dots & * \\ & a_2 & \dots & * \\ & & \ddots & \vdots \\ & & & a_n \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{R} \\ \mathbf{O} \end{pmatrix}$$

因 \mathbf{H}_i 是自逆矩阵, 令 $\mathbf{Q} = \mathbf{H}_1 \mathbf{H}_2 \dots \mathbf{H}_{n-1}$, 则 $\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R} \\ \mathbf{O} \end{pmatrix}$ 。

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 $a_1 = \|\alpha_1\|_2 = 2$, $\mathbf{w}_1 = \frac{\alpha_1 - a_1 \mathbf{e}_1}{\|\alpha_1 - a_1 \mathbf{e}_1\|_2} = \frac{1}{\sqrt{2}}(-1, 0, 1)^T$, 则

$$\mathbf{H}_1 = \mathbf{I} - 2\mathbf{w}_1 \mathbf{w}_1^H = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

从而

$$\mathbf{H}_1 \mathbf{A} = \begin{pmatrix} 2 & 1 & 2 \\ 0 & 4 & -2 \\ 0 & 3 & 1 \end{pmatrix}$$

记 $\beta = (4, 3)^T$, 则 $b_2 = \|\beta\|_2 = 5$, 令 $\mathbf{w}_2 = \frac{\beta - b_2 \mathbf{e}_1}{\|\beta - b_2 \mathbf{e}_1\|_2} = \frac{1}{\sqrt{10}}(-1, 3)^T$

$$\tilde{\mathbf{H}}_2 = \mathbf{I} - 2\mathbf{w}_2 \mathbf{w}_2^H = \begin{pmatrix} \frac{4}{5} & \frac{3}{5} \\ \frac{3}{5} & -\frac{4}{5} \end{pmatrix}$$

记

$$\mathbf{H}_2 = \begin{pmatrix} 1 & \mathbf{0}^T \\ 0 & \tilde{\mathbf{H}}_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & \frac{3}{5} \\ 0 & \frac{3}{5} & -\frac{4}{5} \end{pmatrix}$$

则

$$\mathbf{H}_2(\mathbf{H}_1 \mathbf{A}) = \begin{pmatrix} 2 & 1 & 2 \\ 0 & 5 & -1 \\ 0 & 0 & -2 \end{pmatrix} = \mathbf{R}$$

取

$$\mathbf{Q} = \mathbf{H}_1 \mathbf{H}_2 = \begin{pmatrix} 0 & \frac{3}{5} & -\frac{4}{5} \\ 0 & \frac{4}{5} & \frac{3}{5} \\ 1 & 0 & 0 \end{pmatrix}$$

则 $\mathbf{A} = \mathbf{QR}$ 。

4.4.3 基于 Givens 变换的 QR 分解

定理 4.4.3. 对于任意向量 $\mathbf{x} \in \mathbb{R}^n$, 存在 Givens 变换 \mathbf{T}_{kl} 使得 $\mathbf{T}_{kl}\mathbf{x}$ 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 $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, $\mathbf{T}_{kl}\mathbf{x} = (y_1, y_2, \dots, y_n)^T$

由 Givens 矩阵的定义可得

$$\begin{cases} y_k &= cx_k + sy_l \\ y_l &= -sx_k + cx_l \\ y_j &= x_j, (j \neq k, l) \end{cases}$$

(i) 当 $|x_k|^2 + |x_l|^2 = 0$ 时, 取 $c = 1, s = 0$, 则 $\mathbf{T}_{kl} = \mathbf{I}$, 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

$$c = \frac{x_k}{\sqrt{|x_k|^2 + |x_l|^2}}, s = \frac{x_l}{\sqrt{|x_k|^2 + |x_l|^2}},$$

则

$$\begin{cases} y_k &= \frac{x_k^2}{\sqrt{|x_k|^2 + |x_l|^2}} + \frac{x_l^2}{\sqrt{|x_k|^2 + |x_l|^2}} = \sqrt{|x_k|^2 + |x_l|^2} > 0 \\ y_l &= -\frac{x_k x_l}{\sqrt{|x_k|^2 + |x_l|^2}} + \frac{x_l x_k}{\sqrt{|x_k|^2 + |x_l|^2}} = 0 \\ y_j &= x_j, (j \neq k, l) \end{cases}$$

结论成立。 □

推论 4.4.1. 给定一个向量 $\mathbf{x} \in \mathbb{R}^n$, 则存在一组 Givens 矩阵 $\mathbf{T}_{12}, \mathbf{T}_{13}, \dots, \mathbf{T}_{1n}$, 使得

$$\mathbf{T}_{1n} \dots \mathbf{T}_{13} \mathbf{T}_{12} \mathbf{x} = \|\mathbf{x}\|_2 \mathbf{e}_1,$$

称为用 Givens 变换化向量 $\mathbf{x} \in \mathbb{R}^n$ 与第一自然基向量 \mathbf{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 $\mathbf{x} = (1, 2, 2)^T$ 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
 $\mathbf{T}_{12} = \begin{pmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} & 0 \\ -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} & 0 \\ 0 & 0 & 1 \end{pmatrix}$ 故有 $\mathbf{T}_{12}\mathbf{x} = \begin{pmatrix} \sqrt{5} \\ 0 \\ 2 \end{pmatrix}$ 对于 $\mathbf{T}_{12}\mathbf{x}$ 取 $c_2 = \frac{\sqrt{5}}{3}, s_2 = \frac{2}{3}$ 则
 $\mathbf{T}_{13} = \begin{pmatrix} \frac{\sqrt{5}}{3} & 0 & \frac{2}{3} \\ 0 & 1 & 0 \\ -\frac{2}{3} & 0 & \frac{\sqrt{5}}{3} \end{pmatrix}, \mathbf{T}_{13}\mathbf{T}_{12}\mathbf{x} = 3\mathbf{e}_1$

利用 Givens 变换求矩阵 QR 分解的步骤:

先将矩阵 \mathbf{A} 按列分块,

$$\mathbf{A}^* = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_n)$$

[1] 对于 $\boldsymbol{\alpha}_1$ 存在一组 Givens 矩阵 $\mathbf{T}_{12}, \mathbf{T}_{13}, \dots, \mathbf{T}_{1n}$ 使得

$$\mathbf{T}_{1n} \dots \mathbf{T}_{13}\mathbf{T}_{12}\boldsymbol{\alpha}_1 = \|\boldsymbol{\alpha}_1\|_2 \mathbf{e}_1$$

于是

$$\mathbf{T}_{1n} \dots \mathbf{T}_{13}\mathbf{T}_{12}\mathbf{A} = \begin{pmatrix} a_1 & * \\ 0 & \mathbf{B}_1 \end{pmatrix}, a_1 = \|\boldsymbol{\alpha}_1\|_2$$

[2] 将矩阵 $\begin{pmatrix} * \\ \mathbf{B}_1 \end{pmatrix}$ 按列分块

$$\begin{pmatrix} * \\ \mathbf{B}_1 \end{pmatrix} = \begin{pmatrix} * & * & \dots & * \\ \beta_2 & \beta_3 & \dots & \beta_n \end{pmatrix}$$

又存在一组 Givens 矩阵 $\mathbf{T}_{23}, \mathbf{T}_{24}, \dots, \mathbf{T}_{2n}$ 使得

$$\mathbf{T}_{2n} \dots \mathbf{T}_{24}\mathbf{T}_{23} \begin{pmatrix} * \\ \beta_2 \end{pmatrix} = (*, b_2, 0, \dots, 0)^T, b_2 = \|\beta_2\|_2$$

因此

$$\mathbf{T}_{2n} \dots \mathbf{T}_{24} \mathbf{T}_{23} \mathbf{T}_{1n} \dots \mathbf{T}_{13} \mathbf{T}_{12} \mathbf{A} = \begin{pmatrix} a_1 & * & * & \dots & * \\ 0 & b_1 & * & \dots & * \\ 0 & 0 & \mathbf{C}_2 & & \end{pmatrix}$$

依次进行下去得到

$$\mathbf{T}_{n-1,n} \dots \mathbf{T}_{2n} \dots \mathbf{T}_{23} \mathbf{T}_{1n} \dots \mathbf{T}_{12} \mathbf{A} = \begin{pmatrix} a_1 & * & \dots & * \\ a_2 & \dots & & * \\ \ddots & & & \vdots \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{R} \\ \mathbf{O} \end{pmatrix}$$

[3] 令 $\mathbf{Q} = \mathbf{T}_{12}^T \dots \mathbf{T}_{1n}^T \mathbf{T}_{23}^T \dots \mathbf{T}_{2n}^T \dots \mathbf{T}_{n-1,n}^T$, 则 $\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R} \\ \mathbf{O} \end{pmatrix}$ 。

利用 Givens 变换进行 QR 分解, 需要作 $\frac{n(n-1)}{2}$ 个初等旋转矩阵的连乘积, 当 n 较大时, 计算量较大, 因此常用镜像变换来进行 QR 分解。

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 $\mathbf{G}_{(1,2)}^{(1)} = \begin{pmatrix} 0 & 1 & \\ -1 & 0 & \\ & & 1 \end{pmatrix}$

$$\mathbf{A}^{(2)} = \mathbf{G}_{(1,2)}^{(1)} \mathbf{A}^{(1)} = \begin{pmatrix} 2 & 1 & 1 \\ 0 & -3 & -1 \\ 0 & -4 & 2 \end{pmatrix}$$

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

$$\mathbf{A}^{(3)} = \mathbf{G}_{(2,3)}^{(2)} \mathbf{A}^{(2)} = \begin{pmatrix} 2 & 1 & 1 \\ 0 & 5 & -1 \\ 0 & 0 & -2 \end{pmatrix} = \mathbf{R}$$

易得

$$\mathbf{Q} = \begin{pmatrix} 0 & \frac{3}{5} & -\frac{4}{5} \\ 0 & \frac{4}{5} & \frac{3}{5} \\ 1 & 0 & 0 \end{pmatrix}$$

即得到 QR 分解: $\mathbf{A} = \mathbf{Q}\mathbf{R}$ 。

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 谱分解

定理 4.5.1. (矩阵的特征分解定理) 一个矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 可以分解为 $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$, 其中 \mathbf{P} 是由特征向量构成的可逆矩阵, \mathbf{D} 是对角矩阵且对角元是 \mathbf{A} 的特征值, 当且仅当 \mathbf{A} 有 n 个线性无关的特征向量。

性质 4.5.1. (关于对称矩阵特征值特征向量的有用性质)

- 对称矩阵总是具有实特征值。
- 对称矩阵的不同特征值对应的特征向量是相互正交的。

定理 4.5.2. 设 \mathbf{A} 是实对称矩阵, 则 \mathbf{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 $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 满足 $A\mathbf{x} = \lambda_0\mathbf{x}$ 令 $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ 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

$$\bar{\mathbf{x}}^T(\bar{A}\bar{\mathbf{x}}) = \bar{\mathbf{x}}^T A^T \mathbf{x} = (\bar{A}\bar{\mathbf{x}})^T \mathbf{x} = (\bar{A}\bar{\mathbf{x}})^T \bar{\mathbf{x}}$$

其左边为 $\lambda_0 \bar{\mathbf{x}}^T \mathbf{x}$, 右边为 $\bar{\lambda}_0 \bar{\mathbf{x}}^T \bar{\mathbf{x}}$ 。故

$$\lambda_0 \bar{\mathbf{x}}^T \mathbf{x} = \bar{\lambda}_0 \bar{\mathbf{x}}^T \bar{\mathbf{x}}$$

又因 \mathbf{x} 是非零向量

$$\bar{\mathbf{x}}^T \mathbf{x} = \bar{x}_1 x_1 + \bar{x}_2 x_2 + \dots + \bar{x}_n x_n \neq 0$$

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。

定理 4.5.3. (谱分解定理) 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 是对角矩阵。
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

$$A\mathbf{Q} = A(\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n) = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n) \begin{pmatrix} \lambda_1 & \mathbf{a}^T \\ \mathbf{0} & \mathbf{C} \end{pmatrix} = \mathbf{Q}\Lambda_1$$

故 $A = \mathbf{Q}\Lambda_1\mathbf{Q}^T$, $A_1 = \mathbf{Q}^T A \mathbf{Q}$ 。 A 对称, 所以 A_1 对称, 从而 $\mathbf{a}^T = \mathbf{0}^T$, \mathbf{C} 对称。

根据假设, 存在正交矩阵 \mathbf{Q}_1 使得 $\mathbf{Q}_1^T \mathbf{C} \mathbf{Q}_1 = \Lambda_1$, $\mathbf{C} = \mathbf{Q}_1 \Lambda_1 \mathbf{Q}_1^T$,

$$A_1 = \begin{pmatrix} \lambda_1 & \\ & \mathbf{C} \end{pmatrix} = \begin{pmatrix} 1 & \\ & \mathbf{Q}_1 \end{pmatrix} \begin{pmatrix} \lambda_1 & \\ & \Lambda_1 \end{pmatrix} \begin{pmatrix} 1 & \\ & \mathbf{Q}_1^T \end{pmatrix}$$

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 $A = \mathbf{Q}_2 \Lambda \mathbf{Q}_2^T$, $\mathbf{Q}_2^T A \mathbf{Q}_2 = \Lambda$ 。

2 \rightarrow 3: 将 Q 按列分块 $Q = (q_1, q_2, \dots, q_n)$, 记 $\Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$

$$A\mathbf{Q} = (Aq_1, Aq_2, \dots, Aq_n) = \mathbf{Q}\Lambda = (\lambda_1 q_1, \lambda_2 q_2, \dots, \lambda_n q_n)$$

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 $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n)$

是正交矩阵。记 $\mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix}$ 其中 λ_i 是 \mathbf{q}_i 对应的特征值。

$$\mathbf{A}\mathbf{Q} = (A\mathbf{q}_1, A\mathbf{q}_2, \dots, A\mathbf{q}_n) = (\lambda_1\mathbf{q}_1, \lambda_2\mathbf{q}_2, \dots, \lambda_n\mathbf{q}_n) = \mathbf{Q}\mathbf{\Lambda}$$

$$\text{故 } \mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \mathbf{A}^T = \mathbf{Q}\mathbf{\Lambda}^T\mathbf{Q}^T = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T = \mathbf{A}.$$

□

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 $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, 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, $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ 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 $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ 改写成秩一矩阵和的形式:

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T = \sum_{i=1}^n \lambda_i \mathbf{q}_i \mathbf{q}_i^T$$

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。

(2) 写出左特征向量方阵 \mathbf{Q} 和特征值方阵 $\mathbf{\Lambda}$ 。

(3) 写出其秩一矩阵和的形式。

$$\text{解. (1) 由 } |\lambda\mathbf{I} - \mathbf{A}| = \begin{vmatrix} \lambda - 2 & -1 \\ -1 & \lambda - 2 \end{vmatrix} = 0$$

得到特征值 $\lambda_1 = 3, \lambda_2 = 1$ 。得到对应的特征向量将其单位化有

$$\mathbf{q}_1 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)^T, \mathbf{q}_2 = \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)^T$$

(2).

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2] = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, \quad \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & \\ & \lambda_2 \end{bmatrix} = \begin{bmatrix} 3 & \\ & 1 \end{bmatrix}$$

$$\text{又因为 } \mathbf{A} \text{ 是实对称矩阵, 所以 } \mathbf{A} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 3 & \\ & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

(3).

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

从线性空间的角度看, 在一个定义了内积的线性空间里, 对一个 N 阶对称方阵进行特征分解, 就产生了该空间的 N 个标准正交基, 矩阵对应的变换将空间中的向量投影到这 N 个基上。 N 个特征向量就是 N 个标准正交基, 而特征值的模则代表向量在每个基上的投影长度的伸缩倍数。

定义 4.5.2. 设矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 为对称矩阵, $R(\mathbf{x}) = \frac{\mathbf{x}^T \mathbf{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$, ($0 \neq \mathbf{x} \in \mathbb{R}^n$) 被称为瑞利商。

性质 4.5.2. 给定一个对称矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$, 瑞利商 $R(\mathbf{x})$ 有如下性质:

$$\lambda_{\min}(\mathbf{A}) \leq R(\mathbf{x}) \leq \lambda_{\max}(\mathbf{A})$$

并且有

$$\lambda_{\max}(\mathbf{A}) = \max_{\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{A} \mathbf{x}, \quad \lambda_{\min}(\mathbf{A}) = \min_{\|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{A} \mathbf{x},$$

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 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 为一对称矩阵, 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

$$\mathbf{x} = a_1 \mathbf{q}_1 + a_2 \mathbf{q}_2 + \dots + a_n \mathbf{q}_n$$

则瑞利商分母为

$$\mathbf{x}^T \mathbf{x} = \left(\sum_{i=0}^n a_i \mathbf{q}_i \right)^T \left(\sum_{j=0}^n a_j \mathbf{q}_j \right) = \sum_{i=0}^n a_i^2 \mathbf{q}_i^T \mathbf{q}_i = \sum_{i=0}^n a_i^2$$

而瑞利商分子为

$$\begin{aligned} \mathbf{x}^T \mathbf{A} \mathbf{x} &= \left(\sum_{i=0}^n a_i \mathbf{q}_i \right)^T \mathbf{A} \left(\sum_{j=0}^n a_j \mathbf{q}_j \right) = \sum_{i=0}^n \sum_{j=0}^n a_i \mathbf{q}_i^T \mathbf{A} a_j \mathbf{q}_j \\ &= \sum_{i=0}^n \sum_{j=0}^n a_i a_j \lambda_j \mathbf{q}_i^T \mathbf{q}_j = \sum_{i=0}^n a_i^2 \lambda_i \mathbf{q}_i^T \mathbf{q}_i = \sum_{i=0}^n a_i^2 \lambda_i \end{aligned}$$

$$\mathbf{x}^T \mathbf{x} = \sum_{i=0}^n a_i^2, \quad \mathbf{x}^T \mathbf{A} \mathbf{x} = \sum_{i=0}^n a_i^2 \lambda_i$$

又 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, 所以 $\lambda_n \mathbf{x}^T \mathbf{x} \leq \mathbf{x}^T \mathbf{A} \mathbf{x} \leq \lambda_1 \mathbf{x}^T \mathbf{x}$,

$$\lambda_{\min}(\mathbf{A}) \leq R(\mathbf{x}) \leq \lambda_{\max}(\mathbf{A})$$

当 $\mathbf{x} = \mathbf{u}_1 = a_1 \mathbf{q}_1, (a_1 \neq 0)$ 时:

$$\mathbf{x}^T \mathbf{x} = a_1^2, \quad \mathbf{x}^T \mathbf{A} \mathbf{x} = a_1^2 \lambda_1, \quad R(\mathbf{x}) = \lambda_{\max}(\mathbf{A})$$

当 $\mathbf{x} = \mathbf{u}_n = a_n \mathbf{q}_n, (a_n \neq 0)$ 时:

$$\mathbf{x}^T \mathbf{x} = a_n^2, \quad \mathbf{x}^T \mathbf{A} \mathbf{x} = a_n^2 \lambda_1, \quad R(\mathbf{x}) = \lambda_{\min}(\mathbf{A})$$

如果限制 $\|\mathbf{x}\| = 1$, 此时 $R(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ 。当 $\mathbf{x} = \mathbf{q}_1$ 时, $\mathbf{x}^T \mathbf{A} \mathbf{x}$ 取到极大值 λ_{\max} ; 当 $\mathbf{x} = \mathbf{q}_n$ 时, $\mathbf{x}^T \mathbf{A} \mathbf{x}$ 取到极小值 λ_{\min} 。 \square

通过对瑞利商的讨论, 我们也得到了如下推论:

推论 4.5.3. 对于一个对称矩阵 A 有

$$A \succeq 0 \iff \lambda_i(A) \geq 0, i = 1, \dots, n$$

$$A \succ 0 \iff \lambda_i(A) > 0, i = 1, \dots, n$$

定理 4.5.4. [Poincare 不等式] 令 $A \in S^n$, 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

$$\mathbf{x}^T \mathbf{A} \mathbf{x} \leq \lambda_k(A), \quad \mathbf{y}^T \mathbf{A} \mathbf{y} \geq \lambda_{n-k+1}(A)$$

我们这里将 n 阶对称矩阵的集合可以记为 S^n , n 阶半正定矩阵的集合可以记为 S_+^n , n 阶正定矩阵的集合可以记为 S_{++}^n 。

证明. 令 $A = \mathbf{U} \Lambda \mathbf{U}^T$ 是 A 的谱分解, 记 $\mathbb{Q} = \text{Col}(\mathbf{U}_k)$ 是 $\mathbf{U}_k = (\mathbf{u}_k, \dots, \mathbf{u}_n)$ 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

$$\begin{aligned} \mathbf{x}^T \mathbf{A} \mathbf{x} &= \boldsymbol{\eta}^T \mathbf{U}_k^T \mathbf{U} \Lambda \mathbf{U}^T \mathbf{U}_k \boldsymbol{\eta} = \sum_{i=k}^n \lambda_i(A) \eta_i^2 \\ &\leq \lambda_k(A) \sum_{i=k}^n \eta_i^2 = \lambda_k(A) \end{aligned}$$

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 $-A$ 用同样的处理方式即可证明。 \square

推论 4.5.4. [极小极大准则] 令 $A \in S^n$, 并令 \mathbb{V} 是 \mathbb{R}^n 中的子空间。那么, 对于 $k \in \{1, \dots, n\}$, 有

$$\begin{aligned} \lambda_k(A) &= \max_{\dim \mathbb{V}=k} \min_{\mathbf{x} \in \mathbb{V}, \|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{A} \mathbf{x} \\ &= \min_{\dim \mathbb{V}=n-k+1} \max_{\mathbf{x} \in \mathbb{V}, \|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{A} \mathbf{x} \end{aligned}$$

证明. 根据 Poincare 不等式, 如果 \mathbb{V} 是 \mathbb{R}^n 的 k 维子空间, 那么 $\min_{\mathbf{x} \in \mathbb{V}, \|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{A} \mathbf{x} \leq \lambda_k(\mathbf{A})$ 。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 4.5.5. 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

$$\begin{aligned} \lambda_k(\mathbf{A} + \mathbf{B}) &= \min_{\dim \mathbb{V}=n-k+1} \max_{\mathbf{x}: \|\mathbf{x}\|_2=1} (\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{x}^T \mathbf{B} \mathbf{x}) \\ &\geq \min_{\dim \mathbb{V}=n-k+1} \max_{\mathbf{x}: \|\mathbf{x}\|_2=1} \mathbf{x}^T \mathbf{A} \mathbf{x} + \lambda_{\min}(\mathbf{B}) \\ &= \lambda_k(\mathbf{A}) + \lambda_{\min}(\mathbf{B}) \end{aligned}$$

这就证明了推论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 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, $\mathbf{A} = \mathbf{G}\mathbf{G}^T$ 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

$$\mathbf{G} = \begin{pmatrix} g_{11} & & & \\ g_{21} & g_{22} & & \\ \vdots & \vdots & \ddots & \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{pmatrix} \quad (4.5)$$

比较 $\mathbf{A} = \mathbf{G}\mathbf{G}^T$ 两边, 易得

$$a_{ij} = \sum_{k=1}^j g_{jk} g_{ik}$$

从而有

$$g_{jj} g_{ij} = a_{ij} - \sum_{k=1}^{j-1} g_{jk} g_{ik} = v(i) \quad (4.6)$$

如果知道了 \mathbf{G} 的前 $j-1$ 列, 那么 $v(i)$ 就是可计算的。

在式 (4.6) 中令 $i = j$, 立即有 $g_{jj}^2 = v(j)$ 。然后, 由式 (4.6) 得

$$g_{ij} = v(i)/g_{jj} = v(i)/\sqrt{v(j)} \quad (4.7)$$

总结以上结论, 可得到计算 Cholesky 分解的下述算法 4.3:

算法 4.3 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 $A \in \mathbb{R}^{n \times n}$ 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 $\mathbf{G} \in \mathbb{R}^{n \times n}$ 的非零元素由式(4.7)决定。

例 4.5.2. 求矩阵 A 的 Cholesky 分解

$$A = \begin{bmatrix} 4 & & -1 \\ -1 & 4.25 & 2.75 \\ 1 & 2.75 & 3.5 \end{bmatrix}$$

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:

$$\mathbf{G} = \begin{bmatrix} 2 & 0 & 0 \\ -0.5 & 2 & 0 \\ 0.5 & 1.5 & 1 \end{bmatrix}$$

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

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \ddots & \vdots & \ddots & \ddots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \ddots & & \ddots & \\ l_{n1} & \cdots & l_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{pmatrix} \begin{pmatrix} 1 & l_{21} & \cdots & l_{n1} \\ & 1 & \cdots & l_{n2} \\ & & \ddots & \ddots \\ & & & 1 \end{pmatrix}$$

