基于R应用的统计学丛书

多元统计分析

——R与Python的实现

Multivariate Statistical Analysis with R and Python

吴喜之 编著

k-Means Clustering of 2 Colours

k-Means Clustering of 4 Colours

k-Means Clustering of 16 Colours

k-Means Clustering of 32 Colours

中国人民大学出版社

多元统计分析

——R与Python的实现

Multivariate Statistical Analysis with R and Python

吴喜之 编著

中国人民大学出版社
·北京·

图书在版编目 (CIP) 数据

多元统计分析: R 与 Python 的实现 / 吴喜之编著. —北京: 中国人民大学出版社, 2019.1
(基于 R 应用的统计学丛书)
ISBN 978-7-300-26655-8

I. ①多 … II. ①吴 … III. 多元分析 – 统计分析 IV.①O212.4

中国版本图书馆 CIP 数据核字 (2019) 第 016562 号

基于 R 应用的统计学丛书

多元统计分析——R 与 Python 的实现

吴喜之　编著

Duoyuan Tongji Fenxi——R yu Python de Shixian

出版发行	中国人民大学出版社		
社　　址	北京中关村大街 31 号	**邮政编码**	100080
电　　话	010-62511242(总编室)	010-62511398(质管部)	
	010-82501766(邮购部)	010-62514148(门市部)	
	010-62515195(发行公司)	010-62515275(盗版举报)	
网　　址	http://www.crup.com.cn		
	http://www.ttrnet.com (人大教研网)		
经　　销	新华书店		
印　　刷	北京鑫丰华彩印有限公司		
规　　格	185mm× 260mm　16 开本	**版　　次**	2019 年 1 月第 1 版
印　　张	20.75 插页 1	**印　　次**	2019 年 1 月第 1 次印刷
字　　数	484 000	**定　　价**	39.80 元

前 言

目前, 多数国内多元分析教材有下面几个特点: (1) 着重于数学理论和推导. 这往往使得读者忘记了分析数据的本来目的, 把数据科学当成数学来学. (2) 关于多元正态分布的性质占据很大篇幅. 这实际上过分强调了对数据的狭隘数学假定, 而忽略了真实数据的复杂性和多样性. (3) 对真实的数据分析强调得不够. 一些典型的教材数据往往和现实世界差得很远, 对读者具有误导性. (4) 在所涉及的统计软件应用方面, 大多只限于商业软件. 笔者认为, 使用盗版国外软件不仅犯法, 而且最终会形成对这些盗版软件的依赖性而不能自拔. 使用 "傻瓜式" 商业软件对于学习编程没有任何好处, 反而会阻碍人们获得编程能力, 而是否会编程将如是否文盲一样影响人们的未来发展.

本书力图用简单通俗的语言阐述有关的基本概念, 使得非数学背景的读者亦可以较容易地理解. 考虑到许多读者的主要目标是应用, 本书通过案例来帮助理解各种方法的概念及计算. 本书也包括了传统多元分析课本所具有的数学公式, 目的在于方便读者查询.

本书基于实际数据分析, 在正文中通过 R 软件解释方法及计算输出, 同时在每一章都给出用 Python 3 计算这一章数据例子的代码.

本书包括传统多元分析教材通常包括的内容: 主成分分析、因子分析、聚类分析、判别分析、典型相关分析以及非传统的对应分析方法. 一些多元分析教材还包括完整的多重回归内容, 但这远远超出一个学期的课程内容. 本书对多元回归予以回顾, 介绍了部分机器学习回归方法, 判别分析则安排在 "分类" 一章, 并且介绍了部分机器学习分类方法. 关于回归和分类的更多内容及细节, 请读者参看笔者编写的由中国人民大学出版社出版的《应用回归及分类——基于 R》一书.

应用统计方法的学习, 应该是拉动式学习, 也就是遵从问题驱动或者数据驱动模式: 从数据出发, 需要什么就学什么. 如此, 所有学习都是目标明确的, 学的都是有用的知识和能力. 有的教师喜欢以 "打基础" 的名义给学生一大堆书去读, 这样的教学方法事倍功半.