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：

算法 4.4 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

```

数据科学与工程数学基础初稿

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。

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。

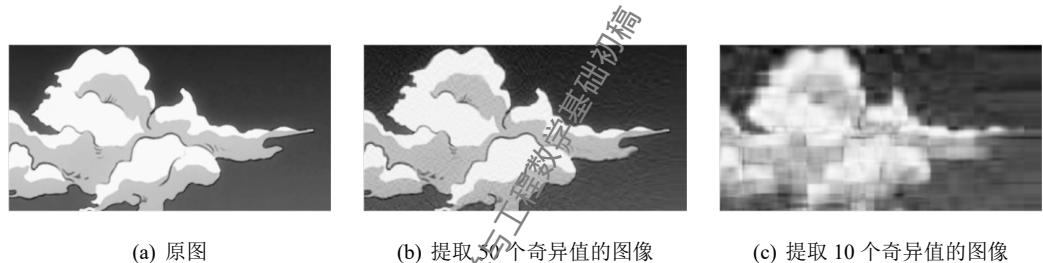


图 4.9: 使用 SVD 进行图像压缩

4.6.1 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：

$$A = U \Sigma V^T \quad (4.8)$$

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。

$$m \begin{array}{|c|} \hline A \\ \hline n \end{array} = m \begin{array}{|c|} \hline U \\ \hline m \end{array} m \begin{array}{|c|} \hline \Sigma \\ \hline n \end{array} n \begin{array}{|c|} \hline V^T \\ \hline n \end{array}$$

图 4.10: 完全奇异值分解

$$\text{例 4.6.1. 矩阵 } A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} \text{ 的奇异值分解为:}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & \sqrt{0.2} & -\sqrt{0.8} & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{0.8} & \sqrt{0.2} & 0 \end{pmatrix} \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & \sqrt{5} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

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 $U_r \Sigma_r V_r^T$ 为 A 的紧奇异值分解, 即

$$A = U_r \Sigma_r V_r^T$$

其中 $U_r \in \mathbb{R}^{m \times r}, V_r \in \mathbb{R}^{n \times r}, \Sigma_r$ 是 r 阶对角矩阵; 矩阵 U_r 由完全奇异值分解中 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。

$$m \begin{array}{|c|} \hline A \\ \hline n \end{array} = m \begin{array}{|c|} \hline U \\ \hline r \end{array} r \begin{array}{|c|} \hline \Sigma \\ \hline r \end{array} r \begin{array}{|c|} \hline V^T \\ \hline n \end{array}$$

图 4.11: 紧 SVD

例 4.6.2. 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 $U_k \Sigma_k V_k^T$ 为矩阵 A 的截断奇异值分解, 即

$$A \approx U_k \Sigma_k V_k^T$$

其中 $U_k \in \mathbb{R}^{m \times k}$, $V_k \in \mathbb{R}^{n \times k}$, Σ_k 是 k 阶对角矩阵。矩阵 U_k 由完全奇异值分解中 U 的前 k 列、矩阵 V_k 由 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 A 的秩低。

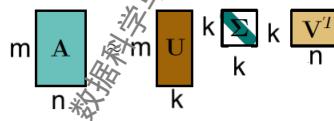


图 4.12: SVD

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

例 4.6.3. 由例 4.6.1 给出的矩阵 $A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$ 的秩为 3, 若取 $k = 2$, 则其截断奇异值

分解为

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} \approx \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 4 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

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。

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 $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ 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 $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m$ 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 $A = U\Sigma V^T$ 的线性变换，等价于经过坐标系的旋转或者反射变换 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 U 得到向量 $A\mathbf{x} \in \mathbb{R}^m$ 。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。

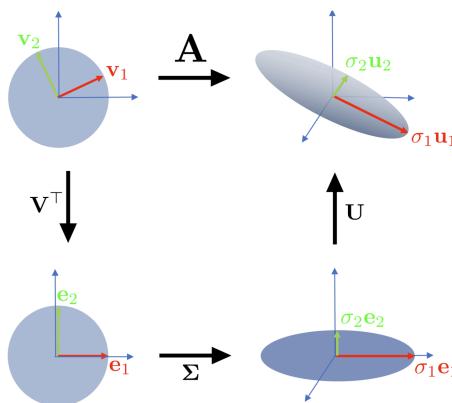


图 4.13: 奇异值分解的几何意义图解

下面通过一个例子直观地说明奇异值分解的几何意义。

例 4.6.4. 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 $\Sigma \mathbf{V}^T \mathbf{e}_1, \Sigma \mathbf{V}^T \mathbf{e}_2$ 旋转得到

$$\mathbf{A} \mathbf{e}_1 = \mathbf{U} \Sigma \mathbf{V}^T \mathbf{e}_1 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \mathbf{A} \mathbf{e}_2 = \mathbf{U} \Sigma \mathbf{V}^T \mathbf{e}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

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。

奇异值分解基本定理

对于半正定矩阵来说, 奇异值分解总是存在的。因为, 我们知道如果 \mathbf{A} 是半正定矩阵, 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

$$(AA^T) A v_i = \lambda_i A v_i, i = 1, \dots, r$$

这就意味着 $A v_i$ 是 AA^T 的特征向量, 因为 $A^T A^T A v_j = \lambda_j v_i^T v_j$, 所以这些特征向量也是正交的。所以将他们标准化则有

$$u_i = \frac{A v_i}{\sqrt{\lambda_i}} = \frac{A v_i}{\sigma_i}, i = 1, \dots, r$$

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

$$\begin{pmatrix} u_1^T \\ \vdots \\ u_r^T \end{pmatrix} A \begin{pmatrix} v_1, \dots, v_r \end{pmatrix} = \text{diag}(\sigma_1, \dots, \sigma_r) = \Sigma_r \quad (4.9)$$

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

$$A v_i = 0, i = r+1, \dots, n$$

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(4.9)有

$$\begin{pmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_m^T \end{pmatrix} \mathbf{A} \begin{pmatrix} \mathbf{v}_1, \dots, \mathbf{v}_n \end{pmatrix} = \begin{pmatrix} \Sigma_r & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \Sigma$$

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。

□

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 $A = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 0 & 0 \end{pmatrix}$ 的奇异值分解

解. 求对称矩阵

$$A^T A = \begin{pmatrix} 5 & 5 \\ 5 & 5 \end{pmatrix}$$

求 $A^T A$ 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 $v_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, v_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, 所以正交矩阵 $V = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ 。

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 U , 基于 A 的正奇异值计算得到列向量 $u_1 = \frac{1}{\sigma_1} A v_1 = \frac{1}{\sqrt{10}} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$, 而列向量 u_2, u_3 是 A^T 零空间 $\text{Null}(A^T)$ 的一组标准正交基, 所以 $u_2 = \frac{1}{\sqrt{5}} \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}, u_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$, 故正交矩阵 U 为 $U = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & -2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & \sqrt{5} \end{pmatrix}$ 。

所以 A 的奇异值分解为

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

奇异值分解和特征分解

性质 4.6.1. 设矩阵 A 的奇异值分解为 $A = U \Sigma V^T$, 则以下关系成立:

$$A^T A = (U \Sigma V^T)^T (U \Sigma V^T) = V (\Sigma^T \Sigma) V^T$$

$$A A^T = (U \Sigma V^T) (U \Sigma V^T)^T = U (\Sigma \Sigma^T) U^T$$

也就是说, 矩阵 $A^T A, A A^T$ 的特征分解存在, 且可以由矩阵 A 的奇异值分解的矩阵表示。 V 的列向量是 $A^T A$ 的特征向量, U 的列向量是 $A A^T$ 的特征向量, Σ 是奇异值是 $A^T A, A A^T$ 的特征值的平方根。

性质 4.6.2. 矩阵 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 U, V 不是唯一的。

在矩阵 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 A 的奇异值分解为 $A = U\Sigma V^T$, 则以下关系成立:

$$A\mathbf{v}_j = \sigma_j \mathbf{u}_j, j = 1, 2, \dots, n$$

$$\begin{cases} A^T \mathbf{u}_j = \sigma_j \mathbf{v}_j & j = 1, 2, \dots, n \\ A^T \mathbf{u}_j = 0 & j = n+1, n+2, \dots, m \end{cases}$$

证明. 由 $A = U\Sigma V^T$, 易知 $AV = U\Sigma$ 。比较这一等式两端的第 j 列, 得到 $A\mathbf{v}_j = \sigma_j \mathbf{u}_j, j = 1, 2, \dots, n$, 这是矩阵 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 A 的左奇异向量和奇异值、右奇异向量的关系。□

考虑矩阵 A 的特征分解 $A = PDP^{-1}$ 和奇异值分解 $A = U\Sigma V^T$ 。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:

1. 改变空间的基底。
2. 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。
3. 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 $A^T A$ 的特征向量。
- A 非零奇异值是 $A^T A$ 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。

4.6.2 基于奇异值分解的矩阵性质

本节, 我们将利用矩阵 $A \in \mathbb{R}^{m \times n}$ 的完全奇异值分解或紧奇异值分解

$$A = U\Sigma V^T = U_r \Sigma_r V_r^T$$

来重新探讨矩阵 A 关于秩、零空间、列空间、矩阵范数、矩阵广义逆、正交投影相关的一些性质：

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)。

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 $\dim(\text{Null}(A)) = n - r$ 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 $\Sigma_r V_r^T \in \mathbb{R}^{r \times n}, r \leq n$, 是一个行满秩矩阵, 则当 \mathbf{x} 张成整个 \mathbb{R}^n 时, $\mathbf{z} = \Sigma_r V_r^T \mathbf{x}$ 张成整个 \mathbb{R}^r 。因此

$$\begin{aligned} \text{Col}(A) &= \{\mathbf{y} | \mathbf{y} = A\mathbf{x}, \mathbf{x} \in \mathbb{R}^n\} \\ &= \{\mathbf{y} | \mathbf{y} = U_r \Sigma_r V_r^T \mathbf{x}, \mathbf{x} \in \mathbb{R}^n\} \\ &= \{\mathbf{y} | \mathbf{y} = U_r \mathbf{z}, \mathbf{z} \in \mathbb{R}^r\} \\ &= \text{Col}(U_r) \end{aligned}$$

□

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

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & \sqrt{0.2} & -\sqrt{0.8} & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \sqrt{0.8} & \sqrt{0.2} & 0 \end{pmatrix} \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & \sqrt{5} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

所以 $Col(A) = Col \begin{pmatrix} 0 & 0 & \sqrt{0.2} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sqrt{0.8} \end{pmatrix}$, $Null(A) = Col \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$

命题 4.6.1. 矩阵的 F 范数满足以下等式

$$\|A\|_F^2 = \text{Tr}(A^T A) = \sum_{i=1}^n \lambda_i(A^T A) = \sum_{i=1}^n \sigma_i^2$$

其中 σ_i 是矩阵 A 的奇异值。

命题 4.6.2. 矩阵 A 2 范数的平方是 $A^T A$ 的最大特征值, 所以 $\|A\|_2^2 = \sigma_1^2$ 。即 A 的 2 范数就是 A 的最大的奇异值。

命题 4.6.3. 对于矩阵核范数, 我们有

$$\|A\|_* = \sum_{i=1}^r \sigma_i, r = \text{rank}(A)$$

定义 4.6.4. 令 A 是一个 $m \times n$ 矩阵, 若存在一个的 $n \times m$ 矩阵 G , 使得下列条件满足:

$$(AG)^T = AG$$

$$(GA)^T = GA$$

$$GAG = G$$

$$AGA = A$$

则称 G 是 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 $A \in \mathbb{R}^{m \times n}$, 如果 G 是 A 的 Moore-Penrose 逆, 那么 G 是 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。

我们可以利用奇异值分解求解广义逆。

若矩阵 M 的奇异值分解为 $M = U\Sigma V^T$, 那么 M 的伪逆为

$$M^\dagger = V\Sigma^\dagger U^T$$

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

$$A = U\Sigma V^T = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{6}} & 0 & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

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

$$(A^T A)^{-1} A^T = (V \Sigma^{-2} V^T) V \Sigma^T U^T = V \Sigma^{-1} U^T = A^\dagger$$

□

A 所有的左逆都可以表示为 $A^{li} = A^\dagger + Q^T$, 其中 Q 满足 $A^T Q = 0$ 。

性质 4.6.10. 当矩阵 $A \in \mathbb{R}^{m \times n}$ 为行满秩矩阵时有

$$A^\dagger = A^T (A A^T)^{-1}$$

证明. 如果 $A \in \mathbb{R}^{m \times n}$ 是一个行满秩矩阵, 因此 $r = m \leq n$, 故有

$$A A^\dagger = U_r U_r^T = I_m$$

所以 A^\dagger 是矩阵 A 的右逆 (即 $A A^\dagger = I_m$)。注意到 $A A^\dagger$ 是可逆的, 所以

$$A^T (A A^T)^{-1} = V \Sigma^T U^T (U \Sigma^{-2} V^T)^{-1} = V \Sigma^{-1} U^T = A^\dagger$$

□

A 所有的右逆都可以表示为 $A^{ri} = A^\dagger + Q$, 其中 Q 满足 $A Q = 0$ 。

例 4.6.9. 求矩阵 $A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$ 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

$$\begin{aligned} A^\dagger &= (A^T A)^{-1} A^T = \left(\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \\ &= \frac{1}{3} \begin{bmatrix} -1 & 1 & 2 \\ 2 & 1 & -1 \end{bmatrix} \end{aligned}$$

我们知道任何一个矩阵 $A \in \mathbb{R}^{m \times n}$ 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:

$$\mathbb{R}^n = \text{Null}(A) \oplus \text{Null}(A)^\perp = \text{Null}(A) \oplus \text{Col}(A^T)$$

$$\mathbb{R}^m = \text{Col}(A) \oplus \text{Col}(A)^\perp = \text{Col}(A) \oplus \text{Null}(A^T)$$

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 $\mathbf{x}^* = \mathbf{B}\alpha$, 其中 $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_d), \alpha \in \mathbb{R}^d$, 并且我们需要解方程 $\mathbf{B}^T \mathbf{B} \alpha = \mathbf{B}^T \mathbf{x}$ 来得到 α 。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 $\alpha = \mathbf{B}^T \mathbf{x}$, 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 $\mathbf{P}_{\text{Null}(\mathbf{A})} = (\mathbf{V}_{nr} \mathbf{V}_{nr}^T) = \mathbf{I}_n - \mathbf{V}_r \mathbf{V}_r^T = \mathbf{I}_n - \mathbf{A}^T \mathbf{A}$ 。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 $\pi_{\text{Null}(\mathbf{A}^T)}(\mathbf{y}) = (\mathbf{I}_m - \mathbf{A} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T) \mathbf{y}$ 。

4.6.3 奇异值分解与低秩表示

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 $m \times n$ 的秩一矩阵, 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。

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

$$\begin{aligned} \min_{\hat{\mathbf{A}}(k) \in \mathbb{R}^{m \times n}} & \|\mathbf{A} - \hat{\mathbf{A}}(k)\|_F, \\ \text{s.t. } & \text{rank}(\hat{\mathbf{A}}(k)) = k \end{aligned}$$

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 $\mathbf{A} \in \mathbb{R}^{m \times n}$, 矩阵的秩 $\text{rank}(\mathbf{A}) = r$, 有奇异值分解 $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$, 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 $\mathbf{X} \in \mathbb{M}$, 满足

$$\|\mathbf{A} - \mathbf{X}\|_F = \min_{\mathbf{S} \in \mathbb{M}} \|\mathbf{A} - \mathbf{S}\|_F$$

则 $\|\mathbf{A} - \mathbf{X}\|_F = (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{\frac{1}{2}}$ 。特别地, 若 $\mathbf{A}' = \mathbf{U}'\Sigma'\mathbf{V}^T$, 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 $\|\mathbf{A} - \mathbf{A}'\|_F = (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{\frac{1}{2}} = \min_{\mathbf{S} \in \mathbb{M}} \|\mathbf{A} - \mathbf{S}\|_F$ 。

证明. 若秩为 k 的矩阵 $\mathbf{X} \in \mathbb{M}$, 满足 $\|\mathbf{A} - \mathbf{X}\|_F = \min_{\mathbf{S} \in \mathbb{M}} \|\mathbf{A} - \mathbf{S}\|_F$ 则

$$\|\mathbf{A} - \mathbf{X}\|_F \leq \|\mathbf{A} - \mathbf{A}'\|_F = (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{\frac{1}{2}}$$

下面证明 $\|\mathbf{A} - \mathbf{X}\|_F \geq (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{\frac{1}{2}}$ 。设 \mathbf{X} 的奇异值分解为 $\mathbf{Q}\Omega\mathbf{P}^T$, 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 $\mathbf{B} = \mathbf{Q}^T \mathbf{A} \mathbf{P}$, 则 $\mathbf{A} = \mathbf{Q} \mathbf{B} \mathbf{P}^T$, 由此得到 $\|\mathbf{A} - \mathbf{X}\|_F = \|\mathbf{Q}(\mathbf{B} - \Omega)\mathbf{P}^T\|_F = \|\mathbf{B} - \Omega\|_F$ 。用 Ω 分块方法对 \mathbf{B} 分块

$$\mathbf{B} = \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{pmatrix}$$

其中 $\mathbf{B}_{11} \in \mathbb{R}^{k \times k}$, $\mathbf{B}_{12} \in \mathbb{R}^{k \times (n-k)}$, $\mathbf{B}_{21} \in \mathbb{R}^{(n-k) \times k}$, $\mathbf{B}_{22} \in \mathbb{R}^{(n-k) \times (n-k)}$ 可得

$$\|\mathbf{A} - \mathbf{X}\|_F^2 = \|\mathbf{B} - \Omega\|_F^2 = \|\mathbf{B}_{11} - \Omega_k\|_F^2 + \|\mathbf{B}_{12}\|_F^2 + \|\mathbf{B}_{21}\|_F^2 + \|\mathbf{B}_{22}\|_F^2$$

现证 $\mathbf{B}_{12} = 0$, $\mathbf{B}_{21} = 0$ 。用反证法。若 $\mathbf{B}_{12} \neq 0$, 令

$$\mathbf{Y} = \mathbf{Q} \begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ 0 & 0 \end{pmatrix} \mathbf{P}^T$$

则 $\mathbf{Y} \in \mathbb{M}$ 且 $\|\mathbf{A} - \mathbf{Y}\|_F^2 = \|\mathbf{B}_{21}\|_F^2 + \|\mathbf{B}_{22}\|_F^2 < \|\mathbf{A} - \mathbf{X}\|_F^2$, 这与 \mathbf{X} 的定义 $\|\mathbf{A} - \mathbf{X}\|_F = \min_{\mathbf{S} \in \mathbb{M}} \|\mathbf{A} - \mathbf{S}\|_F$ 矛盾, 因此 $\mathbf{B}_{12} = 0$ 。

同理可证 $\mathbf{B}_{21} = 0$ 。于是

$$\|\mathbf{A} - \mathbf{X}\|_F^2 = \|\mathbf{B}_{11} - \Omega_k\|_F^2 + \|\mathbf{B}_{22}\|_F^2$$

再证 $\mathbf{B}_{11} = \Omega_k$, 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

$$\|\mathbf{A} - \mathbf{Z}\|_F^2 = \|\mathbf{B}_{22}\|_F^2 \leq \|\mathbf{B}_{11} - \Omega_k\|_F^2 + \|\mathbf{B}_{22}\|_F^2 = \|\mathbf{A} - \mathbf{X}\|_F^2$$

由 \mathbf{X} 的定义

$$\|\mathbf{A} - \mathbf{X}\|_F = \min_{\mathbf{S} \in \mathbb{M}} \|\mathbf{A} - \mathbf{S}\|_F$$

知

$$\|\mathbf{B}_{11} - \Omega_k\|_F^2 = 0$$

即 $\mathbf{B}_{11} = \Omega_k$ 。

最后看 \mathbf{B}_{22} , 设 \mathbf{B}_{22} 有奇异值分解为 $\mathbf{U}_1 \mathbf{\Lambda} \mathbf{V}_1^T$, 则 $\|\mathbf{A} - \mathbf{X}\|_F = \|\mathbf{B}_{22}\|_F = \|\mathbf{\Lambda}\|_F$ 。下面证明 $\mathbf{\Lambda}$ 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

$$\mathbf{U}_2^T \mathbf{Q}^T \mathbf{A} \mathbf{P} \mathbf{V}_2 = \begin{pmatrix} \mathbf{\Omega}_k & \\ & \mathbf{\Lambda} \end{pmatrix}, \mathbf{A} = (\mathbf{Q} \mathbf{U}_2) \begin{pmatrix} \mathbf{\Omega}_k & \\ & \mathbf{\Lambda} \end{pmatrix} (\mathbf{P} \mathbf{V}_2)^T$$

由此可知 $\mathbf{\Lambda}$ 的对角线元素为 \mathbf{A} 的奇异值, 故有

$$\|\mathbf{A} - \mathbf{X}\|_F = \|\mathbf{\Lambda}\|_F \geq (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{\frac{1}{2}}$$

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} 。
 $\mathbf{A}' = \mathbf{U} \mathbf{\Sigma}' \mathbf{V}^T$ 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 $\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}$ 秩为 2 的最优近似。

解. 先对 \mathbf{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

$$\mathbf{A}' = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 4 & 0 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

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, 便可以共使用 $k(n+m+1)$ 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

$$\rho = \frac{nm}{k(n+m+1)} \quad (4.10)$$

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 $k < \frac{nm}{n+m+1}$ 。

图4.14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大致是什么了。
- 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。

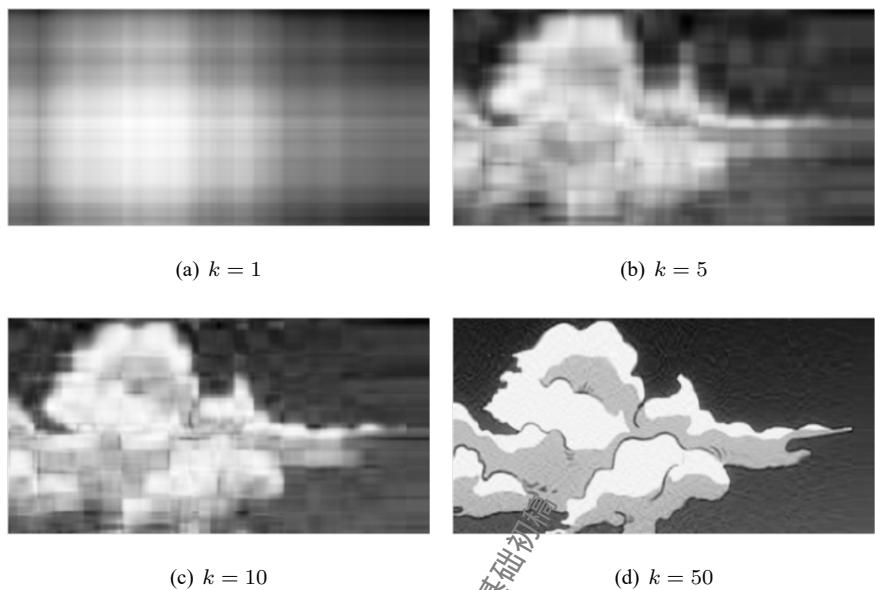
原图是一张 1328×680 的图像, 要传输这样一张图像需要发送 $1328 \times 680 = 903040$ 个数值。而如果使用 $k = 50$ 时的截断 SVD, 那么只需要传送 $50 \times (1328 + 680 + 1) = 100450$ 个数值。也就是说压缩比达到了 8.9899。

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 u_1, u_2, \dots, u_k 和右奇异向量 v_1, v_2, \dots, v_k 后, 即可通过截断奇异值分解公式

$$\hat{A} = \sum_{i=1}^k \sigma_i u_i v_i^T \quad (4.11)$$

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 ρ 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。

图 4.14: 不同 k 值对压缩图像的影响

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：对角矩阵、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一矩阵的线性组合，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，把它表示为对角矩阵、三角矩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)。

工程数学与习题

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:

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

习题 4.8. 求矩阵

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

的 QR 分解。

习题 4.9. 求矩阵 $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ 的奇异值分解。

习题 4.10. 求 $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$ 的奇异值分解。

习题 4.11. 设 $A = \begin{bmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \end{bmatrix}$, 求 A 的奇异值分解。

习题 4.12. 已知

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

求 A 的奇异值分解表达式。

习题 4.13. 已知 $A \in \mathcal{C}_r^{m \times n}$ (秩为 $r > 0$) 的奇异值分解表达式为

$$A = U \begin{pmatrix} \Delta & 0 \\ 0 & 0 \end{pmatrix} V^H$$

试求矩阵 $B = \begin{pmatrix} A \\ A \end{pmatrix}$ 的奇异值分解表达式。

习题 4.14. 已知矩阵

$$A = \begin{pmatrix} 0 & 2 & 4 \\ \frac{1}{2} & 0 & 2 \\ \frac{1}{4} & \frac{1}{2} & 0 \end{pmatrix}$$

验证 A 是可对角化矩阵，并求 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，

$$\begin{aligned} \nabla_{\mu} \sum_{i=1}^n \|x_i - (\mu_n + V\beta_i)\|_2^2 = 0 &\Leftrightarrow \sum_{i=1}^n (x_i - (\mu + V\beta_i)) = 0 \\ &\Leftrightarrow (\sum_{i=1}^n x_i) - n\mu - V(\sum_{i=1}^n \beta_i) = 0 \end{aligned}$$

有不失一般性假设 $\sum_{i=1}^n \beta_i = \mathbf{0}$ 。

证明，对任意的 b ，假设 $\sum_{i=1}^n \beta_i = b$ ，最终得到 x_i 的拟合值 $\mu + V\beta_i$ 是相等的。

习题 4.16. 由 $\|x\|_2^2 = \langle x, x \rangle$ 和 $V^T V = I$ ，证明：

$$\|(x_i - \mu_n) - VV^T(x_i - \mu_n)\|_2^2 = (x_i - \mu_n)^T(x_i - \mu_n) - (x_i - \mu_n)^T VV^T(x_i - \mu_n)$$

习题 4.17. 利用矩阵迹的性质，证明：

$$\sum_{i=1}^n (x_i - \mu_n)^T VV^T (x_i - \mu_n) = (n-1) \operatorname{Tr}(V^T \Sigma_n V)$$

其中 $\Sigma_n = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu_i)(x_i - \mu_i)^T$ 。

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, Schölkopf, Bernhard, and Müller, 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] Ortmann, 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.

第五章 矩阵计算问题

在众多自然科学和工程学科中,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:

- **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。
- **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。
- **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组并给出它的标准形式 $\mathbf{Ax} = \mathbf{b}$, 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, $\mathbf{b} \in \mathbb{R}^m$ 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 $\mathbf{Ax} = \mathbf{b}$ 可以被看成优化问题, 即关于 \mathbf{x} , 最小化 $\|\mathbf{Ax} - \mathbf{b}\|^2$ 。我们描述线性方程组解的集合并且当线性方程组精确解不存在的情况下, 讨论求解线性方程组近似解的方法, 由此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。因为矩阵分解以及工程中很多矩阵计算问题都与特征值计算密切相关, 所以也将详细介绍特征值的求解理论和方法。

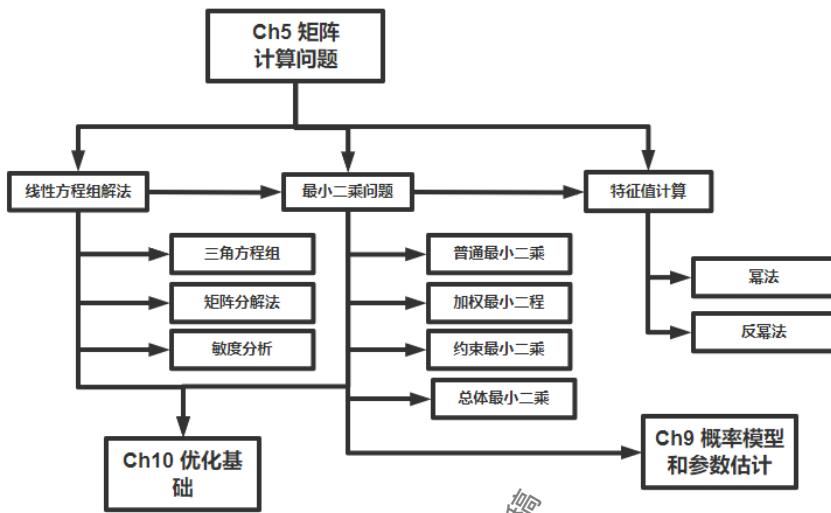


图 5.1: 本章导图

5.1 线性方程组的直接解法

线性方程组的求解问题是一个古老的数学问题。中国古代的《九章算术》中已详细地载述了解线性方程组的消元法。19世纪初，西方也有了Gauss消去法。到了20世纪中叶，如何利用计算机来快速、有效地求解未知数多的大型线性方程组是数值线性代数研究的核心问题。求解线性方程组的数值方法大体上可分为直接法和迭代法两种。直接法是指在没有舍入误差的情况下经过有限次运算可求得方程组的精确解的方法，因此直接法又称为精确法。迭代法是指是采取逐次逼近的方法，亦即从一个初始向量出发，按照一定的计算格式，构造一个向量的无穷序列，其极限才是方程组的精确解，只经过有限次运算得不到精确解。

本节主要介绍解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。线性方程组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。线性方程组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。接下来我们展示一个线性方程组的例子。

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 $\mathbf{x} = (x_1, x_2)^\top$ 到测距点的距离为 d_1, d_2, d_3 , \mathbf{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

$$\|\mathbf{x} - \mathbf{a}_1\|_2^2 = d_1^2, \quad \|\mathbf{x} - \mathbf{a}_2\|_2^2 = d_2^2, \quad \|\mathbf{x} - \mathbf{a}_3\|_2^2 = d_3^2 \quad (5.1)$$

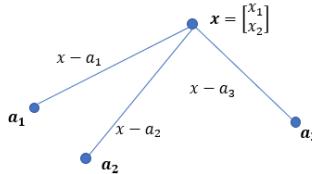


图 5.2: 三点测量位置

我们需要在点 \mathbf{x} 处测量距三个测距点 $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ 的距离, 以便确定 \mathbf{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 \mathbf{x} 的线性方程组。

$$2(\mathbf{a}_2 - \mathbf{a}_1)^\top \mathbf{x} = d_1^2 - d_2^2 + \|\mathbf{a}_2\|_2^2 - \|\mathbf{a}_1\|_2^2$$

$$2(\mathbf{a}_3 - \mathbf{a}_1)^\top \mathbf{x} = d_1^2 - d_3^2 + \|\mathbf{a}_3\|_2^2 - \|\mathbf{a}_1\|_2^2$$

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 $\mathbf{Ax} = \mathbf{b}$ (标准形式的定义在下一小节给出)可以描述为:

$$A = \begin{pmatrix} 2(\mathbf{a}_2 - \mathbf{a}_1)^\top \\ 2(\mathbf{a}_3 - \mathbf{a}_1)^\top \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} d_1^2 - d_2^2 + \|\mathbf{a}_2\|_2^2 - \|\mathbf{a}_1\|_2^2 \\ d_1^2 - d_3^2 + \|\mathbf{a}_3\|_2^2 - \|\mathbf{a}_1\|_2^2 \end{pmatrix} \quad (5.2)$$

上述问题的解, 将在后面详细讨论。

5.1.2 一般线性方程组解的理论

正如先前的示例, 一般的线性方程被描述为如下的向量形式:

$$\mathbf{Ax} = \mathbf{b} \quad (5.3)$$

其中 $\mathbf{x} \in \mathbb{R}^n$ 是未知变量, $\mathbf{A} \in \mathbb{R}^{m \times n}$ 是系数矩阵, $\mathbf{b} \in \mathbb{R}^m$ 是已知向量。

接下来我们先讨论线性方程组的基本性质, 主要针对解的存在性、唯一性以及描述线性方程组所有可能的解。

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

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m. \end{array} \right. \quad (5.4)$$

若记

$$\mathbf{A} = (a_{ij})_{m \times n}, \mathbf{x} = (x_1, x_2, \dots, x_n)^T, \mathbf{b} = (b_1, b_2, \dots, b_m)^T, \quad (5.5)$$

则方程组(5.4)可表为如下的矩阵形式:

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \quad (5.6)$$

当 $\mathbf{b} \neq \mathbf{0}$ 时, 方程组(5.6)称为非齐次线性方程组, 当 $\mathbf{b} = \mathbf{0}$ 时。方程组称为方程组(5.6)对应的齐次线性方程组。

$$\mathbf{A}\mathbf{x} = \mathbf{0}. \quad (5.7)$$

定义 5.1.1. 矩阵 $\mathbf{A} = (a_{ij})_{m \times n}$, 称为方程组(5.6)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(5.7)称为方程组(5.6)的导出组。

定义 5.1.2. 若向量 $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$ 满足方程组(5.6), 即 $\mathbf{A}\mathbf{x}^0 = \mathbf{b}$, 称 \mathbf{x}^0 为方程组(5.6)的解向量。

定义 5.1.3. 给定方程组

$$\bar{\mathbf{A}}\mathbf{x} = \bar{\mathbf{b}}, \quad (5.8)$$

其中 $\bar{\mathbf{A}} = (\bar{a}_{ij})_{m \times n}$, $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, $\bar{\mathbf{b}} = (\bar{b}_1, \bar{b}_2, \dots, \bar{b}_m)^T$ 。当 $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$ 是方程组(5.8)的解向量时, 若它也是方程组(5.6)的解向量, 则称方程组(5.6)与方程组(5.8)是同解方程组。

定理 5.1.1. 对方程组(5.6)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 5.1.4. 设 $\eta_1, \eta_2, \dots, \eta_t$ 是齐次线性方程组(5.7)的解向量组, 如果 $\eta_1, \eta_2, \dots, \eta_t$ 线性无关, 且方程组(5.7)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(5.7)的一个基础解系。

定理 5.1.2. 设齐次线性方程组(5.7)的系数矩阵 \mathbf{A} 的秩为 r , 此时

- (1) 方程组(5.7)有非零解的必要充分条件是 $r < n$ 。
- (2) 若 $r < n$, 则方程组(5.7)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(5.7)的一个基础解系, 则它的一般解为

$$\eta = \lambda_1 \eta_1 + \lambda_2 \eta_2 + \dots + \lambda_{n-r} \eta_{n-r}, \quad (5.9)$$

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(5.6)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 $\text{rank}(\mathbf{A}) = \text{rank}(\tilde{\mathbf{A}}) = r$, γ_0 是非齐次方程组(5.6)的一个解向量(常称为特解), $\eta_1, \eta_2, \dots, \eta_{n-r}$ 是其导出组(5.7)的一个基础解系, 则方程组(5.6)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)。

线性方程组的解集与基本子空间

线性方程组 $\mathbf{Ax} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ 的解集被定义为:

$$S \doteq \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b}\} \quad (5.10)$$

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:

$$\mathbf{Ax} = x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \dots + x_n \mathbf{a}_n \quad (5.11)$$

回顾定义, \mathbf{A} 的列空间定义为:

$$\text{Col}(\mathbf{A}) = \text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_n\},$$

其中 \mathbf{a}_i 为 \mathbf{A} 的列向量。 \mathbf{A} 的零空间定义为:

$$\text{Null}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = 0\} \quad (5.12)$$

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:

$$\mathbb{S}^\perp = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{y}^T \mathbf{x} = 0, \forall \mathbf{y} \in \mathbb{S}\} \quad (5.13)$$

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 $\mathbf{Ax} \in \text{Col}(\mathbf{A})$ 。若 $\mathbf{b} \notin \text{Col}(\mathbf{A})$, 则线性方程组没有解。因此解集 \mathbb{S} 为空。等价地, 线性方程组有解当且仅当 $\mathbf{b} \in \text{Col}(\mathbf{A})$, 也即 \mathbf{b} 是 \mathbf{A} 的列的线性组合。

从矩阵的列空间的角度, 给出定理5.1.3的证明:

定理 5.1.5. 方程组的解(5.6)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 $\mathbf{b} = \sum_{i=1}^n x_i \mathbf{a}_i$, 这里 $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n]$, 所以令 $\mathbf{x} = (x_1, \dots, x_n)^T$, 即有 $\mathbf{Ax} = \mathbf{b}$ 。 \square

定理 5.1.6. 假定方程组(5.6)的解存在, 并且假定 \mathbf{x} 是其任一给定的解, 则(5.6)全部解的集合是

$$\mathbf{x} + \text{Null}(\mathbf{A}) \quad (5.14)$$

证明. 如果 \mathbf{y} 满足(5.6), 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 $\mathbf{A}\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{A}\mathbf{z} = \mathbf{A}\mathbf{x} = \mathbf{b}$ 。□

定理5.1.6告诉我们, 只要知道了方程组(5.6)的一个解, 便可以用它及 $\text{Null}(\mathbf{A})$ 中向量的和得到(5.6)的全部解。由此可知, 方程组(5.6)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(5.6)的解唯一的充分必要条件是 $\text{nullity}(\mathbf{A}) = 0$ 。

线性方程组分类

按照矩阵 \mathbf{A} 的秩与行列数的关系, 可以划分为三种不同类型的方程。我们先简略的考虑这三种情形下的解集, 并在之后的章节中详细介绍如何求解。

方阵系统 线性方程组 $\mathbf{Ax} = \mathbf{b}$ 中方程的个数等于未知变量的个数时, 也即矩阵 \mathbf{A} 是方阵, 则我们称 $\mathbf{Ax} = \mathbf{b}$ 是方阵系统。如果系数矩阵是满秩的即 $\text{rank}(\mathbf{A}) = n$ 可逆, \mathbf{A}^{-1} 唯一且有 $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ 。在这种情况下, 线性方程组的解是唯一的:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y} \quad (5.15)$$

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。

超定系统 线性方程组 $\mathbf{Ax} = \mathbf{b}$ 中线性方程的个数大于未知变量的个数时, 也即矩阵 \mathbf{A} 的行数大于列数: $m > n$, 则我们称 $\mathbf{Ax} = \mathbf{b}$ 是超定系统或超定方程组。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。在超定系统中, $\mathbf{y} \notin \text{Range}(\mathbf{A})$ 是很常见的, 因此引入近似解的概念, 近似解使得 \mathbf{Ax} 与 \mathbf{y} 在合适的度量下距离最小, 这与最小二乘相联系。

欠定系统 线性方程组 $\mathbf{Ax} = \mathbf{b}$ 中未知变量的个数大于方程组的个数时, 或者说 \mathbf{A} 的列数大于行数: $m < n$, 则我们称 $\mathbf{Ax} = \mathbf{b}$ 是欠定系统或欠定方程组。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:

$$\text{rank}(\mathbf{A}) + \dim(\text{Null}(\mathbf{A})) = n \quad (5.16)$$

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。

5.1.3 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 5.1.5. 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 $4mn^2$ 。

例 5.1.3. 为了完成两个向量 $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ 的内积运算 $\mathbf{x} \cdot \mathbf{y}$ ，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 5.1.4. 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 5.1.5. 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。

下三角形线性方程组

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

$$\begin{pmatrix} a_{11} & & & & & & \\ 0 & a_{22} & & & & & \\ 0 & 0 & a_{33} & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ 0 & 0 & 0 & \dots & a_{kk} & & \\ 0 & 0 & 0 & \dots & a_{k+1k} & a_{k+1k+1} & \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \ddots \\ 0 & 0 & 0 & \dots & a_{nk} & a_{nk+1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ \vdots \\ b_k^{(k-1)} \\ b_{k+1}^{(k-1)} \\ \vdots \\ b_n^{(k-1)} \end{pmatrix}$$

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 。

$$\begin{pmatrix} a_{11} & & & & & & \\ 0 & a_{22} & & & & & \\ 0 & 0 & a_{33} & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ 0 & 0 & 0 & \dots & a_{kk} & & \\ 0 & 0 & 0 & \dots & 0 & a_{k+1k+1} & \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \ddots \\ 0 & 0 & 0 & \dots & 0 & a_{nk+1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ \vdots \\ b_k^{(k-1)} \\ b_{k+1}^{(k)} \\ \vdots \\ b_n^{(k)} \end{pmatrix}$$

所以前代法，就从前 (x_1) 往后 (x_n) 来依次求解。

算法 5.1 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

```

上三角形线性方程组

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

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{22} & \dots & \dots & a_{2n} \\ \vdots & & & \vdots \\ a_{nn} & & & \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

与前代法类似，在回代法第 $(n-k+1)$ 个循环内。

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k-1} & a_{1k} & 0 & \dots & 0 \\ a_{22} & \dots & \dots & a_{2k-1} & a_{2k} & 0 & \dots & 0 \\ \vdots & & & \vdots & \vdots & \vdots & & \vdots \\ a_{k-1k-1} & a_{k-1k} & & 0 & \dots & 0 \\ a_{kk} & 0 & \dots & 0 \\ a_{k+1k+1} & \dots & 0 \\ \vdots & & \vdots \\ a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^{(n-k)} \\ b_2^{(n-k)} \\ \vdots \\ b_{k-1}^{(n-k)} \\ b_k^{(n-k)} \\ b_{k+1}^{(n-k-1)} \\ \vdots \\ b_n \end{pmatrix}$$

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 。

$$\left(\begin{array}{ccccccc} a_{11} & a_{12} & \dots & a_{1k-1} & 0 & 0 & \dots & 0 \\ a_{22} & \dots & a_{2k-1} & 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & & & \vdots \\ a_{k-1k-1} & 0 & 0 & \dots & 0 & & & \\ a_{kk} & 0 & \dots & 0 & & & & \\ a_{k+1k+1} & \dots & 0 & & & & & \\ \vdots & & \vdots & & & & & \\ a_{nn} & & & & & & & \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^{(n-k+1)} \\ b_2^{(n-k+1)} \\ \vdots \\ b_{k-1}^{(n-k+1)} \\ b_k^{(n-k)} \\ b_{k+1}^{(n-k-1)} \\ \vdots \\ b_n \end{pmatrix}$$