和其他自然科学一样, 统计需要从数学课程中获得严密的逻辑思维能力, 还需要计算机编程能力. 这不仅需要学会针对一两个统计软件的编程, 还需要获得能迅速掌握任何开源软件的**泛型编程能力**. 有了来自数学的思维能力、想象力和逻辑, 以及计算机编程能力, 再加上学习自然科学所必需的批判性思维及对所面对实际问题的理解, 你就拥有了研究统计最重要的必要条件.

本书的排版是笔者用 LaTeX 软件实现的, 一切错误由笔者负责.

吴喜之

目 录

第 1 章 引 言

1.1 数据科学

1.1.1 统计是数据科学吗?

按照《不列颠百科全书》所述, **统计是收集、分析、展示及解释数据的科学** (Statistics, the science of collecting, analyzing, presenting, and interpreting data).[1] 也就是说, 统计是**数据科学** (the science of data). 早在 20 年前, 华裔统计学家吴建福 (C.F. Jeff Wu) 就多次说过 "统计应该称为数据科学".

但是, 无论口头如何说, 实际上统计界许多人是在搞数学, 并冠之以 "理论统计" 或 "数理统计" 的名字. 这是有历史原因的. 计算机出现之前, 人们只能处理少量数据, 为了计算方便和弥补信息量的不足, 以数学家为主体的学者创造了很多基于主观假定的数学模型. 统计因此被数学所主导, 统计方面的研究文章大部分是数学研究. 《中国大百科全书》明确定义, 数理统计是 "数学的一个分支学科".[2]

事实上, 在前计算机时代, 数据分析一直依赖于这些数学家所创造的数学模型. 因此, 模型驱动而非问题驱动一直是统计界的主流思维方式, 模型驱动思维就是数学式的演绎思维. 一直到 20 世纪 90 年代, 计算机还仅仅是各个领域数据分析者的工具, 是手和笔的延伸. 如果在那个时候, 统计学家认为自己是数据科学家, 而把计算机视为自己不可缺少的工具包括在数据科学之内, 统计学的进程就完全不一样了. 可是, 以数学模型为主导的统计界没有这么做. 当今, 统计界无法和真正的数学界比数学之美, 也无法和计算机群体比处理数据问题的效率和能力, 统计失去了很多热门领域和优秀人才.

机器学习或者基于机器学习方法的人工智能目前发展迅速, 但有多少人会把这些发展和本应是 "数据科学" 的传统数理统计相联系呢? 尽管现在一部分 "大数据中心"(或 "研究院""学院" 等) 的领导者都是专注于数学模型的学者, 但那些计算机领域的人会臣服吗?

从深度、难度以及美学的角度来看, 统计所用的数学和纯粹数学根本无法相比, 与数学家在一起可能会令某些统计学人感到不舒服. 其实, 如果这些统计学人展示的不是数学而是分析数据的能力, 就不会有这种感觉了, 问题是这些统计学人究竟有没有数据处理的能力. 类似地, 部分统计学人和搞计算机的同事在一起时也会有不快的感觉. 如果他们不会编程计算却又专注于一些没有实践背景的数学模型, 自然会感到不爽.

对于整个社会来说, 传统统计的边缘化并不见得是个坏事情. 对于统计学科本身, 危机感能够促进变化, 把统计界从封闭的小圈子里解放出来.

---

[1] 不列颠百科全书. https://www.britannica.com/science/statistics.

[2] 中国大百科全书总编辑委员会. 中国大百科全书: 数学卷. 北京: 中国大百科全书出版社, 1988: 593.

1.1.2 计算机学科在数据科学中的地位

由于统计界的主流实际上不愿意采取问题驱动或数据驱动的思维, 而计算机学科借助数据源、数据清理、数据处理及算法改进等方面的优势, 已经成为数据科学的主要实体, 而且占有越来越重要的地位.

实际上, 主要的机器学习方法的创造者大都不是沉溺于数学模型的 "统计学家", 也不是计算机界的学者. 他们多是有数学背景或其他领域背景的科学家. 他们依赖的并不是某些专业课程知识, 而是自己对实际问题的理解、对随机性的理解、抽象思维能力和想象力, 而这些思维能力很多是从严格的纯粹数学训练中获得的.