算法 5.2 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

```

数据科学与工程数学基础初稿

正交线性方程组

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 $x = A^T b$ 计算 $x = A^{-1} b$, 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 $x = A^{-1} b$ 的效率可以超过 $2n^2$ 。例如, 如果 A 具有 $A = I - 2uu^T$ 的形式, 其中 $\|u\|_2 = 1$, 此时

$$x = A^{-1} b = (I - 2uu^T)^T b = b - 2(u^T b)u$$

我们可以先计算 $u^T b$, 然后计算 $b - 2(u^T b)u$, 其计算成本为 $4n$ 次浮点运算。

置换线性方程组

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。

如果 A 是排列矩阵，求解 $Ax = b$ 将非常容易，用 π^{-1} 对 b 元素进行排列就可以得到 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 $x = A^T b$ 可以达到同样的结论。矩阵 A^T （像 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。

5.1.4 基于矩阵分解的方阵系统的直接解法

本小节从矩阵分解的角度探讨线性方程组的直接解法。我们首先讨论的一类特殊的方阵线性方程组

$$Ax = b, A \in \mathbb{R}^{n \times n}$$

的求解，其中 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 $Ax = b$ 的基本途径是将 A 表示为一系列非奇异矩阵的乘积

$$A = A_1 A_2 \dots A_k$$

因此

$$x = A^{-1}b = A_k^{-1} A_{k-1}^{-1} \dots A_1^{-1} b$$

我们可以从右到左利用这个公式计算 x ：

$$z_1 := A_1^{-1} b$$

$$z_2 := A_2^{-1} z_1 = A_2^{-1} A_1^{-1} b$$

⋮

$$z_{k-1} := A_{k-1}^{-1} z_{k-2} = A_{k-1}^{-1} \dots A_1^{-1} b$$

$$x := A_k^{-1} z_{k-1} = A_k^{-1} \dots A_1^{-1} b$$

这个过程的第 i 步需要计算 $z_i = A_i^{-1} z_{i-1}$ 即求解线性方程组 $A_i z_i = z_{i-1}$ 。如果这些方程组都容易求解（即如果 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 $x = A^{-1}b$ 的一种方法。将 A 表示为因式分解形式（即计算 $A = A_1 A_2 \dots A_k$ ）的步骤被称为

矩阵分解步骤，而通过递推求解一系列 $A_i z_i = z_{i-1}$ 来计算 $x = A^{-1}b$ 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 $Ax = b$ 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 $Ax = b$ 的成本，即计算 $x = A^{-1}b$ 就是 f 。

基于 LU 分解求解线性方程组

设矩阵 A 有 LU 分解 $A = PLU$ ，其中 P 是排列矩阵， L 是下三角矩阵， U 是上三角矩阵。这种形式被称为 A 的 LU 因式分解。我们也可以把因式分解写成 $P^T A = LU$ ，其中矩阵 $P^T A$ 通过重排列 A 的行向量得到。那么我们在求解方程组

$$Ax = b$$

时，等价求解一系列如下方程组

$$Pz_1 = b, Lz_2 = z_1, Ux = z_2$$

对于第一个方程，我们只需要根据其排列规则来将 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。

算法 5.3 利用 LU 因式分解求解线性方程组

- 1: LU 因式分解。将 A 因式分解为 $A = PLU$ ($(2/3)n^3$ 次浮点运算)。
- 2: 排列。求解 $Pz_1 = b$ (0 次浮点运算)。
- 3: 前向代入。求解 $Lz_2 = z_1$ (n^2 次浮点运算)。
- 4: 后向代入。求解 $Ux = z_2$ (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 A 和向量 b ，我们先对 A 进行 LU 分解。并且使用 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。

算法 5.4 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 5.1.6. 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，得

$$\hat{\mathbf{y}} = \begin{pmatrix} 0 \\ -2 \\ -1 \end{pmatrix}$$

最后进行回代法，便得解 $(1, -1, 1)^T$ 。

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 $\mathbf{A} = \mathbf{L}\mathbf{L}^T$ ，其中 \mathbf{L} 是下三角矩阵。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。

算法 5.5 基于 Cholesky 分解求解对称正定线性方程组

- 1: Cholesky 因式分解。将 A 因式分解为 $A = LL^T$ ($(1/3)n^3$ 次浮点运算)。
- 2: 前向代入。求解 $Lz_1 = b$ (n^2 次浮点运算)。
- 3: 后向代入。求解 $L^T x = z_1$ (n^2 次浮点运算)。

例 5.1.7. 求解 $\begin{pmatrix} 4 & -2 & 0 \\ -2 & 2 & 2 \\ 0 & 2 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -6 \\ 8 \\ 12 \end{pmatrix}$

解. 先将矩阵 $\begin{pmatrix} 4 & -2 & 0 \\ -2 & 2 & 2 \\ 0 & 2 & 5 \end{pmatrix}$ 进行 Cholesky 分解得

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

先利用前代法, 得 $\hat{y} = \begin{pmatrix} -3 \\ 5 \\ 2 \end{pmatrix}$ 。最终利用回代法, 得到解 $(-1, 1, 2)^T$

基于 QR 分解求解线性方程组

设可逆矩阵 $A \in \mathbb{R}^{n \times n}$ 有 QR 分解 $A = QR$ 其中 Q 是正交矩阵 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 $Ax = b$ 时, 等价求解方程组 $Rx = Q^T b$ 。对于这个方程组, 我们可以使用回代法来求解。

算法 5.6 基于 QR 分解求解线性方程组

- 1: QR 因式分解。将 A 因式分解为 $A = QR$ ($4n^3$ 次浮点运算)。
- 2: 矩阵-向量乘法。求解 $z = Q^T b$ ($2n^2$ 次浮点运算)。
- 3: 后向代入。求解 $Rx = z$ (n^2 次浮点运算)。

例 5.1.8. 求解 $\begin{pmatrix} 1 & 1 & -3 \\ -1 & 3 & -3 \\ 0 & 2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -3 \\ 2 \end{pmatrix}$

解. 先对矩阵 $\begin{pmatrix} 1 & 1 & -3 \\ -1 & 3 & -3 \\ 0 & 2 & 0 \end{pmatrix}$ 做 QR 分解, 得

$$\begin{pmatrix} 1 & 1 & -3 \\ -1 & 3 & -3 \\ 0 & 2 & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} & \end{pmatrix} \begin{pmatrix} \sqrt{2} & -\sqrt{2} & \\ 2\sqrt{3} & -2\sqrt{3} & \\ & \sqrt{6} & \end{pmatrix}$$

那么原问题等价于

$$\begin{pmatrix} \sqrt{2} & -\sqrt{2} & \\ 2\sqrt{3} & -2\sqrt{3} & \\ & \sqrt{6} & \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2\sqrt{2} \\ 0 \\ \sqrt{6} \end{pmatrix}$$

解得答案为 $\begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}$ 。

基于 SVD 求解线性方程组

设矩阵 $A \in \mathbb{R}^{n \times n}$ 的奇异值分解为 $A = U\Sigma V^T$, 其中 U, 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 $Ax = b$ 时, 等价求解一系列如下方程组 $Uy = b$, $\Sigma z = y$, $V^T x = z$ 。而这些方程对应的解为 $y = U^T b$, $z = \Sigma^{-1} y$, $x = Vz$ 。

算法 5.7 基于 SVD 求解线性方程组

- 1: **SVD 因式分解**。将 A 因式分解为 $A = U\Sigma V^T$ (n^3 次浮点运算)。
- 2: **矩阵-向量乘法**。求解 $Uy = b$ ($2n^2$ 次浮点运算)。
- 3: **求解对角方程组**。求解 $\Sigma z = y$ (n 次浮点运算)。
- 4: **矩阵-向量乘法**。求解 $V^T x = z$ ($2n^2$ 次浮点运算)。

例 5.1.9. 求解 $\begin{pmatrix} 1 & 5 & 5 \\ -5 & 1 & 3 \\ 5 & -3 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -3 \\ 7 \end{pmatrix}$

解. 先对矩阵 $\begin{pmatrix} 1 & 5 & 5 \\ -5 & 1 & 3 \\ 5 & -3 & -1 \end{pmatrix}$ 做 SVD 分解得

$$\begin{pmatrix} 1 & 5 & 5 \\ -5 & 1 & 3 \\ 5 & -3 & -1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 9 & & \\ 6 & & \\ 2 & & \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}$$

方程组等价于

$$\begin{pmatrix} 9 & & \\ 6 & & \\ 2 & & \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -3\sqrt{3} \\ 2\sqrt{6} \\ 2\sqrt{2} \end{pmatrix}$$

即

$$\begin{pmatrix} -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -\frac{\sqrt{3}}{3} \\ \frac{\sqrt{6}}{3} \\ \frac{\sqrt{2}}{3} \end{pmatrix}$$

最后解得答案为 $\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$ 。

5.1.5 非方阵系统的直接求解方法

上面考虑了方阵系统，我们接下来考虑非方阵系统

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m$$

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 $A_1 \in \mathbb{R}^{m \times m}$ 是非奇异矩阵。我们可以将 x_1 表示成

$$x_1 = A_1^{-1}(b - A_2 x_2) = A_1^{-1}b - A_1^{-1}A_2 x_2$$

该表达式让我们能很容易地计算一个解：简单取 $\hat{x}_2 = 0$, $\hat{x}_1 = A_1^{-1}b$ 。其计算成本等于求解 m 个线性方程组 $A_1 \hat{x}_1 = b$ 的成本。我们也可以用 $x_2 \in \mathbb{R}^{n-m}$ 做自由参数表示 $Ax = b$ 的所有解。方程 $Ax = b$ 的一般性解可以表示成

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -A_1^{-1}A_2 \\ I \end{pmatrix} x_2 + \begin{pmatrix} A_1^{-1}b \\ 0 \end{pmatrix}$$

综上所述，假设 A_1 的因式分解成本是 f , 而求解形如 $A_1 x = d$ 的系统成本为 s , 那么找出 $Ax = b$ 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 A 的前 m 个列向量不一定线性独立。因为 $\text{rank}(A) = m$, 我们可以选出 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 P 使 $\tilde{A} = AP$ 的前 m 个列向量线性无关，即

$$\tilde{A} = AP = (A_1, A_2)$$

其中 A_1 可逆。方程 $\tilde{A}\tilde{x} = b$ 其中 $\tilde{x} = P^T x$, 其一般解

$$\tilde{x} = \begin{pmatrix} -A_1^{-1}A_2 \\ I \end{pmatrix} \tilde{x}_2 + \begin{pmatrix} A_1^{-1}b \\ 0 \end{pmatrix}$$

于是 $Ax = b$ 的一般解为

$$x = P\tilde{x} = P \begin{pmatrix} -A_1^{-1}A_2 \\ I \end{pmatrix} z + P \begin{pmatrix} A_1^{-1}b \\ 0 \end{pmatrix}$$

其中 $z \in \mathbb{R}^{n-m}$ 是自由参数。该想法可用于容易发现 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

$$A = (Q_1, Q_2) \begin{pmatrix} R \\ O \end{pmatrix}$$

其中 (Q_1, Q_2) 是正交矩阵, $R \in \mathbb{R}^{m \times m}$ 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 Q 能够有效计算乘积 Qx 和 $Q^T x$)。

算法 5.8 QR 因式分解求列满秩非方阵系统

- 1: **QR 因式分解。** 将 A^T 因式分解为 $A^T = (Q_1, Q_2) \begin{pmatrix} R \\ O \end{pmatrix}$ ($2m^2(n - m/3)$ 次浮点运算)。
- 2: **矩阵-向量乘法。** 求解 $Q_1 y = b$ ($2n^2$ 次浮点运算)。
- 3: **后向代入。** 求解 $Rx = y$ (n^2 次浮点运算)。

例 5.1.10. 求解 $\begin{pmatrix} 1 & 1 \\ -1 & 3 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ 2 \end{pmatrix}$

解. 先对矩阵 $\begin{pmatrix} 1 & 1 \\ -1 & 3 \\ 0 & 2 \end{pmatrix}$ 做 QR 分解, 得

$$\begin{pmatrix} 1 & 1 \\ -1 & 3 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} & 0 \end{pmatrix} \begin{pmatrix} \sqrt{2} & -\sqrt{2} \\ 0 & 2\sqrt{3} \\ 0 & 0 \end{pmatrix}$$

那么原问题等价于

$$\begin{pmatrix} \sqrt{2} & -\sqrt{2} \\ 0 & 2\sqrt{3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2\sqrt{2} \\ 2\sqrt{3} \end{pmatrix}$$

解得答案为 $\begin{pmatrix} 3 \\ 1 \end{pmatrix}$ 。

QR 因式分解可以用来解方程组 $\mathbf{A}\mathbf{x} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m < n$ 。假设

$$\mathbf{A}^T = (\mathbf{Q}_1, \mathbf{Q}_2) \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$$

是 \mathbf{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:

$$\mathbf{A}\tilde{\mathbf{x}} = \mathbf{R}^T \mathbf{Q}_1^T \mathbf{Q}_1 \mathbf{R}^{-T} \mathbf{b} = \mathbf{b}$$

此外, \mathbf{Q}_2 的列向量构成 \mathbf{A} 的零空间的基, 于是所有的解可以参数化为

$$\{ \mathbf{x} = \tilde{\mathbf{x}} + \mathbf{Q}_2 \mathbf{z} \mid \mathbf{z} \in \mathbb{R}^{n-m} \}$$

QR 因式分解方法是求解非方阵方程组最常用方法。

算法 5.9 QR 因式分解求行满秩非方阵系统

- 1: **QR 因式分解。** 将 \mathbf{A}^T 因式分解为 $\mathbf{A}^T = (\mathbf{Q}_1, \mathbf{Q}_2) \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$ ($2m^2(n - m/3)$ 次浮点运算)。
- 2: **前向代入。** 求解 $\mathbf{R}\mathbf{y} = \mathbf{b}$ (n^2 次浮点运算)。
- 3: **矩阵-向量乘法。** 求解 $\mathbf{Q}_1 \mathbf{x} = \mathbf{y}$ ($2n^2$ 次浮点运算)。

例 5.1.11. 求解 $\begin{pmatrix} 1 & -1 & 0 \\ 1 & 3 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$

解. 先对矩阵 $\begin{pmatrix} 1 & 1 \\ -1 & 3 \\ 0 & 2 \end{pmatrix}$ 做 QR 分解, 得

$$\begin{pmatrix} 1 & 1 \\ -1 & 3 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} & 0 \end{pmatrix} \begin{pmatrix} \sqrt{2} & -\sqrt{2} \\ 0 & 2\sqrt{3} \\ 0 & 0 \end{pmatrix}$$

那么原问题等价于

$$\begin{pmatrix} \sqrt{2} & -\sqrt{2} \\ 0 & 2\sqrt{3} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} & 0 \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

解得答案为 $\frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ 。所以方程组的解集为 $x = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \frac{1}{\sqrt{6}} \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix} z, z \in \mathbb{Z}$

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, $L \in \mathbb{R}^{n \times n}$ 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。

算法 5.10 基于 LU 分解求解列满秩非方阵系统

- 1: LU 因式分解。将 A 因式分解为 $A = P_1 L U P_2$ ($(2/3)m^3 + m^2(n - m)$ 次浮点运算)。
- 2: 排列。求解 $P_1 z_1 = b$ (0 次浮点运算)。
- 3: 前向代入。求解 $L z_2 = z_1$ (n^2 次浮点运算)。
- 4: 后向代入。求解 $U z_3 = z_2$ (n^2 次浮点运算)。
- 5: 排列。求解 $P_2 x = z_3$ (0 次浮点运算)。

例 5.1.12. 求解 $\begin{pmatrix} 1 & -1 \\ 0 & 2 \\ -2 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ -3 \end{pmatrix}$

解. 先将矩阵 $\begin{pmatrix} 1 & -1 \\ 0 & 2 \\ -2 & 6 \end{pmatrix}$ 进行 LU 分解得 $\begin{pmatrix} 1 & -1 \\ 0 & 2 \\ -2 & 6 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ -2 & 2 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 0 & 2 \end{pmatrix}$ 先利用前代法, 得 $\hat{y} = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}$ 。最终利用回带法, 得到解 $(1, 1)^T$

假设 $A^T = PLU$ 是方程组 $Ax = b$, $A \in \mathbb{R}^{m \times n}$, $m < n$ 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

$$x = P \begin{pmatrix} -L_1^{-T} L_2^T \\ I \end{pmatrix} z + P(L_1^{-T} U^{-T} b)$$

其中 $z \in \mathbb{R}^{n-m}$ 。

算法 5.11 基于 QR 分解求解行满秩非方阵系统

- 1: QR 因式分解。将 A^T 因式分解为 $A = P \begin{pmatrix} L_1 \\ L_2 \end{pmatrix}$ ($(2/3)m^3 + m^2(n-m)$ 次浮点运算)。
- 2: 前向代入。求解 $L_1^T z_1 = b$ (n^2 次浮点运算)。
- 3: 前向代入。求解 $L_1^T Z_1 = L_2$ (n^2 次浮点运算)。
- 4: 后向代入。求解 $U^T z_2 = z_1$ (n^2 次浮点运算)。
- 5: 排列。求解 $Pz_3 = z_2$ (0 次浮点运算)。
- 6: 排列。求解 $PZ_2 = Z_1$ (0 次浮点运算)。

基于奇异值分解的非方阵系统求解方法

考虑非方阵系统 $Ax = b$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, 我们可以计算 A 的奇异值分解。设 $A = U\tilde{\Sigma}V^T$, 记 $\tilde{x} = V^T x$, $\tilde{b} = U^T b$, 那么我们就得到了一个关于对角矩阵的方程组 $\tilde{\Sigma}\tilde{x} = \tilde{b}$ 。其中 \tilde{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{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 b 不在 A 的列空间中。
- 而当 b 在 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{x} 中后 $n - r$ 个分量可以取任意值。如果 A 是一个列满秩矩阵（即他的零空间为 $\{0\}$ ），那么我们就会有唯一解。

我们一旦计算得到了 \tilde{x} 那么就可以通过 $x = V\tilde{x}$ 来得到方程的解。

算法 5.12 基于 SVD 求解非方阵系统

- 1: **SVD 因式分解**。将 A 因式分解为 $A = U\Sigma V^T$ (n^3 次浮点运算)。
- 2: **矩阵-向量乘法**。求解 $Uy = b$ ($2n^2$ 次浮点运算)。
- 3: **求解对角方程组**。求解 $\Sigma z = y$ (n 次浮点运算)。
- 4: **矩阵-向量乘法**。求解 $V^T x = z$ ($2n^2$ 次浮点运算)。

例 5.1.13. 求解 $\begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$

解. 先对矩阵 $\begin{pmatrix} 1 & 5 & 5 \\ -5 & 1 & 3 \\ 5 & -3 & -1 \end{pmatrix}$ 做 SVD 分解得

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix} = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ \frac{2}{\sqrt{6}} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

方程组等价于

$$\begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ -\sqrt{2} \\ 0 \end{pmatrix}$$

即

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ -\sqrt{2} \\ 0 \end{pmatrix}$$

最后解得答案为 $\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ 。

5.1.6 敏度分析与其他方法

考虑如下两组线性方程组：

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix}, \quad \begin{bmatrix} 2 & 0 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix} \quad (5.17)$$

的解都是 $\mathbf{x} = (1, 1)^T$ 。

但是如果我们对方程的常数项做一点微小的变动，求解方程组

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \quad \begin{bmatrix} 2 & 0 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad (5.18)$$

前者的解为 $\mathbf{x} = (2, 0)^T$ ，而后来的解为 $\mathbf{x} = (1, \frac{10000}{10001})^T \approx (1, 0.9999)^T$ 。

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。

我们将分别讨论输入的扰动对解的影响，系数矩阵的扰动对解的影响，以及输入和系数矩阵联合扰动对解的影响。

输入的扰动敏感性

令 \mathbf{x} 为线性方程 $\mathbf{Ax} = \mathbf{b}$ 的解，其中 \mathbf{A} 为非奇异方阵，且 $\mathbf{b} \neq 0$ 。假设我们通过向它添加一个小的扰动项 $\Delta\mathbf{b}$ 来略微改变 \mathbf{b} ，并将 $\mathbf{x} + \Delta\mathbf{x}$ 称为扰动方程组的解：

$$\mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}$$

关键问题是：如果 $\Delta\mathbf{b}$ 变小， $\Delta\mathbf{x}$ 将会不会变小？从上面的公式看出，并且从 $\mathbf{Ax} = \mathbf{b}$ 的事实看，扰动 $\Delta\mathbf{x}$ 本身就是线性方程组 $\mathbf{A}\Delta\mathbf{x} = \Delta\mathbf{b}$ 的解。并且，由于认为 \mathbf{A} 是可逆的，我们可以写成 $\Delta\mathbf{x} = \mathbf{A}^{-1}\Delta\mathbf{b}$ 。采用该方程两边的 l_2 范数得 $\|\Delta\mathbf{x}\|_2 = \|\mathbf{A}^{-1}\Delta\mathbf{b}\|_2 \leq \|\mathbf{A}^{-1}\|_2 \|\Delta\mathbf{b}\|_2$ 。其中 $\|\mathbf{A}^{-1}\|_2$ 是 \mathbf{A}^{-1} 的谱(最大奇异值)范数。类似地，从 $\mathbf{Ax} = \mathbf{b}$ 得出 $\|\mathbf{b}\|_2 = \|\mathbf{Ax}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|_2$ ，因此 $\|\mathbf{x}\|_2 \leq \frac{\|\mathbf{A}\|_2}{\|\mathbf{b}\|_2} \|\mathbf{b}\|_2$ 。将上面两个公式相乘，得到

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \|\mathbf{A}^{-1}\|_2 \|\mathbf{A}\|_2 \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2}$$

这个结果将“输入项” \mathbf{b} 的相对变化与“输出” \mathbf{x} 的相对变化联系起来了。

定义 5.1.6. 设 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 是可逆矩阵，称数

$$\kappa(\mathbf{A}) = \|\mathbf{A}^{-1}\|_2 \|\mathbf{A}\|_2,$$

是矩阵 \mathbf{A} 的条件数。

设 σ_1, σ_n 分别是矩阵 \mathbf{A} 的最大奇异值和最小奇异值，那么

$$\|\mathbf{A}\|_2 = \sigma_1, \quad \|\mathbf{A}^{-1}\|_2 = 1/\sigma_n$$

因此矩阵 \mathbf{A} 的条件数也可以定义为：

$$\kappa(\mathbf{A}) = \frac{\sigma_1}{\sigma_n}, 1 \leq \kappa(\mathbf{A}) \leq \infty.$$

大的 $\kappa(\mathbf{A})$ 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 \mathbf{A} 是奇异的，那么 $\kappa = \infty$ 。非常大的 $\kappa(\mathbf{A})$ 表明 \mathbf{A} 接近奇异；我们说在这种情况下 \mathbf{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 \mathbf{A} 为非奇异方阵， $\mathbf{x}, \Delta\mathbf{x}$ 满足

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$\mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}$$

然后它认为

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2}$$

其中 $\kappa(\mathbf{A}) = \|\mathbf{A}^{-1}\|_2 \|\mathbf{A}\|_2$ 是矩阵 \mathbf{A} 的条件数

系数矩阵中的扰动敏感性

接下来我们考虑 \mathbf{A} 矩阵的扰动对 \mathbf{x} 的影响。令 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 并且令 $\Delta\mathbf{A}$ 为一个扰动，满足下面等式

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b}, \quad \text{对于一些 } \Delta\mathbf{x}$$

那么有 $\mathbf{A}\Delta\mathbf{x} = -\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})$ ，因此 $\Delta\mathbf{x} = -\mathbf{A}^{-1}\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})$ 。则

$$\|\Delta\mathbf{x}\|_2 = \|\mathbf{A}^{-1}\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})\|_2 \leq \|\mathbf{A}^{-1}\|_2 \|\Delta\mathbf{A}\|_2 \|\mathbf{x} + \Delta\mathbf{x}\|_2$$

并且

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} \leq \|\mathbf{A}^{-1}\|_2 \|\mathbf{A}\|_2 \frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2}$$

我们再次看到只有在条件数不是太大时，小扰动 $\frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2} \ll 1$ 对 \mathbf{x} 的相对影响才很小。就是说，它离 1 不太远， $\kappa(\mathbf{A}) \simeq 1$ 。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 \mathbf{A} 为非奇异方阵， $\mathbf{x}, \Delta\mathbf{A}, \Delta\mathbf{x}$ 满足

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b}$$

那么

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} \leq \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2}$$

对 \mathbf{A}, \mathbf{b} 联合扰动的敏感性

我们最后考虑了 \mathbf{A} 和 \mathbf{b} 的同时扰动对 \mathbf{x} 的影响。令 $\mathbf{Ax} = \mathbf{b}$, 并且令 $\Delta\mathbf{A}, \Delta\mathbf{b}$ 为扰动, 满足下面等式

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}, \text{ 对于一些 } \Delta\mathbf{x}$$

然后, $\mathbf{A}\Delta\mathbf{x} = \Delta\mathbf{b} - \Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})$, 因此 $\Delta\mathbf{x} = \mathbf{A}^{-1}\Delta\mathbf{b} - \mathbf{A}^{-1}\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})$ 。则

$$\begin{aligned} \|\Delta\mathbf{x}\|_2 &= \|\mathbf{A}^{-1}\Delta\mathbf{b} - \mathbf{A}^{-1}\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})\|_2 \\ &\leq \|\mathbf{A}^{-1}\Delta\mathbf{b}\|_2 + \|\mathbf{A}^{-1}\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})\|_2 \\ &\leq \|\mathbf{A}^{-1}\|_2 \|\Delta\mathbf{b}\|_2 + \|\mathbf{A}^{-1}\|_2 \|\Delta\mathbf{A}\|_2 \|\mathbf{x} + \Delta\mathbf{x}\|_2 \end{aligned}$$

接着, 上式除以 $\|\mathbf{x} + \Delta\mathbf{x}\|_2$,

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} \leq \|\mathbf{A}^{-1}\|_2 \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} \frac{\|\mathbf{b}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} + \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2}$$

但是 $\|\mathbf{b}\|_2 = \|\mathbf{Ax}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|_2$, 因此

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} \leq \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} \frac{\|\mathbf{x}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} + \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2}$$

下一步, 我们根据 $\|\mathbf{x}\|_2 = \|\mathbf{x} + \Delta\mathbf{x} - \Delta\mathbf{x}\|_2 \leq \|\mathbf{x} + \Delta\mathbf{x}\|_2 + \|\Delta\mathbf{x}\|_2$ 去写

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} \leq \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} \frac{\|\mathbf{x}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} + \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2}$$

从中我们得到

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} \leq \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} \left(1 + \frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} \right) + \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2}$$

因此

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} \leq \frac{\kappa(\mathbf{A})}{1 - \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2}} \left(\frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} + \frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2} \right)$$

扰动的“放大因子”是受 $\frac{\kappa(\mathbf{A})}{1 - \kappa(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2}}$ 的约束。因此, 该界限小于某些给定的 γ , 如果

$$\kappa(\mathbf{A}) \leq \frac{\gamma}{1 + \gamma \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2}}$$

因此, 我们看到关节扰动的影响仍然由 \mathbf{A} 的条件数控制, 如下所述

引理 5.1.3. (对 A, b 扰动的敏感性) 令 \mathbf{A} 为非奇异方阵, 令 $\gamma > 1$ 已知, 并且令 $\mathbf{x}, \Delta\mathbf{b}, \Delta\mathbf{A}, \Delta\mathbf{x}$ 满足下面等式

$$\mathbf{Ax} = \mathbf{b}$$

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b}$$

且

$$\kappa(\mathbf{A}) \leq \frac{\gamma}{1 + \gamma \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2}}$$

那么

$$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x} + \Delta\mathbf{x}\|_2} \leq \gamma \left(\frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} + \frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2} \right)$$

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 $r(\mathbf{x}_i; \mathbf{w}, b)$:

$$\mathbf{r} = \begin{pmatrix} y_1 - (w_1 x_{11} + w_2 x_{12} + \dots + w_n x_{1n} + b) \\ y_2 - (w_1 x_{21} + w_2 x_{22} + \dots + w_n x_{2n} + b) \\ y_3 - (w_1 x_{31} + w_2 x_{32} + \dots + w_n x_{3n} + b) \\ \vdots \\ y_m - (w_1 x_{m1} + w_2 x_{m2} + \dots + w_n x_{mn} + b) \end{pmatrix}$$

化成矩阵的形式表示为

$$\mathbf{r} = \mathbf{y} - \mathbf{A}\hat{\mathbf{w}},$$

$$\text{其中: } A = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} & 1 \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} & 1 \\ x_{31} & x_{32} & x_{33} & \dots & x_{3n} & 1 \\ \vdots & & & & & \\ x_{m1} & x_{m2} & x_{m3} & \dots & x_{mn} & 1 \end{pmatrix}, \hat{w} = \begin{pmatrix} w \\ b \end{pmatrix}, y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{pmatrix}$$

问题就变为求参数 \hat{w} , 使得残差 r 尽可能的小。若使残差 r 在 l_2 范数意义下最小。也即 $\arg \min_{\hat{w}} \|A\hat{w} - y\|_2$ 把上式中的符号调整成我们常用的符号:

$$\arg \min_x \|Ax - b\|_2 \quad (5.19)$$

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 $b \in \mathbb{R}^m$, 确定 $x \in \mathbb{R}^n$ 使得

$$\|b - Ax\|_2 = \|r(x)\|_2 = \min_{y \in \mathbb{R}^n} \|r(y)\|_2 = \min_{y \in \mathbb{R}^n} \|Ay - b\|_2 \quad (5.20)$$

其中 $r(x) = b - Ax$, 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 x_0 则称为最小二乘解或极小解。

如果残差向量 r 线性依赖于 x , 则称其为线性最小二乘问题; 如果 r 非线性的依赖于 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

$$\mathcal{X}_{LS} = \{x \in \mathbb{R}^n : x \text{ 满足(5.20)}\}.$$

解集 \mathcal{X}_{LS} 中 l_2 范数最小的解称为最小范数解, 记为 x_{LS} , 即

$$\|x_{LS}\|_2 = \min\{\|x\|_2 : x \in \mathcal{X}_{LS}\}.$$

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 l_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:

- $A \in \mathbb{R}^{m \times n}$ 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。
- $A \in \mathbb{R}^{m \times n}$ 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 $x \neq 0$ 使得 $A^T A x = 0$ 。 $x^T A^T A x = 0$, 因此 $A x = 0$ 。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 $x \neq 0$, $A^T A x \neq 0$, 也能推出对于每一个非零的 x , $A x \neq 0$ 。第 2 条性质的证明过程与第 1 条的证明过程相似。□

最小二乘问题的解 x 又称为线性方程组(5.21)的最小二乘解。

$$Ax = b, \quad A \in \mathbb{R}^{m \times n} \quad (5.21)$$

即残差向量 $r(x)$ 的 l_2 范数最小的意义下满足方程组(5.21)。

根据 m 与 n 以及矩阵 A 的秩 $r(A)$ 的不同, 最小二乘问题可分为下面几种情况:

1. $m = n$ 。对应方阵系统，此时如果 $\mathbf{Ax} = \mathbf{b}$ 方程有解，那么方程的解使得 $\|\mathbf{Ax} - \mathbf{b}\|_2$ 最小。
2. $m > n$ 。对应超定方程组或矛盾方程组，在这种情况下，方程常发生无解的情况。
3. $m < n$ 。对应欠定方程组，在这种情况下，方程常发生有无穷多解的情况。

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

定义 5.2.2. 当方程组(5.20)有解时，显然也满足最小二乘问题(5.20)，如何确定 $\mathbf{x}_0 \in \mathbb{R}^n$ ，使得

$$\|\mathbf{x}_0\|_2 = \min_{\mathbf{Ax} = \mathbf{b}} \|\mathbf{x}\|_2$$

称这样的 \mathbf{x}_0 为方程组(5.20)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 l_2 范数的解感兴趣）。

定义 5.2.3. 当方程组(5.20)无解时，此时相应 LS 问题的最小二乘解不是方程组 $\mathbf{Ax} = \mathbf{b}$ 的解，如何确定 $\mathbf{x}_0 \in \mathbb{R}^n$ ，使得

$$\|\mathbf{x}_0\|_2 = \min_{\min_{\|\mathbf{Ax} - \mathbf{b}\|_2}} \|\mathbf{x}\|_2$$

称这样的 \mathbf{x}_0 为方程组(5.20)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 l_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 5.2.1. 如果方程组 $\mathbf{Ax} = \mathbf{b}$ 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 5.2.2. 如果线性方程组 $\mathbf{Ax} = \mathbf{b}$ 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 $\mathbf{Ax} = \mathbf{b}$ 有无数个解并且解的集合是 $\mathbb{S}_x = \{\mathbf{x} : \mathbf{x} = \tilde{\mathbf{x}} + \mathbf{z}, \mathbf{z} \in \text{Null}(\mathbf{A})\}$ ，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 l_2 范数最小的解 \mathbf{x}^* 。也即求解：

$$\min_{\mathbf{x}: \mathbf{Ax} = \mathbf{b}} \|\mathbf{x}\|_2$$

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 $\mathbf{Ax}^* = \mathbf{b}$ ，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：

$$\mathbf{x}^* = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{b}. \quad (5.22)$$

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 5.2.3. 设 $A \in \mathbb{R}^{m \times n}$, $m \leq n$ 是行满秩的, 并且令 $\mathbf{b} \in \mathbb{R}^m$ 。在线性方程组 $A\mathbf{x} = \mathbf{b}$ 的所有解中, 存在唯一的 l_2 范数最小的解, 这个解由(5.22)给出。

最小二乘的特征和一般表示

为了说明最小二乘问题解的存在性, 我们验证如下的定理。

定理 5.2.4. 线性最小二乘问题(5.20)的解总是存在的, 而且其解唯一的充分必要条件是 $\text{Null}(A) = 0$ 。

证明. 因为 $\mathbb{R}^m = \text{Col}(A) \oplus \text{Col}(A)^\perp$, 所以向量 \mathbf{b} 可以唯一地表示为 $\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2$, 其中 $\mathbf{b}_1 \in \text{Col}(A)$, $\mathbf{b}_2 \in \text{Col}(A)^\perp$ 。于是对于任意 $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b}_1 - A\mathbf{x} \in \text{Col}(A)$ 且与 \mathbf{b}_2 正交, 从而

$$\|\mathbf{r}(\mathbf{x})\|_2^2 = \|\mathbf{b} - A\mathbf{x}\|_2^2 = \|(\mathbf{b}_1 - A\mathbf{x}) + \mathbf{b}_2\|_2^2 = \|\mathbf{b}_1 - A\mathbf{x}\|_2^2 + \|\mathbf{b}_2\|_2^2$$

由此即知, $\|\mathbf{r}(\mathbf{x})\|_2^2$ 达到极小当且仅当 $\|\mathbf{b}_1 - A\mathbf{x}\|_2^2$ 达到极小;

而 $\mathbf{b}_1 \in \text{Col}(A)$ 又蕴涵着 $\|\mathbf{b}_1 - A\mathbf{x}\|_2^2$ 达到极小的充分与必要条件是

$$A\mathbf{x} = \mathbf{b}_1$$

这样, 由 $\mathbf{b}_1 \in \text{Col}(A)$ 和 $A\mathbf{x} = \mathbf{b}_1$ 有唯一解的充要条件是 $\text{nullity}(A) = 0$ 。

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乘问题的方法。

定理 5.2.5. $\mathbf{x} \in \mathcal{X}_{LS}$ 当且仅当

$$A^T A \mathbf{x} = A^T \mathbf{b} \tag{5.23}$$

其中方程组(5.23)称为最小二乘问题的正规化方程组或法方程组。

证明. 设 $\mathbf{x} \in \mathcal{X}_{LS}$ 。由定理5.2.4证明知 $A\mathbf{x} = \mathbf{b}_1$, 其中 $\mathbf{b}_1 \in \text{Col}(A)$, 而且

$$\mathbf{r}(\mathbf{x}) = \mathbf{b} - A\mathbf{x} = \mathbf{b} - \mathbf{b}_1 = \mathbf{b}_2 \in \text{Col}(A)^\perp$$

因而 $A^T \mathbf{r}(\mathbf{x}) = A^T \mathbf{b}_2 = 0$ 。将 $\mathbf{r}(\mathbf{x}) = \mathbf{b} - A\mathbf{x}$ 代入 $A^T \mathbf{r}(\mathbf{x}) = 0$ 即得(5.23)。反之, 设 $\mathbf{x} \in \mathbb{R}^n$ 满足 $A^T A \mathbf{x} = A^T \mathbf{b}$, 则对任意的 $\mathbf{z} \in \mathbb{R}^n$ 有

$$\begin{aligned} \|\mathbf{b} - A(\mathbf{x} + \mathbf{z})\|_2^2 &= \|\mathbf{b} - A\mathbf{x}\|_2^2 - 2\mathbf{z}^T A^T (\mathbf{b} - A\mathbf{x}) + \|A\mathbf{z}\|_2^2 \\ &= \|\mathbf{b} - A\mathbf{x}\|_2^2 + \|A\mathbf{z}\|_2^2 \geq \|\mathbf{b} - A\mathbf{x}\|_2^2 \end{aligned}$$

由此即得 $\mathbf{x} \in \mathcal{X}_{LS}$ 。 \square

由定理5.2.5可知, 可以通过求解正规化方程组或法方程组 $A^T A \mathbf{x} = A^T \mathbf{b}$ 来求解 $A\mathbf{x} = \mathbf{b}$ 的最小二乘解。如果 $A^T A$ 可逆, 那么最小二乘解为 $\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b}$ 。

推论 5.2.1. 若矩阵 A 列满秩, 则线性最小二乘问题(5.20)的解是唯一的, 并且解为

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b} = A^\dagger \mathbf{b},$$

其中 $A^\dagger = (A^T A)^{-1} A^T$ 。

如果 A 既不是列满秩也不是行满秩, 它的最小二乘解仍是方程:

$$A^T A \mathbf{x} = A^T \mathbf{b}$$

的解, 但是, $A^T A$ 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。

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 $\mathbf{r} = A\mathbf{x} - \mathbf{b}$ 在 l_2 范数意义下最小的解。
- **在 $\text{Col}(A)$ 上的投影:** 最小二乘问题的解, 是 \mathbf{b} 在 $\text{Col}(A)$ 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 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 $A\mathbf{x} = \mathbf{b}$ 右侧添加在 l_2 范数意义下的最小扰动项 $\delta\mathbf{b}$ 后 $A\mathbf{x} = \mathbf{b} + \delta\mathbf{b}$ 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 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 A 为列满秩的情形下超定方程组

$$A\mathbf{x} = \mathbf{b}$$

的最小二乘解的求解方法: 此时, $A^T A$ 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。

正规化方法 (Cholesky 分解法)

方程组 $A^T A\mathbf{x} = A^T \mathbf{b}$ 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 A 的列向量线性无关的条件下, $A^T A$ 对称正定, 故可用平方根法求解(5.23)。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 $C = A^T A, d = A^T b$ 。
2. 用平方根法计算 C 的 Cholesky 分解: $C = LL^T$ 。
3. 求解三角方程组 $Ly = d$ 和 $L^T x = y$ 。

注意, 正规化方程组 $A^T A x = A^T b$ 的解 x 可以表示为

$$x = (A^T A)^{-1} A^T b = A^\dagger b$$

算法 5.13 正规化方法 (Cholesky 分解法) 求解最小二乘问题

- 1: 矩阵-矩阵乘法。求解 $C = A^T A$ ($2n^3$ 次浮点运算)。
- 2: 矩阵-向量乘法。求解 $d = A^T b$ ($2n^2$ 次浮点运算)。
- 3: Cholesky 因式分解。将 C 因式分解为 $C = LL^T$ ($(1/3)n^3$ 次浮点运算)。
- 4: 前向代入。求解 $Ly = d$ (n^2 次浮点运算)。
- 5: 后向代入。求解 $L^T x = y$ (n^2 次浮点运算)。

例 5.2.1. 利用正规化方法求 $Ax = b$ 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.

$$A^T A = \begin{pmatrix} 4 & 4 & 8 \\ 4 & 40 & 20 \\ 8 & 20 & 36 \end{pmatrix}, A^T b = \begin{pmatrix} 4 \\ 4 \\ 24 \end{pmatrix}$$

对 $A^T A$ 做 Cholesky 分解:

$$A^T A = \begin{pmatrix} 2 & 0 & 0 \\ 2 & 6 & 0 \\ 4 & 2 & 4 \end{pmatrix} \begin{pmatrix} 2 & 2 & 4 \\ 0 & 6 & 2 \\ 0 & 0 & 4 \end{pmatrix}$$

解方程

$$\begin{pmatrix} 2 & 0 & 0 \\ 2 & 6 & 0 \\ 4 & 2 & 4 \end{pmatrix} y = \begin{pmatrix} 4 \\ 4 \\ 24 \end{pmatrix}, \quad y = \begin{pmatrix} 2 \\ 0 \\ 4 \end{pmatrix}$$

再解方程

$$\begin{pmatrix} 2 & 2 & 4 \\ 0 & 6 & 2 \\ 0 & 0 & 4 \end{pmatrix} x^* = \begin{pmatrix} 2 \\ 0 \\ 4 \end{pmatrix} \quad \text{得 } x^* = \begin{pmatrix} -2/3 \\ -1/3 \\ 1 \end{pmatrix}$$

QR 分解法

由 l_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 $\mathbf{R}\mathbf{x} = \mathbf{b}_1$ 。

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 \mathbf{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。

算法 5.14 基于 QR 分解求解最小二乘问题

1: QR 因式分解。将 \mathbf{A} 因式分解为 $\mathbf{A} = (\mathbf{Q}_1, \mathbf{Q}_2) \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$ ($2mn^2 - \frac{2}{3}n^3$ 次浮点运算)。

2: 矩阵-向量乘法。求解 $\mathbf{z} = \mathbf{Q}_1^T \mathbf{b}$ ($2n^2$ 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)。

例 5.2.2. 利用 QR 分解求 $\mathbf{A}\mathbf{x} = \mathbf{b}$ 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. 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 $Rx^* = b^*$ 得

$$x^* = \begin{pmatrix} -2/3 \\ -1/3 \\ 1 \end{pmatrix}$$

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, $A = U \begin{pmatrix} \Sigma \\ O \end{pmatrix} V^T$ 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 l_2 范数的正交不变性得

$$\begin{aligned} \|Ax - b\|_2^2 &= \left\| U \begin{pmatrix} \Sigma \\ O \end{pmatrix} V^T x - b \right\|_2^2 = \left\| \begin{pmatrix} \Sigma \\ O \end{pmatrix} V^T x - \begin{bmatrix} U_n^T \\ \tilde{U}^T \end{bmatrix} b \right\|_2^2 \\ &= \left\| \begin{pmatrix} \Sigma V^T x - U_n^T b \\ -\tilde{U}^T b \end{pmatrix} \right\|_2^2 = \|\Sigma V^T x - U_n^T b\|_2^2 + \|\tilde{U}^T b\|_2^2 \\ &\geq \|\tilde{U}^T b\|_2^2 \end{aligned}$$

等号当且仅当 $\Sigma V^T x - U_n^T b = 0$ 时成立, 即 $x = (\Sigma V^T)^{-1} U_n^T b = V \Sigma^{-1} U_n^T b$ 。

算法 5.15 基于 SVD 求解最小二乘问题

- 1: SVD 因式分解。将 A 因式分解为 $A = U \Sigma V^T$ (n^3 次浮点运算)。
- 2: 矩阵-向量乘法。求解 $Uy = b$ ($2n^2$ 次浮点运算)。
- 3: 求解对角方程组。求解 $\Sigma z = y$ (n 次浮点运算)。
- 4: 矩阵-向量乘法。求解 $V^T x = z$ ($2n^2$ 次浮点运算)。

例 5.2.3. 利用 SVD 分解求 $\mathbf{Ax} = \mathbf{b}$ 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. 我们这里利用 Python 中 NumPy 库中的 `linalg.svd()` 函数计算 SVD, 注意这个函数返回的是 $\mathbf{U}, \Sigma, \mathbf{V}^T$ 。