在当今世界, 人们离不开计算机, 计算机是数据科学最重要的基石之一, 也是工作量最大的部分, 没有计算机, 所有的机器学习、人工智能、大数据分析就都没有意义. 但是必须清醒地认识到, **单纯的 "软件 + 硬件" 能力不会自动产生新的算法, 工程师式的工作代替不了科学家的创造**.

1.1.3 问题驱动应成为数据科学的基本思维方式

笔者觉得目前的统计学科、数学或计算机学科没有一个能够单独代表数据科学. 数据科学本身包括了所有可能的知识积累和人们的才智. **知识不是由学科划分的**, 当前科技进步的进程不断突破人们有限的想象力, 试图当老大或者试图把学科标签固化的思维曾经阻碍了一些学科的发展, 一些故步自封的学科将来也会消亡, 但社会总是在不断前进.

很多人认为, 学完了统计、数学、计算机及各领域的一些课程就可以解决数据科学的问题. 实际上, 课程仅仅是历史的一些记录, 人们可以从中得到一些经验、启发及知识, 而不可能得到能力. 有人觉得, 根据分析某些 "教科书式的数据" 的经验, 按照现成模型可能会找到所谓 "标准答案", 但是, **在实际问题中不可能存在 "标准答案"**, 针对一个问题可能有很多不同的答案, 而每种答案都可能具有某种合理性. 试图寻求一个 "标准答案" 本身就限制了人们的想象力、能力及对实际问题复杂性的判断. 更为大量的实际问题是无法限定于一种或若干种方法 (或组合) 的, 认为只要选择合适的模型和参数就应该得到 "正确" 或 "精准" 答案的想法是把世界简单化, 真实世界比围棋机器人所面对的围棋世界要复杂得多.

因此, 面对复杂的真实问题, 我们对问题本身的意义及有关数据的理解、知识积累、想象力和处理复杂问题的能力就是关键. 能力是从不断解决实际问题中获得的, 是从无数失败中悟出来的, 而不是从书本上学来的.

1.2 多元分析的对象

本书介绍的方法主要适用于规范的横截面数据, 观测值及变量可以按照行列排成方阵. 这里讨论的变量都是相关的, 而且每个观测值都是同时度量的. 根据机器学习术语, 回归和分类属于**有指导 (或有监督) 学习** (supervised learning), 也就是有因变量作为目标变量, 主要目的是建立可以预测的模型, 有指导学习的回归和分类是机器学习的核心, 也是人工智能的基本组成部分; 其他方法都属于**无指导 (或无监督) 学习** (unsupervised learning), 其中没有目标变量, 目的是揭示相关变量所提供的潜在重叠信息所代表数据的背景结构, 因此, 简化、降维、汇总是其主要特点. 这部分多元统计分析基本上是探索性的数据分析, 它可能提出一些假设, 但检验往往并不是其主要目的.

从名称上看, 只要多于一个变量的数据分析都应该属于多元分析. 这实际上包括了除收集数据之外的几乎整个数据科学的内容. 但是, 传统的多元分析要狭窄得多, 除了往往放在回归课程中讲授的回归分析之外, 多元分析传统上是由无指导学习中的一组方法所构成的. 这包括主成分分析、因子分析、聚类分析、典型相关分析和判别分析等, 这些方法传统上多假定多元正态分布, 但由于大多是描述性的, 在方法及结果的解释方面并不完全依赖分布假定. 再拓宽一些, 还可能包括本书没有包括在内的多元方差分析及结构方程模型的协方差方法等, 它们严重依赖于对数据的各种假定.

除了可以应用于所有领域并成为人工智能基础的有指导学习之外, 属于无指导学习的经典多元分析技术的很大一部分最早应用于行为科学和生物科学, 而现在已经扩展到各个领域, 包括物理、化学、生物、医学、心理学、社会学、经济学、教育学、法律、商学、文学、传媒学、护理学、语言学、地质学等等.

读者可以通过本书理解各种多元分析的方法、目的、适用范围、局限性及数据分析的具体计算机实现, 以及各种方法的关联及区别. 在动手操作方面, 期望读者能够根据具体数据选择合适的方法并且解释多元分析方法的计算机输出. 希望读者能够通过编程来熟悉本书的概念, 同时通过分析数据来熟悉编程.

1.3 需要的工具

学习本书, 需要掌握一定的数学知识, 主要是线性代数或矩阵理论等数学, 这些知识在本书的第 2 章专门进行了回顾, 可以自给自足; 微积分用得不多 (只有个别地方应用了导数的概念). 读者有一些初等统计知识就够了. 本书强调对模型和方法的理解, 而不是数学证明和推理的细节, 大多数被展示的结果未给出证明, 因为直观的解释更加有说服力. 本书所使用的 R 或者 Python 等编程语言完全可以在使用中学习. 学习软件最快的方法就是使用, 有问题可从网络上寻求答案, 不必专门在课堂上学习.

1.4 各章的安排

本书从第 2 章开始的各章内容安排如下: 第 2 章 “矩阵代数回顾”, 提供本书所需要的一些代数知识, 供需要时查阅, 这些内容一般不用在课堂上专门介绍 (由于后续各章多少涉及该章内容, 所以没有打星号). 第 3、4 章的内容为有指导学习: 第 3 章 “回归”, 回顾经典回归分析的内容, 同时简单介绍少数机器学习的回归方法, 这一章可以自学, 也可以在课堂上做介绍, 该章的多元多重回归一节仅供参考之用; 第 4 章 “分类”, 介绍经典判别分析, 同时介绍少数机器学习分类方法. 从第 5 章开始介绍无指导学习的概念、方法、目的及案例分析, 包括第 5 章 “主成分分析”、第 6 章 “因子分析”、第 7 章 “聚类分析”、第 8 章 “典型相关分析”、第 9 章 “对应分析”、第 10 章 “多维尺度变换”. 本书的最后设置了关于 R 和 Python 语言的附录.

此外, 本书在正文中使用 R 语言来实现数据分析的计算并解释其输出的结果, 从第 2 章开始的每一章都提供了例题计算的 Python 代码, 以满足学习或使用 Python 语言读者的需要.

除了第 1 章, 每章后面都提供了习题, 帮助读者掌握各章的内容或计算.

1.5 软件和编程

在现代社会, 一个人是否会编程密切关系到其生存状况, 笔者认为, 编程不限于一个软件的语言, 而是要培养明白编程要点的泛型编程能力.

编程必须在应用中学, 上专门的软件理论课学习编程远远不如针对具体目的来完成一个数据分析任务效果好. 在编程中要不怕犯错误, 不断纠正错误是熟悉软件的最快途径.

R 软件的程序大多是统计学家编写的, 对搞统计分析的人很友好. R 网站上不断更新并持续增加的 1 万多个程序包[3]展示了各种最新的数据分析方法. 绝大多数新方法的计算机实现均首先出现在 R 网站上. 这些程序包的程序都是透明的, 可供审视和改写, 这是任何商业软件无法比拟的, 透明是对抗错误的最好方式. R 软件的优点是容易编写, 帮助系统很优秀, 而且对各种统计方法有比较细致的描述, 输出结果丰富; 缺点是软件本身处理较大数据的速度较慢, 但目前许多快速编程软件都有针对 R 的接口, 而且还有一些针对 R 包装的软件, 使得速度慢不会成为其发展及受欢迎的障碍.

不像 R 软件主要用于数据分析, Python 是通用软件, 可以完成任何计算机能够做的任务. 由于 Python 比 C++ 更易编写, 有限的速度差距又已经被一些包装软件所克服, Python 很快成为 C++ 最强有力的替代. Python 的编程非常灵活, 虽然其帮助系统不如 R 软件, 但由于它是开源软件, 网上帮助很好用. Python 不是专门的统计软件, 其统计方法资源远不如 R 软件多, 而且对于一些模型, 它的输出结果也不如 R 软件那么丰富.

人们通常想知道应该学习什么编程软件, 笔者认为, 学习任何编程软件都有益处, 如果学会第一个编程语言花了一个星期, 学会第二个可能只需三四天的时间. 这里所说的 "学会" 是指能够完成你手中的任务, 而不是指精通. 软件是工具, 我们能够使用它完成任务就行了, 没有必要完全精通. 学习编程的关键是遇到问题时知道去哪里寻求帮助, 网络上的信息很全面, 显然任何教师都无法独自掌握如此海量信息, 在网上寻求帮助是编程者的惯例. 软件在不断发展, 我们必须跟上时代的潮流.