```
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)
In [7]: print(np.diag(Sigma))
In [8]: print(V)
In [9]: V*np.diag(1./Sigma)*U[:,0:3].T*b
```

5.2.3 最小二乘问题的变体

对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:

$$\|\mathbf{Ax} - \mathbf{y}\|_2^2 = \sum_{i=1}^m r_i^2, \quad r_i = \mathbf{a}_i^T \mathbf{x} - y_i$$

其中 $\mathbf{a}_i^T, i = 1, \dots, m$ 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

$$\mathbf{W} = \text{diag}(w_1, \dots, w_m), \mathbf{A}_w \doteq \mathbf{W}\mathbf{A}, \mathbf{y}_w \doteq \mathbf{W}\mathbf{y}$$

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：

$$\begin{aligned}\hat{\mathbf{x}}_{\text{WLS}} &= (\mathbf{A}_w^T \mathbf{A}_w)^{-1} \mathbf{A}_w^T \mathbf{y}_w \\ &= (\mathbf{A}^T \mathbf{W}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W}^T \mathbf{W} \mathbf{y}\end{aligned}$$

约束最小二乘

考虑带有约束的最小二乘问题

$$\begin{aligned}\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \\ \text{s.t. } \mathbf{B}\mathbf{x} = \mathbf{f}\end{aligned}$$

其中 $\mathbf{B}\mathbf{x} = \mathbf{f}$ 是约束条件。求解需要凸优化知识，在这里只列出解。如果 $\mathbf{A}^T \mathbf{A}$ 非奇异，且 \mathbf{B} 行满秩，则 $\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1}(\mathbf{A}^T \mathbf{b} - \mathbf{B}^T \boldsymbol{\lambda})$ 其中 $\boldsymbol{\lambda} = (\mathbf{B}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{B}^T)^{-1} (\mathbf{B}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} - \mathbf{f})$ 。

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。假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

$$\begin{aligned}\mathbf{b} + \Delta \mathbf{b} &= \mathbf{b}_0 + \mathbf{e} + \Delta \mathbf{b} \rightarrow \mathbf{b}_0 \\ \mathbf{A} + \Delta \mathbf{A} &= \mathbf{A}_0 + \mathbf{E} + \Delta \mathbf{A} \rightarrow \mathbf{A}_0\end{aligned}$$

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

$$(\mathbf{A} + \Delta \mathbf{A})\mathbf{x} = \mathbf{b} + \Delta \mathbf{b} \implies \mathbf{A}_0 \mathbf{x} = \mathbf{b}_0 \quad (5.24)$$

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：

$$\begin{aligned}\text{TLS: } \min_{\Delta \mathbf{A}, \Delta \mathbf{b}, \mathbf{x}} & \|[\Delta \mathbf{A}, \Delta \mathbf{b}]\|_F^2 = \|\Delta \mathbf{A}\|_F^2 + \|\Delta \mathbf{b}\|_2^2 \\ \text{s.t. } & (\mathbf{A} + \Delta \mathbf{A})\mathbf{x} = \mathbf{b} + \Delta \mathbf{b}\end{aligned}$$

约束条件有时也表示为 $(\mathbf{b} + \Delta \mathbf{b}) \in \text{Range}(\mathbf{A} + \Delta \mathbf{A})$ 。

由(5.24)，校正过方程的解满足：

$$([\mathbf{A}, \mathbf{b}] + [\Delta \mathbf{A}, \Delta \mathbf{b}]) \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = \mathbf{0} \quad (5.25)$$

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:

$$([\mathbf{A}, \mathbf{b}] + [\Delta\mathbf{A}, \Delta\mathbf{b}])\hat{\mathbf{x}} = \mathbf{0} \quad (5.26)$$

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

$$\min \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \quad \text{和} \quad \min \|(\mathbf{b} + \Delta\mathbf{b}) - \mathbf{A}\mathbf{x}\|_2$$

的解, 即

$$\begin{aligned} \mathbf{x} &= \mathbf{A}^\dagger \mathbf{b}, \\ \mathbf{x} + \Delta\mathbf{x} &= \mathbf{A}^\dagger (\mathbf{b} + \Delta\mathbf{b}) = \mathbf{A}^\dagger \tilde{\mathbf{b}} \end{aligned}$$

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 5.2.6. 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

$$\|\Delta\mathbf{x}\|_2 = \|\mathbf{A}^\dagger \mathbf{b} - \mathbf{A}^\dagger \tilde{\mathbf{b}}\|_2 = \|\mathbf{A}^\dagger (\mathbf{b}_1 - \tilde{\mathbf{b}}_1)\|_2 \quad (5.27)$$

$$\leq \|\mathbf{A}^\dagger\|_2 \|\mathbf{b}_1 - \tilde{\mathbf{b}}_1\|_2 \quad (5.28)$$

由 $\mathbf{A}\mathbf{x} = \mathbf{b}_1$ 得

$$\|\mathbf{b}_1\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|_2 \quad (5.29)$$

由 (5.28) 和 (5.29) 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 5.2.7. 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。

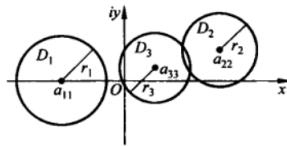
5.3 特征值计算

我们在矩阵的基本特征内容部分已经回顾了特征值、特征向量的概念，并在很多章节对特征值和特征向量的几何意义做了解读。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。

与线性方程组和最小二乘问题的求解一样，矩阵特征值和特征向量的计算也是数值线性代数的重要内容。传统上求一个矩阵的特征值的问题实质上是求一个特征（代数）多项式的根的问题，而数学上已经证明：5 阶以上的多项式的根一般不能用有限次运算求得。因此，矩阵特征值的计算方法本质上都是迭代的。目前，已有不少非常成熟的数值方法用于计算矩阵的全部或部分特征值和特征向量。而全面系统地介绍所有这些重要的数值方法，会远远超出我们这门课的范围，因而这里我们仅介绍几类最常用的基本方法，包括幂法和反幂法等。

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应用，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

$$cond(A)_2 = \sqrt{\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)}}$$

- 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:

(1) 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

$$|\lambda - a_{kk}| \leq \sum_{j=1, j \neq k}^n |a_{kj}| \quad (5.30)$$

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

$$|\lambda - a_{kk}| |x_k| = \left| \sum_{\substack{j=1 \\ j \neq k}}^n a_{kj} x_j \right| \leq \sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}| |x_j| \leq |x_k| \sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}| \quad (5.31)$$

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(5.31)除以 $|x_k|$, 即得到式(5.30), 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。□

例 5.3.1. 试估计矩阵

$$\begin{pmatrix} 4 & 1 & 0 \\ 1 & 9 & -1 \\ 1 & 1 & -4 \end{pmatrix}$$

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

$$D'_1 : |\lambda - 4| \leq 2, \quad D'_2 : |\lambda| \leq 1, \quad D'_3 : |\lambda + 4| \leq 1.$$

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 Python 的 `numpy.eig` 函数可求出矩阵 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 为对角矩阵, 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 $x_0 \in \mathbb{R}^n$, 再进行迭代计算

$$x_k = Ax_{k-1}, k = 1, 2, \dots$$

得到向量序列 $\{x_k\}$, 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 $x^{(0)}$ 均可用 $\xi_1, \xi_2, \dots, \xi_n$ 线性表示, 即

$$x^{(0)} = \alpha_1 \xi_1 + \alpha_2 \xi_2 + \dots + \alpha_n \xi_n$$

且 $\alpha_1, \alpha_2, \dots, \alpha_n$ 不全为零。做向量序列 $x^{(k)} = A^k x^{(0)}$, 则

$$\begin{aligned} x^{(k)} &= A^k x^{(0)} = \alpha_1 A^k \xi_1 + \alpha_2 A^k \xi_2 + \dots + \alpha_n A^k \xi_n \\ &= \alpha_1 \lambda_1^k \xi_1 + \alpha_2 \lambda_2^k \xi_2 + \dots + \alpha_n \lambda_n^k \xi_n \\ &= \lambda_1^k [\alpha_1 \xi_1 + \alpha_2 (\frac{\lambda_2}{\lambda_1})^k \xi_2 + \dots + \alpha_n (\frac{\lambda_n}{\lambda_1})^k \xi_n] \end{aligned}$$

由此可见, 若 $\alpha_1 \neq 0$, 则有

$$\lim_{k \rightarrow \infty} \left(\frac{\lambda_i}{\lambda_1} \right)^k = 0, i = 2, \dots, n$$

故当 k 充分大的时候, 必有

$$x^{(k)} \approx \lambda_1^k \alpha_1 \xi_1$$

即 $x^{(k)}$ 可以近似看成 λ_1 对应的特征向量, 而 $x^{(k)}$ 与 $x^{(k-1)}$ 分量之比为

$$\frac{x^{(k)}}{x^{(k-1)}} \approx \frac{\lambda_1^k \alpha_1 \xi_1}{\lambda_1^{k-1} \alpha_1 \xi_1} = \lambda_1$$

于是利用向量序列 $\{x^{(k)}\}$ 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, $\lambda_1^k \rightarrow \infty$; $|\lambda_1| < 1$ 时, $\lambda_1^k \rightarrow 0$, 因而计算 $x^{(k)}$ 时可能会发生上溢或者下溢, 故每一步将 $x^{(k)}$ 归一化处理, 即将 $x^{(k)}$ 的各分量都除以模最大的分量, 使 $\|x^{(k)}\| = 1$, 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。

上述算法我们称为幂法。

算法 5.16 幂法

- 1: $k = 0; x^{(k)} = x$
- 2: **repeat**
- 3: $y^{(k+1)} = Ax^{(k)}$
- 4: $x^{(k+1)} = y^{(k+1)} / \|y^{(k+1)}\|_\infty$
- 5: $\lambda^{(k+1)} = x^{(k+1)T} Ax^{(k+1)}$
- 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 5.3.2. 设矩阵 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 $\{x^{(k)}\}$ 有算法产生, 则有

$$\lim_{k \rightarrow \infty} x^{(k)} = \frac{\xi_1}{\max\{\xi_1\}} = \xi_1^0, \quad \lim_{k \rightarrow \infty} m_k = \lambda_1, \quad (5.32)$$

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5.16的步2和步3知

$$\mathbf{x}^{(k)} = \frac{\mathbf{v}^{(k)}}{m_k} = \frac{\mathbf{A}\mathbf{x}^{(k-1)}}{m_k} = \frac{\mathbf{A}^2\mathbf{x}^{k-2}}{m_k m_{k-1}} = \cdots = \frac{\mathbf{A}^k\mathbf{x}^{(0)}}{m_k m_{k-1} \cdots m_1}.$$

由于 $\mathbf{x}^{(k)}$ 的最大分量为 1, 即 $\max\{\mathbf{x}^{(k)}\} = 1$, 故

$$m_k m_{k-1} \cdots m_1 = \max\{\mathbf{A}^k \mathbf{x}^{(0)}\}$$

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

例 5.3.3. 求矩阵 $\begin{pmatrix} 8 & -7 & 3 \\ -2 & 13 & -3 \\ -2 & 10 & 0 \end{pmatrix}$

以 $(0, 1, 1)^T$ 为初始迭代向量, 前 10 次迭代为

$$\left\{ \begin{array}{l} \mathbf{x}^{(1)} = \begin{pmatrix} -0.4 \\ 1 \\ 1 \end{pmatrix}, \quad \lambda^{(1)} = 10 \\ \mathbf{x}^{(2)} = \begin{pmatrix} -0.6667 \\ 1 \\ 1 \end{pmatrix}, \quad \lambda^{(2)} = 10.8 \\ \mathbf{x}^{(3)} = \begin{pmatrix} -0.8235 \\ 1 \\ 1 \end{pmatrix}, \quad \lambda^{(3)} = 11.3334 \\ \mathbf{x}^{(4)} = \begin{pmatrix} -0.9091 \\ 1 \\ 1 \end{pmatrix}, \quad \lambda^{(4)} = 11.6471 \end{array} \right.$$

$$\left\{ \begin{array}{l} \mathbf{x}^{(5)} = \begin{pmatrix} -0.9538 \\ 1 \\ 1 \end{pmatrix}, \quad \lambda^{(5)} = 11.8181 \\ \mathbf{x}^{(6)} = \begin{pmatrix} -0.9767 \\ 1 \\ 1 \end{pmatrix}, \quad \lambda^{(6)} = 11.9077 \\ \mathbf{x}^{(7)} = \begin{pmatrix} -0.9883 \\ 1 \\ 1 \end{pmatrix}, \quad \lambda^{(7)} = 11.9535 \end{array} \right.$$

$$\left\{ \begin{array}{l} \mathbf{x}^{(8)} = \begin{pmatrix} -0.9942 \\ 1 \\ 1 \end{pmatrix}, \quad \lambda^{(8)} = 11.9767 \\ \mathbf{x}^{(9)} = \begin{pmatrix} -0.9971 \\ 1 \\ 1 \end{pmatrix}, \quad \lambda^{(9)} = 11.9833 \\ \mathbf{x}^{(10)} = \begin{pmatrix} -0.9985 \\ 1 \\ 1 \end{pmatrix}, \quad \lambda^{(10)} = 11.9941 \end{array} \right.$$

故矩阵 $\begin{pmatrix} 8 & -7 & 3 \\ -2 & 13 & -3 \\ -2 & 10 & 0 \end{pmatrix}$ 的最大特征值约为 11.9941(实际为 12), 对应的特征向量为 $(-0.9985, 1, 1)^T$ (实际为 $(-1, 1, 1)^T$)。

5.3.3 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 A 按模最小的特征值及其特征向量。对于一个非奇异矩阵 A , A^{-1} 的特征值为矩阵 A 的特征值的倒数, 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

$$Ax^{(k+1)} = x^{(k)}, k = 0, 1, 2, \dots$$

求 $x^{(k+1)}$ 。为防止计算机溢出, 实际计算时所用公式为

$$\begin{aligned} v^{(k)} &= x^{(k)} / \max(x^{(k)}), \\ Ax^{(k+1)} &= x^{(k)}, \end{aligned} \quad k = 0, 1, 2, \dots \quad (5.33)$$

式中: $\max(x^{(k)})$ 为 $x^{(k)}$ 模最大的分量。

算法 5.17 反幂法

1: $k = 0; x^{(k)} = x$

2: **repeat**

3: $y^{(k+1)} = A^{-1}x^{(k)}$

4: $x^{(k+1)} = y^{(k+1)} / \|y^{(k+1)}\|_{\infty}$

5: $\lambda^{(k+1)} = x^{(k+1)T}Ax^{(k+1)} / (x^{(k+1)T}x^{(k+1)})$

6: $k = k + 1$

7: **until** 收敛

例 5.3.4. 求矩阵 $\begin{pmatrix} 1 & -1 & 2 \\ 3 & 0 & 2 \\ 3 & 5 & -1 \end{pmatrix}$

以 $(1, 0, 1)^T$ 为初始迭代向量, 前 7 次迭代为

$$\left\{ \begin{array}{l} x^{(1)} = \begin{pmatrix} -0.667 \\ 0.722 \\ 1 \end{pmatrix}, \\ \lambda^{(1)} = 0.10 \end{array} \right. , \left\{ \begin{array}{l} x^{(2)} = \begin{pmatrix} 0.874 \\ -0.552 \\ -1 \end{pmatrix}, \\ \lambda^{(2)} = -0.82 \end{array} \right. , \left\{ \begin{array}{l} x^{(3)} = \begin{pmatrix} -0.806 \\ 0.532 \\ 1 \end{pmatrix}, \\ \lambda^{(3)} = -0.78 \end{array} \right. , \left\{ \begin{array}{l} x^{(4)} = \begin{pmatrix} 0.813 \\ -0.523 \\ -1 \end{pmatrix}, \\ \lambda^{(4)} = -0.82 \end{array} \right. , \\ \left\{ \begin{array}{l} x^{(5)} = \begin{pmatrix} -0.810 \\ 0.522 \\ 1 \end{pmatrix}, \\ \lambda^{(5)} = -0.82 \end{array} \right. , \left\{ \begin{array}{l} x^{(6)} = \begin{pmatrix} 0.810 \\ -0.521 \\ -1 \end{pmatrix}, \\ \lambda^{(6)} = -0.83 \end{array} \right. , \left\{ \begin{array}{l} x^{(7)} = \begin{pmatrix} -0.810 \\ 0.521 \\ 1 \end{pmatrix}, \\ \lambda^{(7)} = -0.83 \end{array} \right. \end{array} \right.$$

故矩阵 $\begin{pmatrix} 8 & -7 & 3 \\ -2 & 13 & -3 \\ -2 & 10 & 0 \end{pmatrix}$ 的模最小特征值约为 -1.59 , 对应的特征向量为 $\begin{pmatrix} -0.810 \\ 0.51 \\ 1 \end{pmatrix}^T$

原点位移法

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, 于是式 5.33 相应变为

$$(A - \alpha I)x^{(k+1)} = x^{(k)}, k = 0, 1, 2, \dots$$

以求得 $x^{(k+1)}$ 。为防止计算机溢出, 实际计算时所用公式为

$$\begin{aligned} v^{(k)} &= x^{(k)} / \max(x^{(k)}), \\ (A - \alpha I)x^{(k+1)} &= x^{(k)}, \end{aligned} k = 0, 1, 2, \dots$$

算法 5.18 原点位移法

1: $k = 0; x^{(k)} = x$

2: **repeat**

3: $y^{(k+1)} = (A - \alpha I)^{-1}x^{(k)}$

4: $x^{(k+1)} = y^{(k+1)} / \|y^{(k+1)}\|_\infty$

5: $\lambda(k+1) = x^{(k+1)T}Ax^{(k+1)} / (x^{(k+1)T}x^{(k+1)})$

6: $k = k + 1$

7: **until** 收敛

例 5.3.5. 求矩阵 $\begin{pmatrix} 1 & -1 & 2 \\ 3 & 0 & 2 \\ 3 & 5 & -1 \end{pmatrix}$, 取 $\alpha = -1$, 以 $(1, 0, 1)^T$ 为初始迭代向量, 前 4 次迭代为

$$\left\{ \begin{array}{l} x^{(1)} = \begin{pmatrix} 0.824 \\ -0.471 \\ -1 \end{pmatrix}, \\ \lambda^{(1)} = -1.01 \end{array} \right. , \left\{ \begin{array}{l} x^{(2)} = \begin{pmatrix} 0.813 \\ -0.524 \\ -1 \end{pmatrix}, \\ \lambda^{(2)} = -0.82 \end{array} \right. , \left\{ \begin{array}{l} x^{(3)} = \begin{pmatrix} 0.810 \\ -0.521 \\ -1 \end{pmatrix}, \\ \lambda^{(3)} = -0.83 \end{array} \right. , \left\{ \begin{array}{l} x^{(4)} = \begin{pmatrix} 0.810 \\ -0.521 \\ -1 \end{pmatrix} \\ \lambda^{(4)} = -0.83 \end{array} \right. \right.$$

故矩阵 $\begin{pmatrix} 8 & -7 & 3 \\ -2 & 13 & -3 \\ -2 & 10 & 0 \end{pmatrix}$ 的模最小特征值约为 -1.59 , 对应的特征向量为 $\begin{pmatrix} -0.810 \\ 0.51 \\ 1 \end{pmatrix}^T$

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。

算法 5.19 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: $\mathbf{y}^{(k+1)} = (\mathbf{A} - \lambda_k \mathbf{I})^{-1} \mathbf{x}^{(k)}$
 - 5: $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)} / \|\mathbf{y}^{(k+1)}\|_\infty$
 - 6: $k = k + 1$
 - 7: **until** 收敛
-

例 5.3.6. 求矩阵 $\begin{pmatrix} 8 & -7 & 3 \\ -2 & 13 & -3 \\ -2 & 10 & 0 \end{pmatrix}$

以 $(0, 1, 1)^T$ 为初始迭代向量, 前 4 次迭代为

$$\left\{ \begin{array}{l} \mathbf{x}^{(1)} = \begin{pmatrix} -1 \\ 0.5 \\ 0.5 \end{pmatrix}, \\ \lambda^{(1)} = 11.3 \end{array} \right. , \left\{ \begin{array}{l} \mathbf{x}^{(2)} = \begin{pmatrix} -0.9 \\ 1 \\ 1 \end{pmatrix}, \\ \lambda^{(2)} = 12.0 \end{array} \right. , \left\{ \begin{array}{l} \mathbf{x}^{(3)} = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}, \\ \lambda^{(3)} = 12.0 \end{array} \right. , \left\{ \begin{array}{l} \mathbf{x}^{(4)} = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} \\ \lambda^{(4)} = 12.0 \end{array} \right.$$

故矩阵 $\begin{pmatrix} 8 & -7 & 3 \\ -2 & 13 & -3 \\ -2 & 10 & 0 \end{pmatrix}$ 的模最大特征值为 12, 对应的特征向量为 $\begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}^T$

5.3.4 特征值计算的应用：PageRank 网页排名

接下来我们介绍一个用于网页排名的算法——PageRank，他依赖于特征值的计算。

(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 阶对角矩阵, \mathbf{e} 为全是 1 的 n 维向量, 则

$$\mathbf{A} = p\mathbf{GD} + \frac{1-p}{n}\mathbf{ee}^T.$$

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 $\sum_{j=1}^n a_{ij} x_j^{(k)}$, 此时浏览个网页的概率分布为 $\mathbf{x}^{(k+1)} = \mathbf{Ax}^{(k)}$ 。

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 $\mathbf{Ax} = \mathbf{x}$, 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}

$$\begin{cases} \mathbf{Ax} = \mathbf{x} \\ \sum_{i=1}^n x_i = 1 \end{cases}$$

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 \mathbf{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 \mathbf{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 $\mathbf{Ax}^{(k)}$ 的结果。

例 5.3.7. 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。

我们使用 Python 中的 NumPy 库生成 \mathbf{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 \mathbf{A}

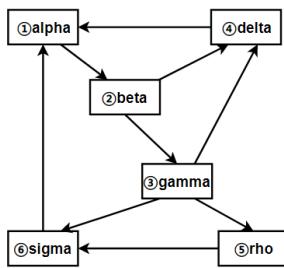


图 5.4: 网页链接关系

```

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 $\mathbf{x}_0 = [1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$

(2) $\mathbf{x}^{k+1} = \mathbf{A}\mathbf{x}^k$

(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

$$\mathbf{x} = [0.2675 \ 0.2524 \ 0.1323 \ 0.1697 \ 0.0625 \ 0.1156]^T$$

使用 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

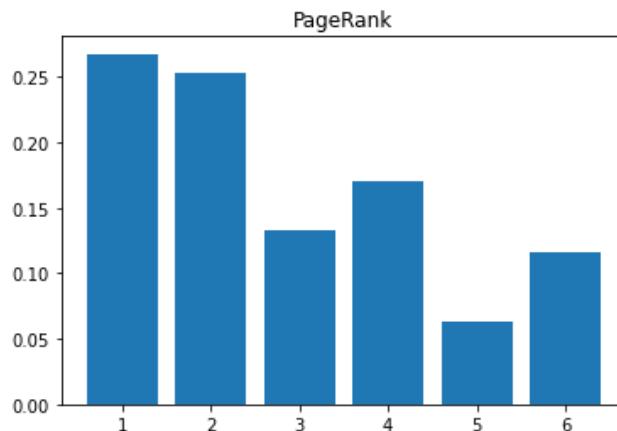


图 5.5: 网页的级别高低

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. 求解方程组

$$\begin{pmatrix} 2 & 1 & & \\ 1 & 2 & 1 & \\ & 1 & 2 & 1 \\ & & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

习题 5.2. 使用 LU 分解方程组 $Ax = b$, 其中 $A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{bmatrix}$, $b = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$.

习题 5.3. 设 $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ 用正则化方法求对应的 LS 问题的解。

习题 5.4. 设 $A = \begin{bmatrix} 1 & 3 & 1 & 1 \\ 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$, $b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ 求对应的 LS 问题的全部解。

习题 5.5. 设 $A \in \mathbb{R}^{m \times n}$ 且存在 $X \in \mathbb{R}^{n \times m}$ 使得对每一个 $b \in \mathbb{R}^m$, $x = Xb$ 均极小化 $\|Ax - b\|_2$. 证明 $AXA = A$ 和 $(AX)^T = AX$.

习题 5.6. 利用等式

$$\|A(x + \alpha w) - b\|_2^2 = \|Ax - b\|_2^2 + 2\alpha w^T A^T (Ax - b) + \alpha^2 \|Aw\|_2^2$$

证明: 如果 $x \in X_{LS}$, 那么 $A^T Ax = A^T b$.

习题 5.7. 给定点集 $p_1, \dots, p_m \in \mathbb{R}^n$ 构成的 $m \times n$ 矩阵 $P = [p_1, \dots, p_m]$ 。考虑问题

$$\min_X F(X) = \sum_{i=1}^m \|x_i - p_i\|_2^2 + \frac{\lambda}{2} \sum_{1 \leq i, j \leq m} \|x_i - x_j\|_2^2$$

其中 $\lambda \geq 0$ 为参数, 变量是一个 $m \times n$ 矩阵 $X = [x_1, \dots, x_m]$, 其中 $x_i \in \mathbb{R}^n$ 是 X 的第 i 列, $i = 1, \dots, m$. 上述问题尝试聚类点集 p_i , 第一项鼓励聚类中心 x_i 靠近对应的点 p_i , 第二项鼓励 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 $\frac{1}{2} \sum_{1 \leq i, j \leq m} \|x_i - x_j\|_2^2 = \text{Tr}(XH X^T)$, 其中 $H = mI_m - \mathbf{1}\mathbf{1}^T$ 是一个 $m \times m$ 矩阵, 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 H 是半正定的。

4. 证明函数 F 在矩阵 X 处的梯度是一个 $n \times m$ 矩阵, 为:

$$\nabla F(X) = 2(X - P + \lambda XH)$$

提示: 对于第二项, 找到函数的一阶展式, $\Delta \rightarrow \text{Tr}((X + \Delta)H(X + \Delta)^T)$, 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:

$$x_i = \frac{1}{m\lambda + 1} p_i + \frac{m\lambda}{m\lambda + 1} \hat{p}, i = 1, \dots, m,$$

其中 $\hat{p} = (1/m)(p_1 + \dots + p_m)$ 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 5.8. 判断 $[1, 3, 4]$ 的转置是否在 A 的零空间中?

$$A = \begin{bmatrix} 3 & 5 & -3 \\ 6 & -2 & 0 \\ -8 & 4 & 1 \end{bmatrix}$$

习题 5.9. 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 5.10. 简答: 阐述非负矩阵分解和主成分分析的相同点和不同点。

习题 5.11. 估计矩阵 $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & -4 \end{pmatrix}$ 特征值范围。

习题 5.12. 利用幂法求解矩阵 $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & -4 \end{pmatrix}$ 模最大的特征值与对应的特征向量。(特征值答案保留两位有效数字, 特征向量答案保留三位有效数字)

习题 5.13. 利用反幂法求解矩阵 $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & -4 \end{pmatrix}$ 模最小的特征值与对应的特征向量。(特征值答案保留两位有效数字, 特征向量答案保留三位有效数字)

习题 5.14. 利用原点位移法求解矩阵 $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & -4 \end{pmatrix}$ 全部特征值与对应的特征向量。(特征值答案保留两位有效数字, 特征向量答案保留三位有效数字)

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 solution 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, Vm, 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 embedding 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 OptimizatiorL 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.

数据科学与工程数学基础初稿

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，通过重复应用链式法则来最小化重构误差；高斯混合模型用于数据分布的建模，优化每个用来混合的分布的位置和形状参数，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。

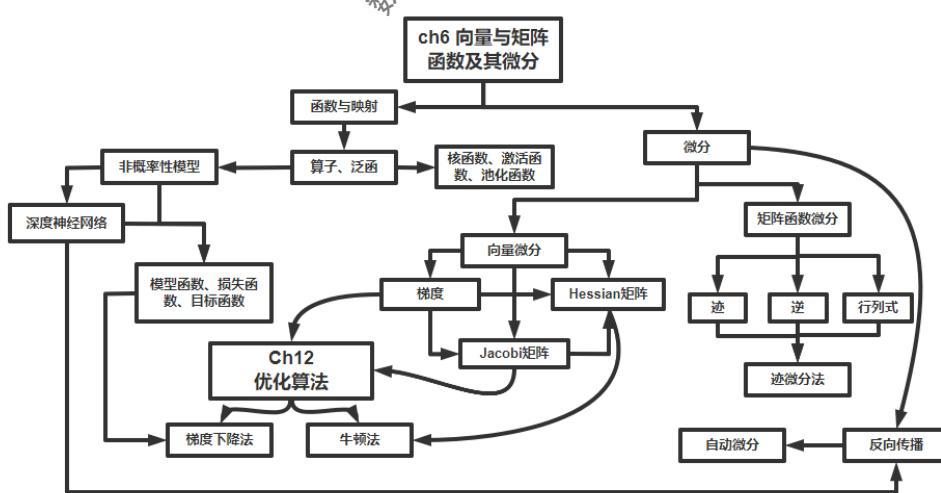


图 6.1: 本章导图

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 $T : M \rightarrow \mathbb{R}$ 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 $y = T(A) = T(a_{11}, a_{12}, \dots, a_{mn})$ 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 ，若函数 f 满足性质：

$$\begin{aligned} f(\alpha\mathbf{x} + \beta\mathbf{y}) &= \mathbf{a}^T(\alpha\mathbf{x} + \beta\mathbf{y}) = \mathbf{a}^T(\alpha\mathbf{x}) + \mathbf{a}^T(\beta\mathbf{y}) \\ &= \alpha(\mathbf{a}^T\mathbf{x}) + \beta(\mathbf{a}^T\mathbf{y}) = \alpha f(\mathbf{x}) + \beta f(\mathbf{y}) \end{aligned}$$

则称这个函数满足叠加性。

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：

定义 6.1.4. 齐次性：对于任意 n 维向量 \mathbf{x} 和标量 α ，函数 f 有 $f(\alpha\mathbf{x}) = \alpha f(\mathbf{x})$

可加性：对于任意 n 维向量 \mathbf{x}, \mathbf{y} ，函数 f 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，如果一个函数是线性的，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 $\mathbf{a} = \mathbf{b}$ 。 \square

定义 6.1.5. 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 6.1.6. 设 M 是一非空集合, 当 $N = \mathbb{R}^n$ 时, 函数 $T: M \rightarrow \mathbb{R}^n$ 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 \mathbf{x} 的向量值函数:

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad f(\mathbf{x}) = A\mathbf{x},$$

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 $f(\mathbf{x}) = A\mathbf{x}$, 则它是线性函数, 也即满足叠加性:

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

对于 n 维向量 \mathbf{x}, \mathbf{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 $f(\mathbf{x}) = A\mathbf{x}$ 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 $f(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha f(\mathbf{x}) + \beta f(\mathbf{y})$ 对于所有的 n 维向量 \mathbf{x}, \mathbf{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 $f(\mathbf{x}) = A\mathbf{x}$ 对所有 \mathbf{x} 成立。

定义 6.1.7. 向量值函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 如果能够写成 $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ 的形式, 那么 f 是一个仿射函数, 其中 A 是 $m \times n$ 矩阵, \mathbf{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 $f(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha f(\mathbf{x}) + \beta f(\mathbf{y})$ 对于所有的 n 维向量 \mathbf{x}, \mathbf{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 $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ 的形式, 矩阵 A 和向量 \mathbf{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:

$$A = [f(e_1) - f(0), f(e_2) - f(0), \dots, f(e_n) - f(0)], \mathbf{b} = f(0).$$

与标量值函数的情形下相同, 只有 $\mathbf{b} = 0$ 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: $f(\mathbf{x}) = (|x_1|, |x_2|, \dots, |x_n|)$ 是非线性向量值函数。取 $n = 1, x = 1, y = 0, \alpha = -1, \beta = 0$ 有

$$f(\alpha\mathbf{x} + \beta\mathbf{y}) = 1 \neq \alpha f(\mathbf{x}) + \beta f(\mathbf{y}) = -1$$

所以不满足叠加性。

例 6.1.8. 排序函数: f 将 \mathbf{x} 的元素降序排列, 是非线性向量值函数 ($n > 1$)。取 $n = 2, \mathbf{x} = (1, 0), \mathbf{y} = (0, 1), \alpha = \beta = 1$ 则

$$f(\alpha\mathbf{x} + \beta\mathbf{y}) = (1, 1) \neq \alpha f(\mathbf{x}) + \beta f(\mathbf{y}) = (2, 0)$$

所以不满足叠加性。

C. 矩阵值函数

定义 6.1.8. 设 \mathbb{M} 和 \mathbb{N} 是两个非空的矩阵集合, 函数 $T: \mathbb{M} \rightarrow \mathbb{N}$ 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 $\mathbf{L} \in \mathbb{R}^{m \times n}$ 和 $T: \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{n \times n}$, $T(\mathbf{L}) = \mathbf{L}^T \mathbf{L}$ 是一个矩阵函数。
- 逆函数: $f(\mathbf{A}) = \mathbf{A}^{-1}$

6.1.2 算子

定义 6.1.9. 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 6.1.10. 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 6.1.11. 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 $\forall \mathbf{x}, \mathbf{y} \in D, \forall \alpha, \beta \in \mathbb{K}$, 有 $T(\alpha \mathbf{x} + \beta \mathbf{y}) = \alpha T(\mathbf{x}) + \beta T(\mathbf{y})$, 则称 T 为 D 上的线性算子。

定义 6.1.12. 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. (一些常见的算子)

- (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, $\forall x \in X, 0x = 0$.
- (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, $\forall x(t) \in C[a, b], Tx = \int_a^t x(\tau) d\tau, t \in [a, b]$ 。

证明. 证明设 $x(t), y(t) \in C[a, b], \alpha, \beta \in \mathbb{R}$, 则

$$T(\alpha x + \beta y) = \int_a^t (\alpha x(\tau) + \beta y(\tau)) d\tau = \alpha \int_a^t x(\tau) d\tau + \beta \int_a^t y(\tau) d\tau = \alpha Tx + \beta Ty$$

$$Tx \leq \max_{t \in [a, b]} \left| \int_a^t x(\tau) d\tau \right| \leq \max_{t \in [a, b]} \int_a^t |x(\tau)| d\tau \leq \max_{t \in [a, b]} |x(t)| \int_a^t 1 d\tau = \|x(t)\|(b-a)$$

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 6.1.13. 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 。

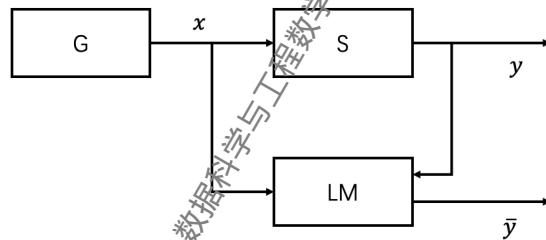


图 6.2: 从实例学习的模型。在学习过程中, 学习机器观测一系列点对 (x, y) (训练集)。训练后, 机器对任何一个给定的 x 必须返回一个 \bar{y} 值。目标是返回一个非常接近于训练器响应 y 的 \bar{y} 。

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 \mathbb{Z} 上, 给定一个容许函数集 $|g(z)|, z \in \mathbb{Z}$, 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 $R(g(z)) = \int L(z, g(z)) dF(z)$ 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(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。

6.2 统计机器学习中的非概率型函数模型

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。

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。

给定数据集 $\mathbb{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$, 其中 $\mathbf{x}_i = (x_{i1}; x_{i2}; \dots; x_{id})$, $y_i \in \mathbb{R}$ 。线性回归的模型函数为 $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + b$ 。该函数 $f(\mathbf{x}_i) \simeq y_i$ 。通常我们使用均方误差来度量线性回归的损失, 所以损失函数为 $L(f; T) = \frac{1}{m} \sum_{i=1}^m (f(\mathbf{x}_i) - y_i)^2$ 。所以目标函数为

$$(\mathbf{w}^*, b^*) = \arg \min_{(\mathbf{w}, b)} \sum_{i=1}^m (f(\mathbf{x}_i) - y_i)^2 = \arg \min_{(\mathbf{w}, b)} \sum_{i=1}^m (\mathbf{w}^T \mathbf{x}_i + b - y_i)^2$$

线性回归中的变体

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, 如果使用对数几率函数 $y = \frac{1}{1+e^{-z}}$ 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 $y = \frac{1}{1+e^{-(\mathbf{w}^T \mathbf{x} + b)}}$ 。类似对数线性回归, 对数几率回归可以变化为 $\ln \frac{y}{1-y} = \mathbf{w}^T \mathbf{x} + b$ 。

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:

$$f(x) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$$

称为感知机。其中, \mathbf{w} 和 b 为感知机模型参数, $\mathbf{w} \in \mathbb{R}^n$ 叫作权值 (weight) 或权值向量 (weight vector), $b \in \mathbb{R}$ 叫作偏置 (bias), $\mathbf{w}^T \mathbf{x}$ 表示 \mathbf{w} 和 \mathbf{x} 的内积。 sign 是符号函数, 即

$$\text{sign}(x) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

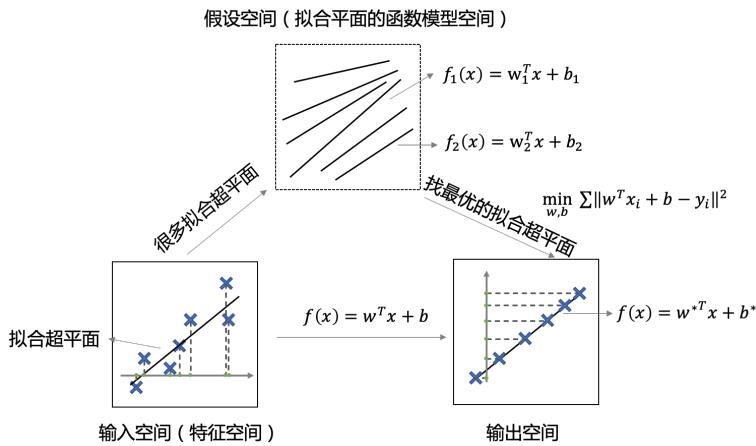


图 6.3: 线性回归

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 $\{f | f(x) = \mathbf{w}^T \mathbf{x} + b\}$ 。

函数模型对应的线性方程 $\mathbf{w}^T \mathbf{x} + b = 0$ 称为对应于特征空间的分离超平面，它由法向量 \mathbf{w} 和截距 b 决定，可用 (\mathbf{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 \mathbf{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，输入空间 \mathbb{R}^n 中任一点 x_0 到超平面 S 的距离为 $\frac{1}{\|\mathbf{w}\|_2} |\mathbf{w}^T \mathbf{x}_0 + b|$ 。误分类点 \mathbf{x}_i 到超平面 S 的距离是 $-\frac{1}{\|\mathbf{w}\|_2} y_i (\mathbf{w}^T \mathbf{x}_i + b)$ 。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 $-\frac{1}{\|\mathbf{w}\|_2} \sum_{x_i \in M} y_i (\mathbf{w}^T \mathbf{x}_i + b)$ 。 $\frac{1}{\|\mathbf{w}\|_2}$ 是常系数可以省略，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。

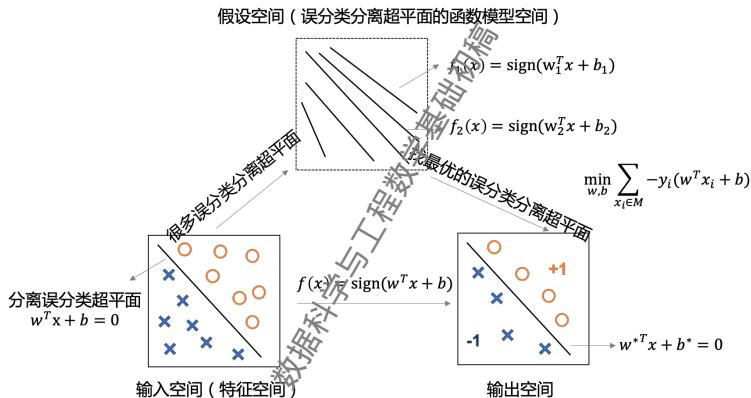


图 6.4: 感知机

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

$$\mathbf{w}^*{}^T \mathbf{x} + b^* = 0$$

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置信度，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 (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 (w, b) 关于训练数据集 T 的函数间隔为超平面 (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 (w, b) ，定义超平面关于样本点 (x_i, y_i) 的几何间隔为

$$\gamma_i = y_i \left(\frac{w}{\|w\|} x_i + \frac{b}{\|w\|} \right)$$

定义超平面 (w, b) 关于训练数据集 T 的几何间隔为超平面 (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

$$\gamma = \min_{i=1, \dots, N} \gamma_i$$

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置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)也有足够大的置信度将它们分开。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:

$$\begin{aligned} & \max_{w,b} \gamma \\ & \text{s.t. } y_i \left(\frac{w}{\|w\|} x_i + \frac{b}{\|w\|} \right) \geq \gamma, i = 1, 2, \dots, N \end{aligned}$$

考虑几何间隔和函数间隔的关系,可以将问题改写成

$$\begin{aligned} & \max_{w,b} \frac{\hat{\gamma}}{\|w\|} \\ & \text{s.t. } y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq \hat{\gamma}, i = 1, 2, \dots, N \end{aligned}$$

更进一步,将 $\hat{\gamma} = 1$ 代入上面的最优化问题,实际上,这是对上述优化问题进行等价的变量代换。再注意到最大化 $\frac{1}{\|w\|}$ 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:

$$\begin{aligned} & \min_{w,b} \frac{1}{2} \|w\|^2 \\ & \text{s.t. } y_i (\mathbf{w}^\top \mathbf{x}_i + b) - 1 \geq 0, i = 1, 2, \dots, N \end{aligned}$$

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。

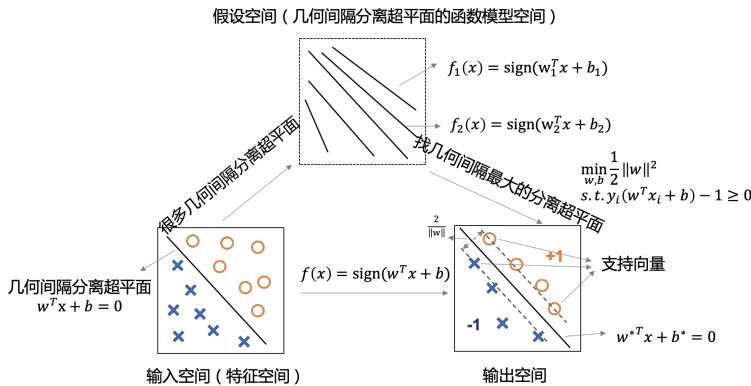


图 6.5: 线性可分支持向量机

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

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1,$$

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

$$\begin{aligned} & \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ & \text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, 2, \dots, N \\ & \quad \xi_i \geq 0, i = 1, 2, \dots, N \end{aligned}$$

这就是常用的线性支持向量机。线性支持向量机处理的是线性不可分的数据集, 然而训练数据集又近似线性可分时, 通过软间隔最大化, 也可学习出一个线性分类器, 它也被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:

$$\begin{aligned} & \min_{\mathbf{w}, b, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \\ & \text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, 2, \dots, N \\ & \quad \xi_i \geq 0, i = 1, 2, \dots, N \end{aligned}$$

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

$$\mathbf{w}^{*T} \mathbf{x} + b^* = 0$$

以及相应的分类决策函数

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^{*T} \mathbf{x} + b^*)$$

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。

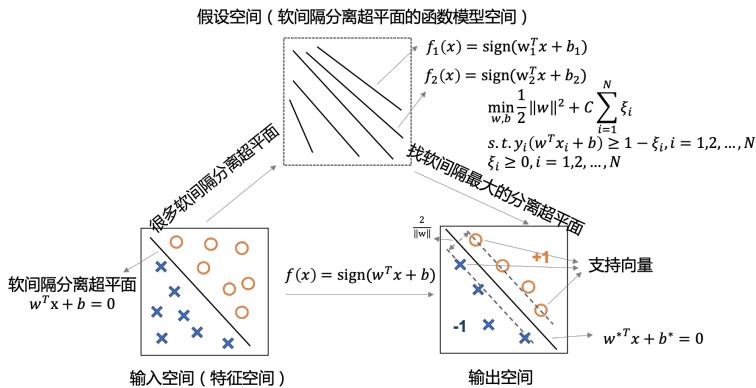


图 6.6: 线性支持向量机

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 $T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, 其中, 实例 \mathbf{x}_i 属于输入空间, $\mathbf{x}_i \in \mathbb{X} = \mathbb{R}^n$, 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 $\mathbb{X} \subset \mathbb{R}^2$, $\mathbf{x} = (x^{(1)}, x^{(2)})^T \in \mathbb{X}$, 新空间为 $\mathbb{Z} \subset \mathbb{R}^2$, $\mathbf{z} = (z^{(1)}, z^{(2)})^T \in \mathbb{Z}$, 定义从原空间到新空间的变换 (映射):

$$\mathbf{z} = \phi(\mathbf{x}) = (((x^{(1)})^2, (x^{(2)})^2))^T$$

经过变换 $\mathbf{z} = \phi(\mathbf{x})$, 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。

问题的关键是如何构造变换? 实际上这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 $K(x, z) = (x^T z)^2$, 试找出其相关的特征空间 \mathbb{H} 和映射 $\phi(x) : \mathbb{R}^2 \rightarrow \mathbb{H}$ 。

解: 取特征空间 $\mathbb{H} = \mathbb{R}^3$, 记 $x = (x^{(1)}, x^{(2)})^T$, $z = (z^{(1)}, z^{(2)})^T$, 由于

$$(x^T z)^2 = (x^{(1)}z^{(1)} + x^{(2)}z^{(2)})^2 = (x^{(1)}z^{(1)})^2 + 2x^{(1)}z^{(1)}x^{(2)}z^{(2)} + (x^{(2)}z^{(2)})^2$$

所以可以取映射 $\phi(x) = ((x^{(1)})^2, \sqrt{2}x^{(1)}x^{(2)}, (x^{(2)})^2)^T$,

容易验证 $\phi(x)^T \phi(z) = (x^T z)^2 = K(x, z)$ 。

仍取 $\mathbb{H} = \mathbb{R}^3$ 以及 $\phi(x) = \frac{1}{\sqrt{2}}((x^{(1)})^2 - (x^{(2)})^2, 2x^{(1)}x^{(2)}, (x^{(1)})^2 + (x^{(2)})^2)^T$,

同样有 $\phi(x)^T \phi(z) = (x^T z)^2 = K(x, z)$ 。

还可以取 $\mathbb{H} = \mathbb{R}^4$ 和 $\phi(x) = ((x^{(1)})^2, x^{(1)}x^{(2)}, x^{(1)}x^{(2)}, (x^{(2)})^2)^T$ 。

定理 6.2.1. 设 $K : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ 是对称函数, 则 $K(x, z)$ 为正定核函数的充要条件是对任意 $x_i \in \mathbb{X}, i = 1, 2, \dots, m$, $K(x, z)$ 对应的 Gram 矩阵

$$\mathbf{K} = [K(x_i, x_j)]_{m \times m}$$

是半正定矩阵。

定义 6.2.7. 设 $\mathbb{X} \subset \mathbb{R}^n$, $K(x, z)$ 是定义在 $\mathbb{X} \times \mathbb{X}$ 上的对称函数, 如果对于任意 $x_i \in \mathbb{X}, i = 1, 2, \dots, m$, $K(x, z)$ 对应的 Gram 矩阵

$$\mathbf{K} = [K(x_i, x_j)]_{m \times m}$$

是半正定矩阵, 则称 $K(x, z)$ 是正定核。

例 6.2.5. (常见的核函数)

- 线性核函数: $\kappa(x, z) = x^T z + c$

- 多项式核函数: $\kappa(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z})^d$
- 高斯核函数: $\kappa(\mathbf{x}, \mathbf{z}) = e^{-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2}}$
- 拉普拉斯核: $\kappa(\mathbf{x}, \mathbf{z}) = e^{-\frac{\|\mathbf{x}-\mathbf{z}\|}{\sigma}}$
- Sigmoid 核: $\kappa(\mathbf{x}, \mathbf{z}) = \tan(\alpha \mathbf{x}^T \mathbf{z} + c)$
- 字符串核函数

性质 6.2.1. 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 $g(\mathbf{x})$,

$$K(\mathbf{x}, \mathbf{z}) = g(\mathbf{x})K_1(\mathbf{x}, \mathbf{z})g(\mathbf{z})$$

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 中的超平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

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i^* y_i K(\mathbf{x}, \mathbf{x}_i) + b^*\right)$$

称为非线性支持向量机, $K(\mathbf{x}, \mathbf{z})$ 是正定核函数。

选取适当的核函数 $K(\mathbf{x}, \mathbf{z})$ 和适当的参数 C , 构造并求解最优化问题:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, \dots, N \end{aligned}$$

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。

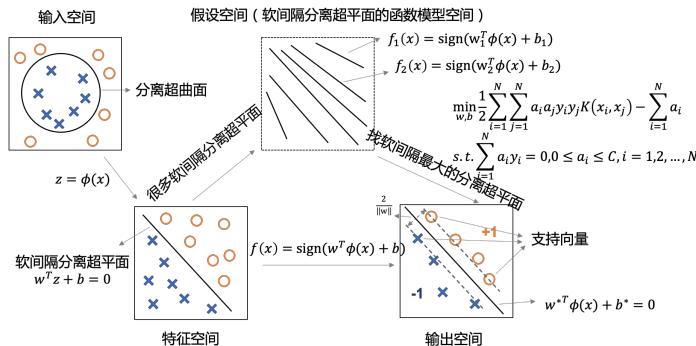


图 6.7: 支持向量机

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)。缓解维数灾难的一个重要途径是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”。这主要有两类方法:

- 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。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, $\mathbf{Z} = (z_1, z_2, \dots, z_m) \in \mathbb{R}^{d' \times m}$ 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。

我们从使样本点到子空间的距离最近的思想出发, 即我们要求最近重构性。如果我们假定了数据样本进行了中心化, 即 $\sum_i \mathbf{x}_i = \mathbf{0}$, 并且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 $z_i = \mathbf{W}^T \mathbf{x}_i$ 。而这些样本点在原始空间中的坐标为 $\mathbf{W} \mathbf{W}^T \mathbf{x}_i$ 。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 $\mathbf{W} \mathbf{W}^T \mathbf{x}_i$ 之间的距离为

$$\|\mathbf{x}_i - \mathbf{W} \mathbf{W}^T \mathbf{x}_i\|_2$$

那么考虑整个训练集, 对于所有样本总的距离为

$$\|\mathbf{X} - \mathbf{W} \mathbf{W}^T \mathbf{X}\|_F,$$

这可以看成投影变换后新样本和原样本之间的损失函数。

所以优化问题为

$$\begin{aligned}\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{W}\mathbf{W}^T \mathbf{X}\|_F^2 &= \min_{\mathbf{W}} \text{Tr}((\mathbf{X} - \mathbf{W}\mathbf{W}^T \mathbf{X})^T (\mathbf{X} - \mathbf{W}\mathbf{W}^T \mathbf{X})) \\ &= \min_{\mathbf{W}} \text{Tr}(\mathbf{X}^T \mathbf{X} - 2\mathbf{X}^T \mathbf{W}\mathbf{W}^T \mathbf{X} + \mathbf{X}^T \mathbf{W}\mathbf{W}^T \mathbf{W}\mathbf{W}^T \mathbf{X}) \\ &= \min_{\mathbf{W}} \text{Tr}(-\mathbf{X}^T \mathbf{W}\mathbf{W}^T \mathbf{X})\end{aligned}$$

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，就得到最终的优化问题

$$\begin{aligned}\min_{\mathbf{W}} \text{Tr}(-\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}) \\ \text{s.t. } \mathbf{W}^T \mathbf{W} = \mathbf{I}\end{aligned}$$