鉴于 R 和 Python 的区别, 以及各种模型的选项和程序包或模块的编写过程及意图的不同, 本书中使用两种软件处理同一个数据的输出结果可能不同. 我们对数据和程序包或模块采用了一些比较随意的选择方式, 希望这不会对可能有 "洁癖" 或 "强迫症" 心理的人造成不适.

1.6 如何教学

人都是在不断学习中成长的, 而且绝大多数能力是自学的. 回想一下, 你有哪一能力是课堂上当场学会的 (不用做作业)? 教师应该引导学生产生兴趣及疑问、展开独立思考并理解世界, 让学生关注世上的最新事物, 还是应该告诉学生世上有什么? 学生难道不会自己查阅资料和自学吗? 离开学校就是知识获取过程的终结吗? 教学是否等于播放陈旧的录音? 不看讲义而能不错一个符号地把书上的内容抄写在黑板上是好老师的标准吗? 教师自己有必要不断学习吗? 学生挑战教师好不好? 没有学生挑战说明什么? 教师应如何面对学生 (在许多方面) 比自己强这样的现实? 不带疑问和怀疑去上课的学生会有收获吗? 课后学生没有疑问的课堂教学正常吗?

[3]在 2009 年底, R 网站上的程序包不到 1000 个.

思考了上面一些问题之后, 关于本书的教学方式也就十分清楚了. 最主要的是弄明白每个模型的直观含义及拟合数据的过程和原理. 至于有关的数学公式, 只需要介绍其目的及意义, 而不必做过度的数学推导, 这些数学公式大都是有百年历史的既定结论. 实际上, 解释这些模型的原理并不比推导数学公式更容易. 本书中标有 * 号的部分可作为参考材料, 教师可以根据需要来确定讲哪些内容. 但是, 部分没有标 * 号的内容也可以用于自学或作为参考材料. **教科书仅仅是教学的参考和工具, 教科书的使用应该服从于获取知识和能力的需要, 教科书不应是限制师生思维的教条和框框.**

我相信, 任何读者只要愿意努力, 就有可能成为一个名副其实的数据科学家.

第 2 章　矩阵代数回顾

本章的内容是非常经典的数学知识, 在几乎所有的高等代数书中都可以找到. 在本章安排的这些内容, 主要是方便读者查阅, 它们只是工具, 不是必须要在课堂上讲授. 由于后续各章多少会涉及本章内容, 所以本章内容没有标星号 (*).

2.1　矩　阵

2.1.1 基本定义

矩阵是数值或数值变量的一个矩形数字阵, m 行 n 列 (亦称 $m \times n$ 维或 $m \times n$ 阶) 的矩阵 $\boldsymbol{A}$ 的形式为:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

$\boldsymbol{A}$ 的表示方法或记号有很多, 比如显示维数的 $\boldsymbol{A}_{(m\times n)} = \{a_{ij}\}$ 或者显示元素的 $\boldsymbol{A} = \{a_{ij}\}$. 此外, 根据不同的需要及方便起见, 元素下标的次序并没有统一规定, 比如, 元素 a_{ij} 的两个下标不一定必须是 i 代表行及 j 代表列, 也可能相反.

前述矩阵 $\boldsymbol{A}$ 的**转置**记为 $\boldsymbol{A}^\top$(这是本书采用的记号) 或 $\boldsymbol{A}'$, 定义为:

$$\boldsymbol{A}^\top = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}.$$

一个 $m \times 1$ 的矩阵称为**列向量**, 而一个 $1 \times n$ 的矩阵称为**行向量**. 比如 $1 \times m$ 矩阵 $\boldsymbol{a} = [a_1, a_2, \ldots, a_m]$ 是一个行向量, 也记为 $\boldsymbol{a} = (a_1, a_2, \ldots, a_m)$, 其元素为 $a_1, a_2, \ldots, a_m$, 而其转置为列向量, 即

$$\boldsymbol{a}^\top = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}.$$

也可以用列向量或行向量表示矩阵, 比如, 前述矩阵 $\boldsymbol{A}$ 的 n 个列向量记为 $\boldsymbol{a}_j = (a_{1j}, a_{2j}, \ldots, a_{mj})^\top$ $(j = 1, 2, \ldots, n)$, 那么矩阵 $\boldsymbol{A}$ 可表示为 $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n]$; 如果

前述矩阵 $\boldsymbol{A}$ 的 m 个行向量记为 $\boldsymbol{a}_i = (a_{i1}, a_{i2}, \ldots, a_{in})^\top$ $(i = 1, 2, \ldots, m)$, 那么, 矩阵 $\boldsymbol{A}$ 可表示为 $\boldsymbol{A}^\top = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m]$.

对于一个方阵 $\boldsymbol{A} = \{a_{ij}\}$, 如果 $a_{ij} = a_{ji}$, $\forall i, j$ [1], 则称该矩阵为**对称矩阵**.

如果一个方阵 $\boldsymbol{U} = \{u_{ij}\}$ 的元素满足 $u_{ij} = 0$, $\forall j < i$, 则称该矩阵为**上三角矩阵**; 如果一个方阵 $\boldsymbol{L} = \{l_{ij}\}$ 的元素满足 $l_{ij} = 0$, $\forall j > i$, 则称该矩阵为**下三角矩阵**.

如果一个方阵 $\boldsymbol{D} = \{d_{ij}\}$ 的元素满足 $d_{ij} = 0$, $\forall j \neq i$, 则称该矩阵为**对角矩阵**, 这时可记 d_{ii} 为 d_i, 以及 $\boldsymbol{D} = \mathrm{diag}(d_1, d_2, \ldots, d_m)$. 对角线元素全部为 1 的对角矩阵 $\boldsymbol{I} = \mathrm{diag}(1, 1, \ldots, 1)$ 称为**单位矩阵**; 如果其维度为 n, 往往记为 $\boldsymbol{I}_n$ 或 $\boldsymbol{I}_{(n\times n)}$. 元素全部是 1 的向量记为 $\boldsymbol{1} = (1, 1, \ldots, 1)^\top$.

2.1.2 基本矩阵运算

两个同样维数的矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 之和定义为一个新矩阵 $\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B}$, 其元素等于 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 相应元素之和, 即 $\boldsymbol{C} = \{c_{ij}\} = \{a_{ij} + b_{ij}\}$. 同样定义两个矩阵之差. 一个数量 c 和一个矩阵 $\boldsymbol{A} = \{a_{ij}\}$ 相乘意味着逐项相乘, 即 $\boldsymbol{B} = c\boldsymbol{A} = \{ca_{ij}\}$. $-\boldsymbol{A}$ 定义为 $\boldsymbol{A}$ 的每个元素乘 -1. 显然有下面的性质:

$$
\begin{aligned}
&\boldsymbol{A} + \boldsymbol{B} = \boldsymbol{B} + \boldsymbol{A}, \quad \boldsymbol{A} + (\boldsymbol{B} + \boldsymbol{C}) = (\boldsymbol{A} + \boldsymbol{B}) + \boldsymbol{C},\\
&\boldsymbol{A} - \boldsymbol{B} = \boldsymbol{A} + (-\boldsymbol{B}) = -(\boldsymbol{B} - \boldsymbol{A}), \quad \boldsymbol{A} - \boldsymbol{A} = \boldsymbol{0},\\
&(\boldsymbol{A} + \boldsymbol{B})^\top = \boldsymbol{A}^\top + \boldsymbol{B}^\top, \quad (b + c)\boldsymbol{A} = b\boldsymbol{A} + c\boldsymbol{A},\\
&\boldsymbol{A} + \boldsymbol{0} = \boldsymbol{A}, \quad c\boldsymbol{A} = \boldsymbol{A}c, \quad -(-\boldsymbol{A}) = \boldsymbol{A},\\
&c(\boldsymbol{A} + \boldsymbol{B}) = c\boldsymbol{A} + c\boldsymbol{B}, \quad 0\boldsymbol{A} = \boldsymbol{0}, \quad 1\boldsymbol{A} = \boldsymbol{A}.
\end{aligned}
$$