事实上，主成分分析的另外一种想法是使得样本点在子空间上的投影能尽可能分开。而这两种不同的思想最终推断出优化问题是相同的。

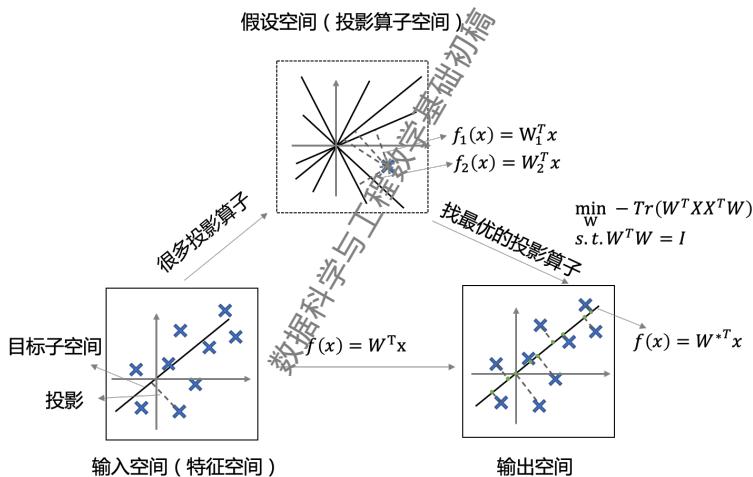


图 6.8: 降维

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，而这些类和簇事先是不知道的。

聚类算法有很多，主要有两类方法：层次聚类和 k 均值聚类。我们这里只介绍 k 均值聚类的思想和其中的函数。

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，采用平方欧氏距离（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：

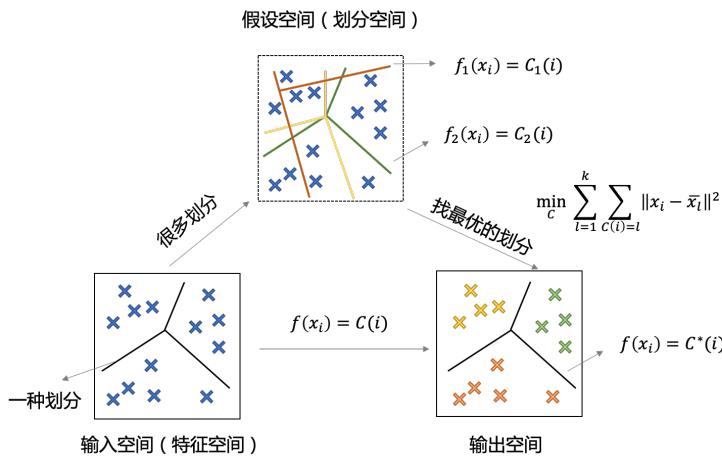
$$C^* = \arg \min_C W(C) = \arg \min_C \sum_{l=1}^k \sum_{C(i)=l} \|V\mathbf{x}_i - \bar{\mathbf{x}}_l\|^2$$

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 MINST 数字识别这个任务。

例 6.3.1. 在 MINST 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 \mathbf{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 \mathbf{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 \mathbf{x} ，已知它所代表的数字。我们想要找到一个函数 f （即分类规则，位于假设空间中），

$$f : \mathbb{R}^{784} \rightarrow \mathbb{R}^{10}$$

$$\mathbf{y}' = f(\mathbf{x})$$

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 \mathbf{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 $\|\mathbf{y}' - \mathbf{y}\|$ 尽可能小，也即求解最优化问题

$$\min \|\mathbf{y}' - \mathbf{y}\|$$

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的 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, 而分片用于实现真实图像和数据必然要求的非线性。

CPL 函数所在的假设空间是连续分片线性函数空间。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 $\mathbf{x} \in \mathbb{R}^n$, 存在一个无洞的子集 $\mathbb{I} \subset \mathbb{R}^n$ 包含 \mathbf{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。

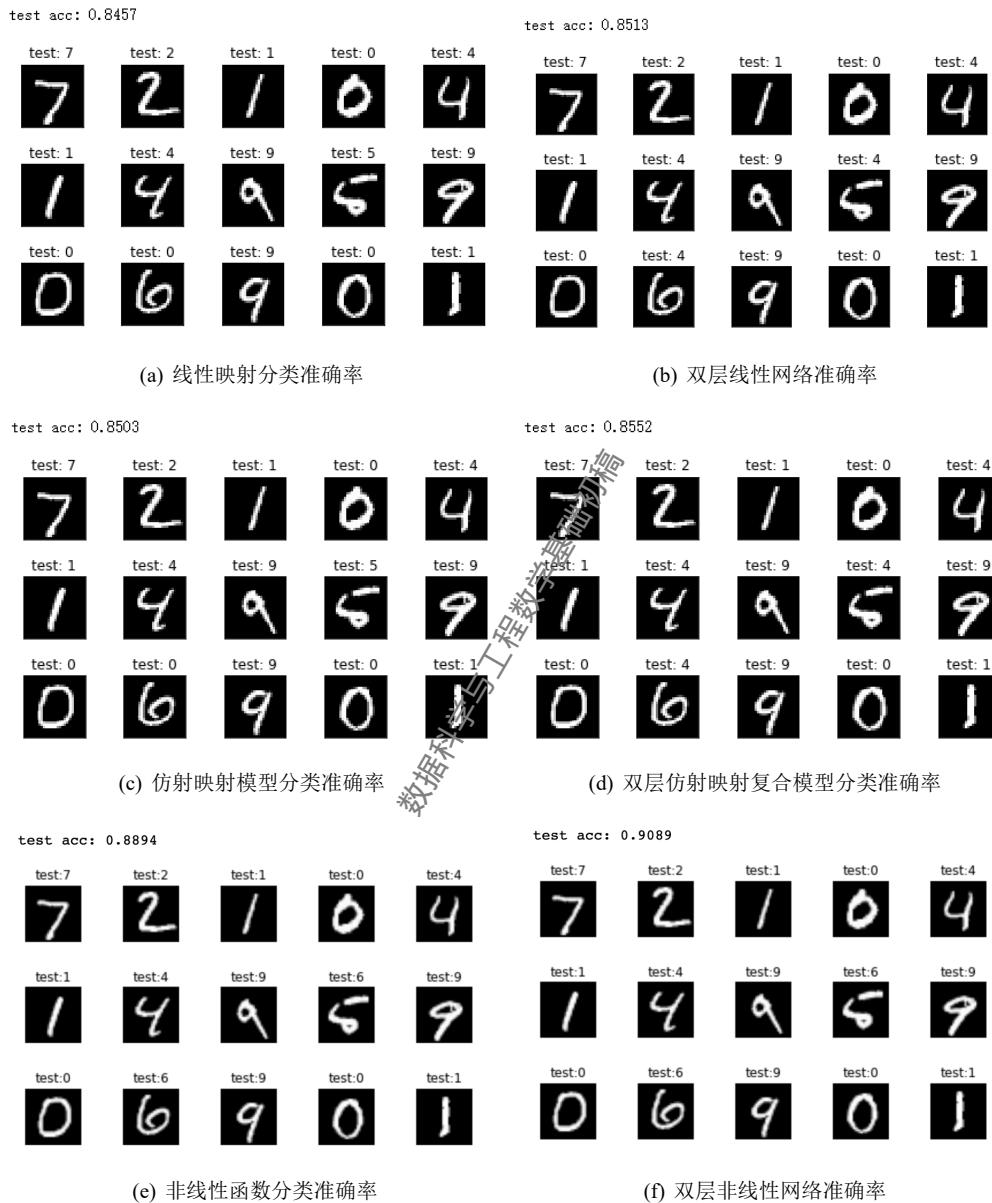
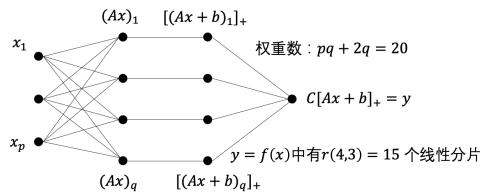


图 6.10: 线性映射、双层线性映射、仿射映射、双层仿射映射、非线性函数和双层非线性网络准确率的对比

图 6.11: 数据向量 v 的分片线性函数的神经网络架构

我们首先来看看连续分片线性 (CPL) 函数的构造。

图6.11是数据向量 v 的分片线性函数的初步构造:

1. 首先确定矩阵 A 和向量 b 。
2. 接着将 $Av + b$ 中所有的负分量设为 0(此步是非线性的)。
3. 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。

在图6.11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)函数施加 $ReLU$, 即 $F_i(x) = (Ax_i + b_i)_+$ 是非线性的, 它们的复合是 $F(x) = CF_L(F_{L-1}(\dots F_2(F_1(x))))$, 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。

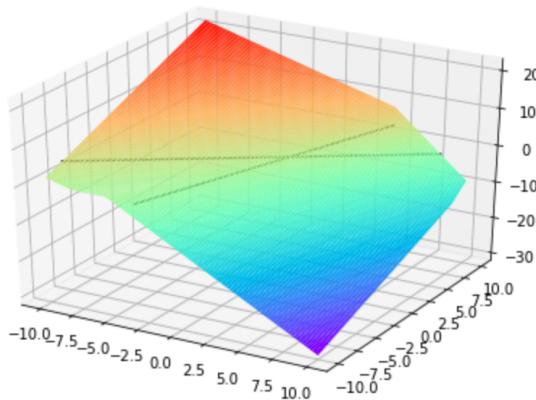


图 6.12: 一层的神经网络

例 6.3.3. 考虑一个具有三个隐藏层的神经网络, 其中

$$F_1 = \text{ReLU} \left(\begin{pmatrix} 2 & -1 \\ -1 & 0 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 1 \\ -2 \end{pmatrix} \right),$$

$$F_2 = \text{ReLU} \left(\begin{pmatrix} 1 & 2 \\ -2 & -3 \end{pmatrix} \mathbf{x} + \begin{pmatrix} -1 \\ 2 \end{pmatrix} \right),$$

$$F_3 = \text{ReLU} \left(\begin{pmatrix} 2 & 4 \\ -2 & 3 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 1 \\ 2 \end{pmatrix} \right)$$

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 $\mathbf{A}\mathbf{x} + \mathbf{b}$ 或者卷积, 然后再通过激活函数作用变成一个非线性函数。

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。

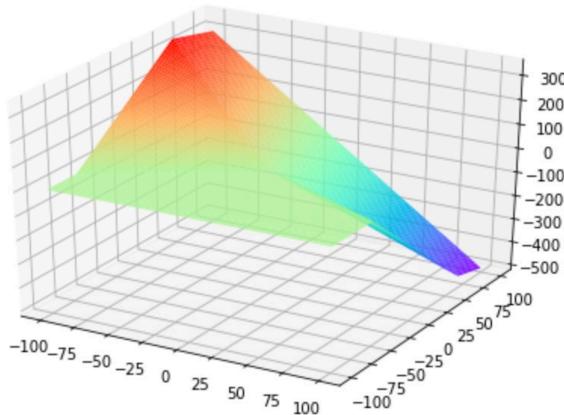


图 6.13: 三个隐藏层的神经网络

- 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片函数来近似。

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 函数可以用分片函数 hard-logistic(\mathbf{x}) 来近似

$$\begin{aligned} \text{hard-logistic}(x) &= \begin{cases} 1 & g_l(x) \geq 1 \\ g_l(x) & 0 < g_l(x) < 1 \\ 0 & g_l(x) \leq 0 \end{cases} \\ &= \max(\min(g_l(x), 1), 0) \\ &= \max(\min(0.25x + 0.5, 1), 0) \end{aligned}$$

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 函数也可以用分片函数 hard-tanh(\mathbf{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

$$\text{Swish}(x) = x\sigma(\beta x)$$

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} 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 $\mathbf{A}\mathbf{x} + \mathbf{b}$ 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 $F(\mathbf{x}) = F_n(\dots F_2(F_1(\mathbf{x})))$ 。

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。

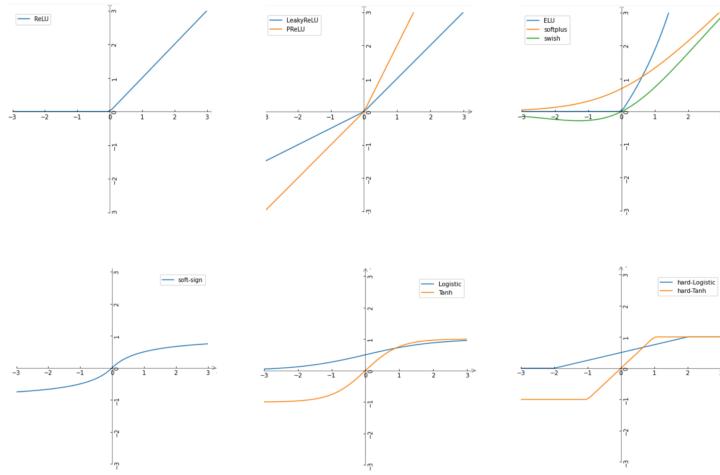


图 6.14: 激活函数大览

6.4 向量和矩阵函数的梯度

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:

$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 > 0 \end{aligned}$$

• PCA 对应的优化问题:

$$\begin{aligned} & \min_{\mathbf{W}} -\text{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}) \\ & \text{s.t. } \mathbf{W}^T \mathbf{W} = \mathbf{I} \end{aligned}$$

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

$$\mathbf{h} = \text{ReLU}(\mathbf{A}_1 \mathbf{x} + \mathbf{b}_1)$$

$$\mathbf{y}' = \text{ReLU}(\mathbf{A}_2 \mathbf{h} + \mathbf{b}_2)$$

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:

$$\min \|\mathbf{y} - \mathbf{y}'\|_2^2$$

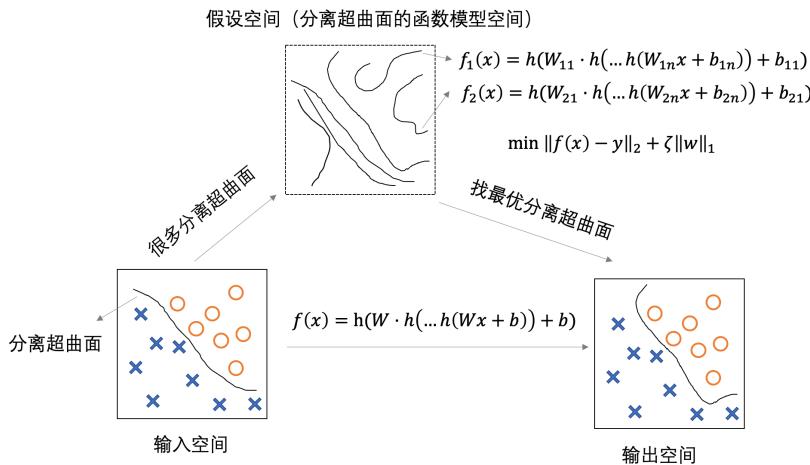


图 6.15: 神经网络

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

$$\frac{df}{dx} := \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

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 $f: \mathbb{R} \rightarrow \mathbb{R}$, $f \in \mathbb{C}^\infty$ 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

$$\begin{aligned}\frac{\partial f}{\partial x_1} &= \lim_{h \rightarrow 0} \frac{f(x_1 + h, x_2, \dots, x_n) - f(\mathbf{x})}{h} \\ &\vdots \\ \frac{\partial f}{\partial x_n} &= \lim_{h \rightarrow 0} \frac{f(x_1, x_2, \dots, x_n + h) - f(\mathbf{x})}{h}\end{aligned}$$

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 \mathbf{x} 的梯度为一 $n \times 1$ 列向量, 定义为

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

$$\frac{\partial}{\partial \mathbf{x}} f = \nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

通常还可以从可微的角度定义梯度如下:

定义 6.4.6. (梯度) 给定函数 $f : \mathbb{R}^n \rightarrow \mathbb{R}$, 且 f 在点 \mathbf{x} 的一个邻域内有意义, 若存在向量 $\mathbf{g} \in \mathbb{R}^n$ 满足

$$\lim_{\mathbf{p} \rightarrow 0} \frac{f(\mathbf{x} + \mathbf{p}) - f(\mathbf{x}) - \mathbf{g}^T \mathbf{p}}{\|\mathbf{p}\|} = 0,$$

其中 $\|\cdot\|$ 是任意的向量范数, 就称 f 在点 \mathbf{x} 处可微 (或 Fréchet 可微). 此时 \mathbf{g} 称为 f 在点 \mathbf{x} 处的梯度, 记作 $\nabla f(\mathbf{x})$.

实际上, 若 f 在点 \mathbf{x} 处的梯度存在 (或 Fréchet 可微), 在上式中令 $\mathbf{p} = \varepsilon \mathbf{e}_i$, \mathbf{e}_i 是第 i 个分量为 1 的单位向量, 可知 $\nabla f(\mathbf{x})$ 的第 i 个分量为 $\frac{\partial f(\mathbf{x})}{\partial x_i}$. 因此,

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial f(\mathbf{x})}{\partial x_n} \right]^T.$$

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 $f(\mathbf{x}) = \sin x_1 + 2x_1x_2 + x_2^2$ 。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

$$\nabla f(\mathbf{x}) = \frac{1}{\|\mathbf{x} - \mathbf{p}\|_2} (\mathbf{x} - \mathbf{p})$$

向量函数导数的运算法则

与一元函数类似, 向量函数导数有如下运算法则:

- 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

$$\frac{\partial[c_1 f(\mathbf{x}) + c_2 g(\mathbf{x})]}{\partial \mathbf{x}} = c_1 \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} + c_2 \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \quad (6.7)$$

- 乘法法则: 若 $f(\mathbf{x})$ 和 $g(\mathbf{x})$ 都是向量 \mathbf{x} 的实值函数, 则

$$\frac{\partial f(\mathbf{x})g(\mathbf{x})}{\partial \mathbf{x}} = g(\mathbf{x}) \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} + f(\mathbf{x}) \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \quad (6.8)$$

- 商法则: 若 $g(\mathbf{x}) \neq 0$, 则

$$\frac{\partial f(\mathbf{x})/g(\mathbf{x})}{\partial \mathbf{x}} = \frac{1}{g^2(\mathbf{x})} \left(g(\mathbf{x}) \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} - f(\mathbf{x}) \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \right) \quad (6.9)$$

关于复合函数的链式法则我们后面再进行介绍。

6.4.2 矩阵函数的梯度

定义 6.4.7. 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 6.4.6. 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 6.4.7, 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 6.4.7, 由此可导出向量函数梯度定义 6.4.5。通过定义 6.4.7 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 6.4.7, 由此可导出向量函数梯度定义 6.4.5。通过定义 6.4.7 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6.4.7, 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 6.4.7. 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 6.4.8. 对于一个可分的支持向量机, 相应的优化问题为

$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{s.t. } y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 > 0 \end{aligned}$$

考虑其目标函数的梯度 $\frac{\partial}{\partial \mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2$ 。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 $\frac{\partial}{\partial \mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 = \mathbf{w}$ 。

矩阵函数导数的运算法则

实值函数相对于矩阵变元的梯度具有以下性质。

- 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

$$\frac{\partial [c_1 f(\mathbf{A}) + c_2 g(\mathbf{A})]}{\partial \mathbf{A}} = c_1 \frac{\partial f(\mathbf{A})}{\partial \mathbf{A}} + c_2 \frac{\partial g(\mathbf{A})}{\partial \mathbf{A}}$$

- 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

$$\frac{\partial f(\mathbf{A})g(\mathbf{A})}{\partial \mathbf{A}} = g(\mathbf{A}) \frac{\partial f(\mathbf{A})}{\partial \mathbf{A}} + f(\mathbf{A}) \frac{\partial g(\mathbf{A})}{\partial \mathbf{A}}$$

- 商法则: 若 $g(\mathbf{A}) \neq 0$, 则

$$\frac{\partial f(\mathbf{A})/g(\mathbf{A})}{\partial \mathbf{A}} = \frac{1}{g^2(\mathbf{A})} \left[g(\mathbf{A}) \frac{\partial f(\mathbf{A})}{\partial \mathbf{A}} - f(\mathbf{A}) \frac{\partial g(\mathbf{A})}{\partial \mathbf{A}} \right]$$

6.5 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 6.5.1. 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 6.5.2. 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 6.5.1. 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 6.5.1. 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.

$$\begin{aligned} \text{左边} &= d \left(\sum_i a_{ii} \right) = \sum_i da_{ii} \\ \text{右边} &= \text{Tr} \left(\begin{pmatrix} da_{11} & da_{12} & \dots & da_{1n} \\ da_{21} & da_{22} & \dots & da_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ da_{n1} & da_{n2} & \dots & da_{nn} \end{pmatrix} \right) = \sum_i da_{ii} = \text{左边} \end{aligned}$$

□

6.5.1 矩阵微分与偏导数的联系

多元函数的微分和偏导的具有如下关系

$$df(x_1, x_2, \dots, x_n) = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 + \dots + \frac{\partial f}{\partial x_n} dx_n$$

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 dx 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。

定理 6.5.2. 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.

$$\begin{aligned} \text{左边} &= df = \sum_{ij} \frac{\partial f}{\partial x_{ij}} dx_{ij} \\ \text{右边} &= \text{Tr} \left(\left(\frac{\partial f}{\partial A} \right)^T dA \right) \\ &= \sum_{ij} \left(\frac{\partial f}{\partial A} \right)_{ij} (dA)_{ij} \\ &= \sum_{ij} \frac{\partial f}{\partial x_{ij}} dx_{ij} = \text{左边} \end{aligned}$$

□

注意对于向量也有类似的结果。这里不再叙述。

6.5.2 关于逆矩阵的函数的微分

我们由单位矩阵的微分出发, 有

$$0 = d\mathbf{I} = d(\mathbf{XX}^{-1}) = d\mathbf{XX}^{-1} + \mathbf{X}d(\mathbf{X}^{-1})$$

$$d(\mathbf{X}^{-1}) = -\mathbf{X}^{-1}d\mathbf{XX}^{-1}$$

这样我们就得到了关于逆矩阵微分的一个结论。

例 6.5.1. 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

$$\frac{\partial \mathbf{x}^T \mathbf{A}^{-1} \mathbf{y}}{\partial \mathbf{A}} = -\mathbf{A}^{-T} \mathbf{x} \mathbf{y}^T \mathbf{A}^{-T}$$

例 6.5.2. 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.

$$\begin{aligned} f(\mathbf{X}) &= \text{Tr}(\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1}) \\ df(\mathbf{X}) &= \text{Tr}[d(\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1})] \\ &= \text{Tr}(d\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1} + \mathbf{X}^{-T} \mathbf{A}^T \mathbf{AdX}^{-1}) \\ &= \text{Tr}(2\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AdX}^{-1}) \\ &= \text{Tr}(-2\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1} d\mathbf{X}^{-1}) \\ &= \text{Tr}(-2\mathbf{X}^{-1} \mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1} d\mathbf{X}) \end{aligned}$$

故

$$\frac{\partial f}{\partial \mathbf{X}} = -2\mathbf{X}^{-T} \mathbf{A}^T \mathbf{AX}^{-1} \mathbf{X}^{-T}$$

6.5.3 关于行列式函数的微分

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 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 6.5.3. 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 6.4.7 逐元素进行求导。根据行列式的展开式, 易求得对第 i 行第 j 列元素 a_{ij} 的偏导数有

$$\frac{\partial |A|}{\partial a_{ij}} = \frac{\partial \left(\sum_j a_{ij} A_{ij} \right)}{\partial a_{ij}} = A_{ij}$$

使用定义 6.4.7 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 6.5.2, 我们还能够给出对应的微分关系

定理 6.5.4. 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 6.5.3. 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 6.5.4 有

$$\begin{aligned} d|A^{-1}| &= \text{Tr}(|A^{-1}| A dA^{-1}) \\ &= \text{Tr}(-|A^{-1}| A A^{-1} dA A^{-1}) \\ &= \text{Tr}(-|A^{-1}| A^{-1} dA) \end{aligned}$$

故

$$\frac{\partial |A^{-1}|}{\partial A} = -|A^{-1}| A^{-T} = -|A|^{-1} A^{-T}$$

6.6 迹函数的微分和迹微分法

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6.5.2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6.5.2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 6.6.1. 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 $\frac{\partial \text{Tr}(\mathbf{A})}{\partial \mathbf{A}} = \mathbf{I}$ 。

例 6.6.2. 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

$$\left(\frac{\partial \text{Tr}(\mathbf{AB})}{\partial \mathbf{A}} \right)_{ij} = \frac{\partial}{\partial A_{ij}} \left(\sum_{p=1}^m \sum_{l=1}^n A_{pl} B_{lp} \right) = B_{ji}$$

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 6.6.3. 由于 $\text{Tr}(\mathbf{xy}^T) = \text{Tr}(\mathbf{yx}^T) = \mathbf{x}^T \mathbf{y}$, 易知

$$\frac{\partial \text{Tr}(\mathbf{xy}^T)}{\partial \mathbf{x}} = \frac{\partial \text{Tr}(\mathbf{yx}^T)}{\partial \mathbf{x}} = \mathbf{y}$$

例 6.6.4. 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

$$\begin{aligned} df &= d \operatorname{Tr}(\mathbf{x}^T \mathbf{A} \mathbf{x}) \\ &= \operatorname{Tr}(d(\mathbf{x}^T \mathbf{A} \mathbf{x})) \\ &= \operatorname{Tr}(d(\mathbf{x}^T) \mathbf{A} \mathbf{x} + \mathbf{x}^T d(\mathbf{A} \mathbf{x})) \\ &= \operatorname{Tr}(d(\mathbf{x}^T) \mathbf{A} \mathbf{x} + \mathbf{x}^T d\mathbf{A} \mathbf{x} + \mathbf{x}^T \mathbf{A} d\mathbf{x}) \\ &= \operatorname{Tr}(d\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{x}^T \mathbf{A} d\mathbf{x}) \\ &= \operatorname{Tr}(d\mathbf{x}^T \mathbf{A} \mathbf{x}) + \operatorname{Tr}(\mathbf{x}^T \mathbf{A} d\mathbf{x}) \\ &= \operatorname{Tr}(\mathbf{x}^T \mathbf{A}^T d\mathbf{x}) + \operatorname{Tr}(\mathbf{x}^T \mathbf{A} d\mathbf{x}) \\ &= \operatorname{Tr}(\mathbf{x}^T \mathbf{A}^T d\mathbf{x} + \mathbf{x}^T \mathbf{A} d\mathbf{x}) \\ &= \operatorname{Tr}((\mathbf{x}^T \mathbf{A}^T + \mathbf{x}^T \mathbf{A}) d\mathbf{x}) \end{aligned}$$

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 6.6.5. 根据上面的推导可以知道, 在谱聚集中我们要求解的优化问题

$$\min_{\mathbf{x}} \mathbf{x}^T \mathbf{L} \mathbf{x}$$

$$\text{s.t. } \mathbf{x}^T \mathbf{1} = 0$$

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 6.6.6. 在 PCA 中, 我们需要求解优化问题

$$\begin{aligned} \min_{\mathbf{W}} & -\operatorname{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}) \\ \text{s.t. } & \mathbf{W}^T \mathbf{W} = \mathbf{I} \end{aligned}$$

我们现在考虑求梯度 $\nabla_{\mathbf{W}} -\operatorname{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W})$ 。

解. 利用迹微分法有

$$\begin{aligned} d(-\operatorname{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W})) &= -\operatorname{Tr}(d(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W})) \\ &= -2 \operatorname{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T d\mathbf{W}) \end{aligned}$$