如果一个矩阵的列数和另一矩阵的行数相同 (称为可相乘的), 比如 $\boldsymbol{A}_{(m\times n)}$ 和 $\boldsymbol{B}_{(n\times p)}$, 则它们的积定义为一个 $m \times p$ 矩阵 $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B} = \{c_{ij}\}$, 这里

$$
c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}.
$$

两个相同维列向量 $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)^\top$ 和 $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)^\top$ 的内积 (又称点积或点乘)$\boldsymbol{a}^\top\boldsymbol{b}$, 也记为 $\boldsymbol{a} \cdot \boldsymbol{b}$ 或者 $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$, 定义为数量

$$
\boldsymbol{a}^\top \boldsymbol{b} = \sum_{i=1}^{n} a_i b_i.
$$

[1] 符号 "$\forall$" 是 "对于所有"(for all) 的简写.

关于矩阵乘法有下列性质:

$$
\begin{aligned}
&(\boldsymbol{A}+\boldsymbol{B})\boldsymbol{C}=\boldsymbol{A}\boldsymbol{C}+\boldsymbol{B}\boldsymbol{C},\quad \boldsymbol{A}(\boldsymbol{B}+\boldsymbol{C})=\boldsymbol{A}\boldsymbol{B}+\boldsymbol{A}\boldsymbol{C},\\
&\boldsymbol{A}_{(m\times n)}\boldsymbol{I}_n=\boldsymbol{I}_m\boldsymbol{A}_{(m\times n)},\quad \boldsymbol{A}_{(m\times n)}\boldsymbol{0}_{(n\times p)}=\boldsymbol{0}_{(m\times p)},\\
&\boldsymbol{0}_{(q\times m)}\boldsymbol{A}_{(m\times n)}=\boldsymbol{0}_{(q\times n)},\quad \boldsymbol{A}\boldsymbol{B}^{\top}=(\boldsymbol{B}\boldsymbol{A}^{\top})^{\top}.
\end{aligned}
$$

分块矩阵 (见2.1.14) 的运算和矩阵运算类似, 只不过把块 (子矩阵) 当成矩阵元素, 比如, 对分块矩阵

$$
\boldsymbol{A}=\begin{bmatrix}\boldsymbol{M}_{11} & \boldsymbol{M}_{12}\\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22}\end{bmatrix}=\left[\begin{array}{ccc|cc}a_{11} & a_{12} & a_{13} & b_{11} & b_{12}\\ a_{21} & a_{22} & a_{23} & b_{21} & b_{22}\\ \hline c_{11} & c_{12} & c_{13} & d_{11} & d_{12}\\ c_{21} & c_{22} & c_{23} & d_{21} & d_{22}\\ c_{31} & c_{32} & c_{33} & d_{31} & d_{32}\end{array}\right],
$$

有

$$
\boldsymbol{A}^2=\begin{bmatrix}\boldsymbol{M}_{11} & \boldsymbol{M}_{12}\\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22}\end{bmatrix}\begin{bmatrix}\boldsymbol{M}_{11} & \boldsymbol{M}_{12}\\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22}\end{bmatrix}=\begin{bmatrix}\boldsymbol{M}_{11}^2+\boldsymbol{M}_{12}\boldsymbol{M}_{21} & \boldsymbol{M}_{11}\boldsymbol{M}_{12}+\boldsymbol{M}_{12}\boldsymbol{M}_{22}\\ \boldsymbol{M}_{21}\boldsymbol{M}_{11}+\boldsymbol{M}_{22}\boldsymbol{M}_{21} & \boldsymbol{M}_{21}\boldsymbol{M}_{12}+\boldsymbol{M}_{22}^2\end{bmatrix}.
$$

2.1.3 行列式

一个 2×2 矩阵

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

的行列式定义为

$$
|\boldsymbol{A}|=\det(\boldsymbol{A})=\begin{vmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{vmatrix}=a_{11}a_{22}-a_{21}a_{12},
$$

而一个 3×3 矩阵

$$
\boldsymbol{A}=\begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}
$$

的行列式定义为:

$$
|\boldsymbol{A}|=\det(\boldsymbol{A})=a_{11}\begin{vmatrix}a_{22} & a_{23}\\ a_{32} & a_{33}\end{vmatrix}-a_{12}\begin{vmatrix}a_{21} & a_{23}\\ a_{31} & a_{33}\end{vmatrix}+a_{13}\begin{vmatrix}a_{21} & a_{22}\\ a_{31} & a_{32}\end{vmatrix}.
$$

一般的 $n \times n$ 矩阵 $\boldsymbol{A} = \{a_{ij}\}$ 的行列式定义为:

$$|\boldsymbol{A}| = \det(\boldsymbol{A}) = a_{k1}\alpha_{k1} + a_{k2}\alpha_{k2} + \cdots + a_{kn}\alpha_{kn} = \sum_{j=1}^{n} a_{kj}\alpha_{kj}, \ \forall\ k$$

$$= \sum_{i=1}^{n} a_{ik}\alpha_{ik}, \ \forall\ k,$$

这里

$$\alpha_{ij} = (-1)^{i+j}\beta_{ij},$$

而 β_{ij} 为 $\boldsymbol{A}$ 中没有第 i 行及第 j 列元素的 $(n-1) \times (n-1)$ 维子矩阵的行列式. 这里的 α_{ij} 称为 a_{ij} 的**余子式**.

对于 n 阶方阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$, 以及 $c \in \mathbb{R}$, 有下列性质:

$$|c\boldsymbol{A}| = c^n|\boldsymbol{A}|, \quad |\boldsymbol{AB}| = |\boldsymbol{BA}| = |\boldsymbol{A}||\boldsymbol{B}|.$$

2.1.4 矩阵的逆

一个 n 阶方阵 $\boldsymbol{A}$ 的**逆**也是 n 阶方阵, 记为 $\boldsymbol{A}^{-1}$, 满足 $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}_n$. 当然, 矩阵的逆不一定存在, 如果 $\boldsymbol{A}$ 的逆存在, 则称 $\boldsymbol{A}$ 为**非奇异的**, 否则称它为**奇异的**. 方阵 $\boldsymbol{A}$ 存在逆的充分必要条件是其行列式 $|\boldsymbol{A}| \neq 0$.

关于矩阵的逆, 有下列性质:

$$\begin{aligned}
&\boldsymbol{I}^{-1} = \boldsymbol{I}, \ \ (\boldsymbol{A}^{-1})^{-1} = \boldsymbol{A}, \ (\boldsymbol{A}^{\top})^{-1} = (\boldsymbol{A}^{-1})^{\top} \\
&(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}, \ \ (c\boldsymbol{A})^{-1} = c^{-1}\boldsymbol{A}^{-1} \\
&\mathrm{diag}^{-1}(d_1, d_2, \ldots, d_n) = \mathrm{diag}(1/d_1, 1/d_2, \ldots, 1/d_n),
\end{aligned}$$

这里, $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 为非奇异方阵, c 为数量.

2.1.5 矩阵的广义逆

如果 $\boldsymbol{A}$ 为 $n \times m$ 矩阵, $\boldsymbol{y} \in \mathbb{R}^n$ 及 $\boldsymbol{x} \in \mathbb{R}^m$, 考虑方程组

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}, \tag{2.1.1}$$

则在 $m = n$ 时, 如果 $\boldsymbol{A}$ 的逆存在, 则方程组 (2.1.1) 有解 $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y}$; 如果 $\boldsymbol{A}$ 不可逆或者 $m \neq n$, 则无解或者解不唯一. 这时可以定义 $\boldsymbol{A}$ 的广义逆.

矩阵 $\boldsymbol{A}$ 的**广义逆**$\boldsymbol{A}^{-}$ 满足下列条件:

$$\boldsymbol{A}\boldsymbol{A}^{-}\boldsymbol{A} = \boldsymbol{A}. \tag{2.1.2}$$

容易验证, 如果方阵 $\boldsymbol{A}$ 的逆 $\boldsymbol{A}^{-1}$ 存在, 则 $\boldsymbol{A}^{-1} = \boldsymbol{A}^{-}$. 这种情况之外的广义逆并不唯一.