所以 $\nabla_{\mathbf{W}} -\operatorname{Tr}(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W}) = -2\mathbf{X} \mathbf{X}^T \mathbf{W}$ 。

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 6.6.7. 设 $A \in \mathbb{R}^{n \times m}$, 求 $\frac{\partial \|A\|_F^2}{\partial A}$, 其中 $\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m a_{ij}^2}$

解.

$$\begin{aligned} d\|A\|_F^2 &= d\text{Tr}(A^T A) \\ &= \text{Tr}(d(A^T A)) \\ &= \text{Tr}((dA)^T A) + \text{Tr}(A^T dA) \\ &= \text{Tr}(2A^T dA) \end{aligned}$$

故

$$\frac{\partial \|A\|_F^2}{\partial A} = 2A$$

6.7 向量值函数和矩阵值函数的梯度

6.7.1 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 $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \in \mathbb{R}^m$ 。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:

$$\frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}^T} = \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_n} \right) = \begin{pmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

定义 6.7.1. 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:

$$\mathbf{J} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}^T} = \begin{pmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

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 $\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}$ 以及 $\frac{\partial \mathbf{f}(\mathbf{x})^T}{\partial \mathbf{x}^T}$, 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。

6.7.2 矩阵值函数的梯度

求矩阵关于向量或其它矩阵的梯度, 通常会导致一个多维张量。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 6.7.2. 函数 $\mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{nm}$ 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 6.7.3. 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 6.7.4. 设矩阵 \mathbf{J} 是一 Jacobian 矩阵, 则其行列式 $J = |\mathbf{J}|$ 称为 Jacobian 行列式。

6.7.3 向量值函数微分

定理 6.7.1. 设函数 $f(\mathbf{x}) : \mathbb{R}^m \rightarrow \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^m$ 则有 $df = \left(\frac{\partial f^T}{\partial \mathbf{x}} \right)^T d\mathbf{x} = \mathbf{J}\mathbf{x}$ 。

证明. 显然, $\mathbf{d}\mathbf{f}$ 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。

$$\begin{aligned} \text{左边}_j &= \mathbf{d}f_j = \sum_{i=1}^m \frac{\partial f_j}{\partial x_i} dx_i \\ \text{右边}_j &= \left(\left(\frac{\partial f}{\partial \mathbf{x}} \right)^T dx \right)_j \\ &= \sum_{i=1}^m \left(\frac{\partial f}{\partial \mathbf{x}} \right)_{ij} dx_i \\ &= \sum_{i=1}^m \left(\frac{\partial f_j}{\partial x_i} \right) dx_i = \text{左边}_j \end{aligned}$$

□

注意这个式子在形式上与之前我们推得的定理 6.5.2 是很像的。

利用定理 6.7.1, 仿照求解实值函数梯度的步骤, 可以简化求解向量对向量的导数。

例 6.7.1. 考虑向量变换 $\mathbf{x} = \sigma \mathbf{\Lambda}^{-0.5} \mathbf{W}^T \boldsymbol{\eta}$, \mathbf{x} 和 $\boldsymbol{\eta}$ 的维数是 d , 其中 σ 是一个实变量, $\mathbf{\Lambda}$ 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。

$$d\mathbf{x} = d(\sigma \mathbf{\Lambda}^{-0.5} \mathbf{W}^T \boldsymbol{\eta}) = \sigma \mathbf{\Lambda}^{-0.5} \mathbf{W}^T d\boldsymbol{\eta}$$

应用定理 6.7.1 我们有

$$\mathbf{J} = \left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\eta}} \right)^T = \sigma \mathbf{\Lambda}^{-0.5} \mathbf{W}^T$$

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。

$$\begin{aligned} |J| &= |\det(\mathbf{J})| \\ &= \sqrt{|\det(\mathbf{J})| |\det(\mathbf{J})|} \\ &= \sqrt{|\det(\mathbf{J})| |\det(\mathbf{J}^T)|} \\ &= \sqrt{|\det(\mathbf{J}^T \mathbf{J})|} \\ &= \sqrt{|\det(\mathbf{W} \mathbf{\Lambda}^{-0.5} \sigma \sigma \mathbf{\Lambda}^{-0.5} \mathbf{W}^T)|} \\ &= \sqrt{|\det(\sigma^2 \mathbf{W} \mathbf{\Lambda}^{-1} \mathbf{W}^T)|} \end{aligned}$$

我们令 $\Sigma = \mathbf{W} \mathbf{\Lambda} \mathbf{W}^T$ 。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 6.7.2. 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 6.7.1

$$df = \left(\frac{\partial f^T}{\partial x} \right)^T dx \Rightarrow \left(\left(\frac{\partial f^T}{\partial x} \right)^T \right)^{-1} df = dx \Rightarrow dx = \left(\left(\frac{\partial f^T}{\partial x} \right)^{-1} \right)^T df$$

所以, 我们就有 $\frac{\partial x^T}{\partial f} = \left(\frac{\partial f^T}{\partial x} \right)^{-1}$ 。

□

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。

6.8 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 6.8.1. 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

$$\begin{aligned} \frac{\partial z}{\partial x} &= \left(\frac{\partial y_1}{\partial x}, \frac{\partial y_2}{\partial x} \right) \begin{pmatrix} \frac{\partial z}{\partial y_1} \\ \frac{\partial z}{\partial y_2} \end{pmatrix} \\ &= \left(\cos x - x \sin x, \sin x + x \cos x \right) \begin{pmatrix} y_2^2 e^{y_1 y_2^2} \\ 2y_1 y_2 e^{y_1 y_2^2} \end{pmatrix} \\ &= (y_2^2 (\cos x - x \sin x) + 2y_1 y_2 (\sin x + x \cos x)) e^{y_1 y_2^2} \end{aligned}$$

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

$$\frac{\partial z}{\partial x} = \frac{\partial \mathbf{y}^T}{\partial x} \frac{\partial z}{\partial \mathbf{y}}$$

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 6.8.1. 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

$$\frac{\partial (\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(1)}} = \frac{\partial (\mathbf{x}^{(2)})^T}{\partial \mathbf{x}^{(1)}} \frac{\partial (\mathbf{x}^{(3)})^T}{\partial \mathbf{x}^{(2)}} \cdots \frac{\partial (\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(n-1)}}$$

证明. 根据向量值函数梯度的定义6.7.1和向量值函数微分定理6.7.1, 将定理6.7.1应用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

$$\begin{aligned} d\mathbf{x}^{(n)} &= \left(\frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(n-1)}} \right)^T \dots \left(\frac{\partial(\mathbf{x}^{(3)})^T}{\partial \mathbf{x}^{(2)}} \right)^T \left(\frac{\partial(\mathbf{x}^{(2)})^T}{\partial \mathbf{x}^{(1)}} \right)^T d\mathbf{x}^{(1)} \\ &= \left(\frac{\partial(\mathbf{x}^{(2)})^T}{\partial \mathbf{x}^{(1)}} \frac{\partial(\mathbf{x}^{(3)})^T}{\partial \mathbf{x}^{(2)}} \dots \frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(n-1)}} \right)^T d\mathbf{x}^{(1)} \end{aligned}$$

再次应用定理6.7.1可得

$$\frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(1)}} = \frac{\partial(\mathbf{x}^{(2)})^T}{\partial \mathbf{x}^{(1)}} \frac{\partial(\mathbf{x}^{(3)})^T}{\partial \mathbf{x}^{(2)}} \dots \frac{\partial(\mathbf{x}^{(n)})^T}{\partial \mathbf{x}^{(n-1)}}$$

□

例 6.8.2. 考虑线性回归中的优化问题: $\min_{\theta} \sum_{i=1}^n (\theta^T \mathbf{x}_i - y_i)^2$ 。我们将其目标函数改写成 $\|\mathbf{X}\theta - \mathbf{y}\|_2^2$ 并关于 θ 求梯度, 其中 $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T$, $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$, 由链式法则我们有

$$\begin{aligned} &\nabla_{\theta} \|\mathbf{X}\theta - \mathbf{y}\|_2^2 \\ &= \frac{\partial(\mathbf{X}\theta - \mathbf{y})^T}{\partial \theta} \frac{\partial \|\mathbf{z}\|_2^2}{\partial \mathbf{z}}, \quad \text{其中 } \mathbf{z} = \mathbf{X}\theta - \mathbf{y} \\ &= \mathbf{X}^T \frac{\partial \mathbf{z}^T \mathbf{z}}{\partial \mathbf{z}} \\ &= 2\mathbf{X}^T \mathbf{z} \\ &= 2\mathbf{X}^T \mathbf{X}\theta - \mathbf{X}^T \mathbf{y} \end{aligned}$$

例 6.8.3. 计算 $(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$ 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

$$\begin{aligned} &\frac{\partial ((\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}))}{\partial \boldsymbol{\mu}} \\ &= \frac{\partial[(\mathbf{x} - \boldsymbol{\mu})^T]}{\partial \boldsymbol{\mu}} \frac{\partial ((\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}))}{\partial [\mathbf{x} - \boldsymbol{\mu}]} \\ &= \frac{\partial[(\mathbf{x} - \boldsymbol{\mu})^T]}{\partial \boldsymbol{\mu}} 2\boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \\ &= -I2\boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \\ &= -2\boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \end{aligned}$$

6.9 反向传播和自动微分

6.9.1 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

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos(x^2 + \exp(x^2)). \quad (6.10)$$

应用链式法则，注意微分是线性的，计算梯度。

$$\begin{aligned} \frac{df}{dx} &= \frac{2x + 2x \exp(x^2)}{2\sqrt{x^2 + \exp(x^2)}} - \sin(x^2 + \exp(x^2))(2x + 2x \exp(x^2)) \\ &= 2x\left(\frac{1}{2\sqrt{x^2 + \exp(x^2)}} - \sin(x^2 + \exp(x^2))\right)(1 + \exp(x^2)). \end{aligned}$$

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 \mathbf{y} 是函数深度复合来进行计算的。

$$\mathbf{y} = (f_K \circ f_{K-1} \circ \cdots \circ f_1)(\mathbf{x}) = f_K(f_{K-1}(\cdots(f_1(\mathbf{x}))\cdots)), \quad (6.11)$$

其中 \mathbf{x} 是输入（例如，图像）， \mathbf{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 $f_i(\mathbf{x}_{i-1}) = \sigma(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i)$ 。其中 \mathbf{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 $\mathbf{A}_j, \mathbf{b}_j, j = 1, \dots, K$ 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 \mathbf{x} 和观测 y ，那么网络结构定义为

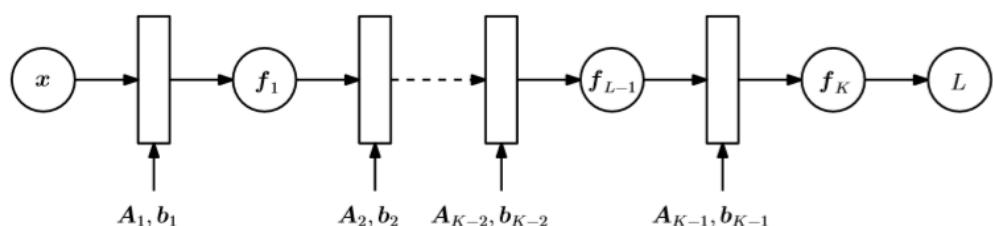


图 6.16: 多层神经网络中的正向传播，用于计算作为输入 \mathbf{x} 和参数 $\mathbf{A}_i, \mathbf{b}_i$ 的函数的损失 L .

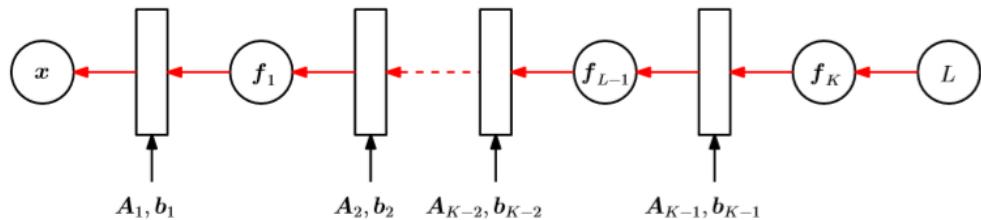


图 6.17: 三阶张量的 3-模式向量积的原理图

观察 y 和由网络结构图6.16的定义

$$\begin{aligned} f_0 &:= x \\ f_i &:= \delta_i(\mathbf{A}_{i-1}f_{i-1} + \mathbf{b}_{i-1}), i = 1, \dots, K, \end{aligned} \quad (6.12)$$

考虑平方损失 $L(\theta) = \|y - f_K(\theta, x)\|^2$ 的最小化问题, 其中 $\theta = \{\mathbf{A}_0, \mathbf{b}_0, \dots, \mathbf{A}_{K-1}, \mathbf{b}_{K-1}\}$ 。我们需要得到关于参数集 θ 的梯度。为此, 我们需要 L 关于每层参数 $\theta_j = \{\mathbf{A}_j, \mathbf{b}_j\}$ 的 ($j = 0, \dots, K-1$) 偏导数。由链式法则可得

$$\frac{\partial L}{\partial \theta_{K-1}} = \frac{\partial L}{\partial f_K} \frac{\partial f_K}{\partial f_{K-1}} \quad (6.13)$$

$$\frac{\partial L}{\partial \theta_{K-2}} = \frac{\partial L}{\partial f_K} \frac{\partial f_K}{\partial f_{K-1}} \frac{\partial f_{K-1}}{\partial f_{K-2}} \quad (6.14)$$

$$\frac{\partial L}{\partial \theta_{K-3}} = \frac{\partial L}{\partial f_K} \frac{\partial f_K}{\partial f_{K-1}} \frac{\partial f_{K-1}}{\partial f_{K-2}} \frac{\partial f_{K-2}}{\partial f_{K-3}} \quad (6.15)$$

$$\frac{\partial L}{\partial \theta_i} = \frac{\partial L}{\partial f_K} \frac{\partial f_K}{\partial f_{K-1}} \dots \frac{\partial f_{i+2}}{\partial f_{i+1}} \frac{\partial f_{i+1}}{\partial \theta_i} \quad (6.16)$$

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6.17显示了通过网络向后传播的过程。

在了解了什么是反向传播后, 我们考虑为什么需要反向传播算法。我们从以下两个问题出发。

1. 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 $\mathbf{AB} = (m \times n)(n \times p)$ 需要 mnp 次乘法, $(\mathbf{AB})\mathbf{C} = (m \times p)(p \times q)$ 需要 mpq 次乘法。
- 第二种方式 $\mathbf{BC} = (n \times p)(p \times q)$ 需要 nq 次乘法, $\mathbf{A}(\mathbf{BC}) = (m \times n)(n \times q)$ 需要 mnq 次乘法。

因此我们比较 $mp(n + q)$ 和 $nq(m + p)$, 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 $\frac{\partial \mathbf{v}_i^T}{\partial (\mathbf{A}_i)}$ 。

6.9.2 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 6.18: 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

在图6.18表示的计算图中，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：

$$\frac{dy}{dx} = \frac{dy}{db} \frac{db}{da} \frac{da}{dx} \quad (6.17)$$

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(6.18)或(6.19)。

$$\frac{dy}{dx} = \left(\frac{dy}{db} \frac{db}{da} \right) \frac{da}{dx} \quad (6.18)$$

$$\frac{dy}{dx} = \frac{dy}{db} \left(\frac{db}{da} \frac{da}{dx} \right) \quad (6.19)$$

等式(6.18)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(6.19)将是正向模式，其中梯度随着数据从左到右流过图。

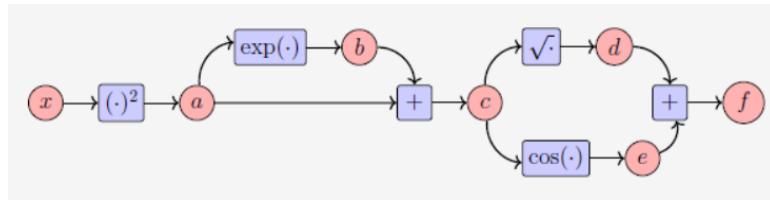
在下文中，我们将重点关注反向模式自动微分，即反向传播。在神经网络的背景下，输入维度通常远高于标签维数，反向模式在计算上比正向模式容易得多。让我们从一个有教育意义的例子开始。

例 6.9.1. 从(6.10)中考虑函数

$$f(x) = \sqrt{x^2 + \exp(x^2)} + \cos(x^2 + \exp(x^2)) \quad (6.20)$$

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：

$$\begin{aligned} a &= x^2 \\ b &= \exp(a) \\ c &= a + b \\ d &= \sqrt{c} \\ e &= \cos(c) \\ f &= d + e \end{aligned} \quad (6.21)$$

图 6.19: 输入 x , 函数 f 以及中间变量为 a, b, c, d, e 的计算图。

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(6.10)中定义的函数 $f(x)$ 所需的操作要少。图6.19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:

数据科学与工程数学基础初稿

$$\begin{aligned}
 \frac{\partial a}{\partial x} &= 2x \\
 \frac{\partial b}{\partial a} &= \exp(a) \\
 \frac{\partial c}{\partial a} &= 1 = \frac{\partial c}{\partial b} \\
 \frac{\partial d}{\partial c} &= \frac{1}{2\sqrt{c}} \\
 \frac{\partial e}{\partial c} &= -\sin(c) \\
 \frac{\partial f}{\partial d} &= 1 = \frac{\partial f}{\partial e}
 \end{aligned} \tag{6.22}$$

通过看图6.19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:

$$\frac{\partial f}{\partial c} = \frac{\partial f}{\partial d} \frac{\partial d}{\partial c} + \frac{\partial f}{\partial e} \frac{\partial e}{\partial c} \tag{6.23}$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} \frac{\partial c}{\partial b} \tag{6.24}$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} \frac{\partial b}{\partial a} + \frac{\partial f}{\partial c} \frac{\partial c}{\partial a} \tag{6.25}$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a} \frac{\partial a}{\partial x} \tag{6.26}$$

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

$$\frac{\partial f}{\partial c} = 1 \cdot \frac{1}{2\sqrt{c}} + 1 \cdot (-\sin(c)) \tag{6.27}$$

$$\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} \cdot 1 \quad (6.28)$$

$$\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} \exp(a) + \frac{\partial f}{\partial c} \cdot 1 \quad (6.29)$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a} \cdot 2x \quad (6.30)$$

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。

例 6.9.2. 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

$$\frac{\partial L}{\partial \mathbf{b}_1} = \frac{\partial \mathbf{k}^T}{\partial \mathbf{b}_1} \frac{\partial \mathbf{y}^T}{\partial \mathbf{k}} \frac{\partial L}{\partial \mathbf{y}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ -2 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\frac{\partial L}{\partial \mathbf{b}_2} = \frac{\partial \mathbf{y}^T}{\partial \mathbf{b}_2} \frac{\partial L}{\partial \mathbf{y}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

6.10 高阶微分和泰勒展开

6.10.1 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 6.10.1. 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

$$\frac{\partial^2 \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x} \partial \mathbf{x}^T} = \mathbf{O}_{n \times n} \quad (6.31)$$

- 若 \mathbf{A} 是 $n \times n$ 矩阵，则

$$\frac{\partial^2 \mathbf{x}^T \mathbf{A} \mathbf{x}}{\partial \mathbf{x} \partial \mathbf{x}^T} = \mathbf{A} + \mathbf{A}^T \quad (6.32)$$

- 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

$$\frac{\partial^2 (\mathbf{a} - \mathbf{A} \mathbf{x})^T \mathbf{B} (\mathbf{a} - \mathbf{A} \mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} = 2 \mathbf{A}^T \mathbf{B} \mathbf{A} \quad (6.33)$$

- 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

$$\begin{aligned} \frac{\partial^2 (\mathbf{y}(\mathbf{x}))^T \mathbf{A} \mathbf{y}(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} &= \frac{\partial [\mathbf{y}(\mathbf{x})]^T}{\partial \mathbf{x}} (\mathbf{A} + \mathbf{A}^T) \frac{\partial \mathbf{y}(\mathbf{x})}{\partial \mathbf{x}^T} + \\ &((\mathbf{y}(\mathbf{x}))^T (\mathbf{A} + \mathbf{A}^T) \otimes \mathbf{I}_n) \frac{\partial}{\partial \mathbf{x}^T} \left(\frac{\partial [\mathbf{y}(\mathbf{x})]^T}{\partial \mathbf{x}} \right) \end{aligned} \quad (6.34)$$

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。

6.10.2 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

$$f(\mathbf{x}) \approx f(\mathbf{x}_0) + (\nabla \mathbf{x} f)^T(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

这里 $(\nabla \mathbf{x} f)^T(\mathbf{x}_0)$ 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 6.10.2. 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 6.10.3. 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

$$D_x^k f(\mathbf{x}_0) \Delta^k = \sum_a \dots \sum_k D_x^k f(\mathbf{x}_0)[a, \dots, k] \delta[a] \dots \delta[k]$$

其中

$$D_x^k f(\mathbf{x})[i_1, \dots, i_k] = \frac{\partial^k}{\partial x_{i_1} \dots \partial x_{i_k}} f(\mathbf{x})$$

所以 $D_x^k f(\mathbf{x}_0) \Delta^k$ 包含了所有 k 次多项式。

$$k = 0: D_x^0 f(\mathbf{x}_0) \Delta^0 = f(\mathbf{x}_0) \in \mathbb{R}$$

$$k = 1: D_x^1 f(\mathbf{x}_0) \Delta^1 = \nabla_x f(\mathbf{x}_0) \Delta = \sum_i \nabla f(\mathbf{x}_0)[i] \delta[i] \mathbb{R}$$

$$k = 2: D_x^2 f(\mathbf{x}_0) \Delta^2 = \Delta^T H \Delta = \sum_i \sum_j H[i, j] \delta[i] \delta[j] \in \mathbb{R}$$

$$k = 3: D_x^3 f(\mathbf{x}_0) \Delta^3 = \sum_i \sum_j \sum_k D_x^3 f(\mathbf{x}_0)[i, j, k] \delta[i] \delta[j] \delta[k] \in \mathbb{R}$$

例 6.10.1. 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

$$T_2 = f(\mathbf{0}) + (\nabla_{\mathbf{x}} f(\mathbf{0}))^T (\mathbf{x} - \mathbf{0}) + \frac{1}{2} (\mathbf{x} - \mathbf{0})^T (\nabla_{\mathbf{x}}^2 f(\mathbf{0})) (\mathbf{x} - \mathbf{0})$$

通过计算可得

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = (\mathbf{a}_1 e^{\mathbf{x}_1}, \mathbf{a}_2 e^{\mathbf{x}_2}, \dots, \mathbf{a}_n e^{\mathbf{x}_n})^T$$

$$\nabla_{\mathbf{x}}^2 f(\mathbf{x}) = \text{diag}(\nabla_{\mathbf{x}} f(\mathbf{x}))$$

所以 $f(\mathbf{0}) = \sum_{i=1}^n \mathbf{a}_i$, $\nabla_{\mathbf{x}} f(\mathbf{0}) = \mathbf{a}$, $\nabla_{\mathbf{x}}^2 f(\mathbf{0}) = \text{diag}(\mathbf{a})$

故 $T_2 = \sum_{i=1}^n \mathbf{a}_i + \mathbf{a}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \text{diag}(\mathbf{a}) \mathbf{x} = \sum_{i=1}^n \mathbf{a}_i (1 + \mathbf{x}_i + \frac{1}{2} \mathbf{x}_i^2)$

6.11 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

$$E[f(\mathbf{x})] = \int f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \quad (6.35)$$

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 $p(\mathbf{x})$ 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 $\mathbf{x}_0 = 0$ 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

$$f_1(\mathbf{x}) = \sin(\mathbf{x}_1) \cos(\mathbf{x}_2), \mathbf{x} \in \mathbb{R}^2$$

$$f_1(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$$

$$f_1(\mathbf{x}, \mathbf{y}) = \mathbf{x} \in \mathbb{R}^n$$

1. $\frac{\partial f_1}{\partial \mathbf{x}}$ 的维数是多少?

2. 计算雅克比矩阵。

习题 6.6. f 对 t 求导, g 对 \mathbf{X} 求导, 其中

$$f(\mathbf{t}) = \sin(\log(\mathbf{t}^T \mathbf{t})), \mathbf{t} \in \mathbb{R}^D$$

$$g(\mathbf{X}) = \text{Tr}(\mathbf{A} \mathbf{X} \mathbf{B}), \mathbf{A} \in \mathbb{R}^{D \times E}, \mathbf{X} \in \mathbb{R}^{E \times F}, \mathbf{B} \in \mathbb{R}^{F \times D}$$

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.

$$f(z) = \sin(z), z = \mathbf{A} \mathbf{x} + b, \mathbf{A} \in \mathbb{R}^{E \times D}, \mathbf{x} \in \mathbb{R}^D, b \in \mathbb{R}^E$$

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。

$$f(z) = \exp\left(-\frac{1}{2}z\right)$$

$$z = g(\mathbf{y}) = \mathbf{y}^T \mathbf{S}^{-1} \mathbf{y}$$

$$\mathbf{y} = h(\mathbf{x}) = \mathbf{x} - \boldsymbol{\mu}$$

其中 $\mathbf{x}, \boldsymbol{\mu} \in \mathbb{R}^D, \mathbf{S} \in \mathbb{R}^{D \times D}$ 。

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。

$$f = \tanh(\mathbf{z}) \in \mathbb{R}^M$$

$$\mathbf{z} = \mathbf{A}\mathbf{z} + \mathbf{b}, \mathbf{x} \in \mathbb{R}^N, \mathbf{A} \in \mathbb{R}^{M \times N}, \mathbf{b} \in \mathbb{R}^M$$

这里的 \tanh 作用于 \mathbf{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 l_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 $\frac{\partial \mathbf{x}^T \mathbf{A} \mathbf{x}}{\partial \mathbf{x}}$, $\frac{\partial \mathbf{x}^T \mathbf{A} \mathbf{x}}{\partial \mathbf{A}}$

习题 6.12. $f(\mathbf{z}) = \frac{\exp(\mathbf{z})}{\mathbf{1}^T \exp(\mathbf{z})}$ 称为 softmax 函数, $(\exp(\mathbf{z}))_i = \exp(z_i)$, 如果 $\mathbf{q} = \frac{\exp(\mathbf{z})}{\mathbf{1}^T \exp(\mathbf{z})}$, $J = -\mathbf{p}^T \log(\mathbf{q})$, 其中 $\mathbf{p}, \mathbf{q}, \mathbf{z} \in \mathbb{R}^n$, 并且 $\mathbf{1}^T \mathbf{p} = 1$

- 证: $\frac{\partial J}{\partial \mathbf{z}} = \mathbf{q} - \mathbf{p}$
- 若 $\mathbf{z} = \mathbf{Wx}$, 其中 $\mathbf{W} \in \mathbb{R}^{n \times m}, \mathbf{x} \in \mathbb{R}^m$, $\frac{\partial J}{\partial \mathbf{W}} = (\mathbf{q} - \mathbf{p})\mathbf{x}^T$ 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 $\frac{\partial |\mathbf{X}^k|}{\partial \mathbf{X}}$ 。

习题 6.15. 求 $\frac{\partial \text{Tr}(\mathbf{AXBX}^T \mathbf{C})}{\partial \mathbf{X}}$ 。

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

数据科学与工程数学基础初稿