高等院校研究生用书
TEXTBOOKS FOR GRADUATES

复杂数据统计方法

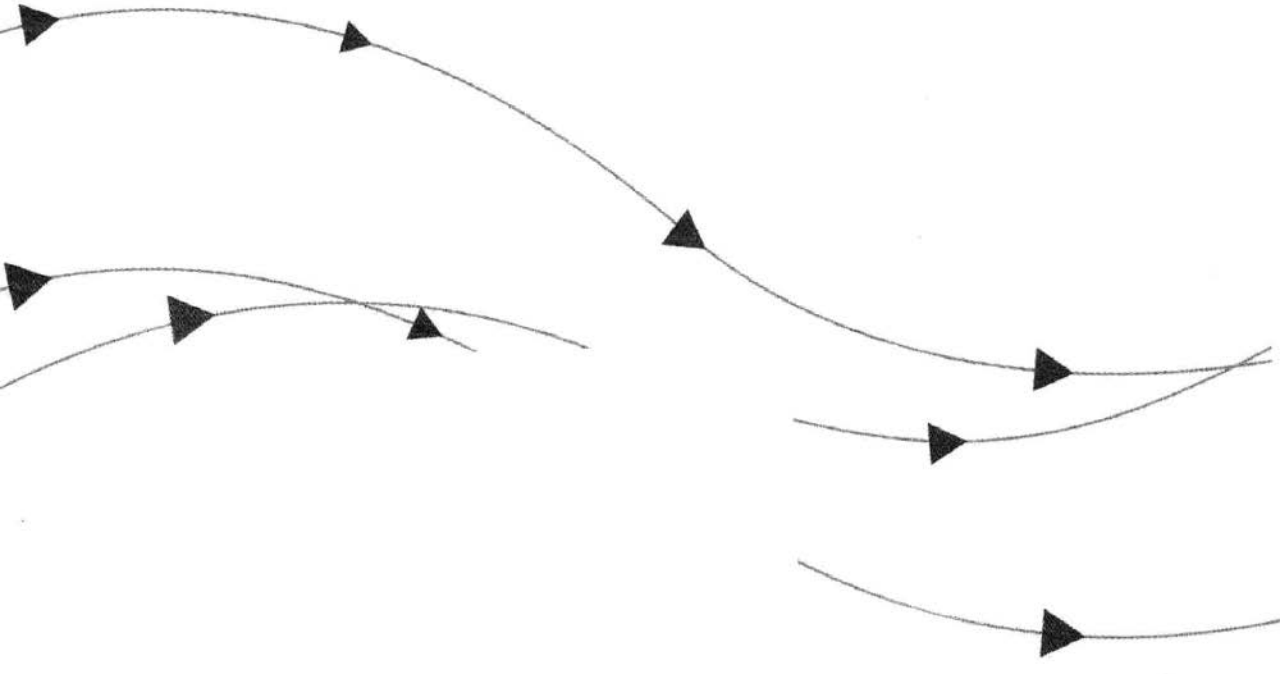
——基于R的应用

（第三版）

吴喜之 编著

中国人民大学出版社

高等院校研究生用书
TEXTBOOKS FOR GRADUATES



复杂数据统计方法

——基于R的应用（第三版）

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

复杂数据统计方法: 基于R的应用/ 吴喜之编著. —3版. —北京: 中国人民大学出版社, 2015.7

高等院校研究生用书

ISBN 978-7-300-21594-5

I. ①复··· II. ①吴··· III. ①统计分析 – 应用软件 – 研究生 – 教材 IV. ①C819

中国版本图书馆CIP数据核字(2015) 第152677号

高等院校研究生用书

复杂数据统计方法(第三版)

—— 基于R的应用

吴喜之 编著

Fuza Shuju Tongji Fangfa

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)		010-62511770(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	北京东方圣雅印刷有限公司	**版　　次**	2012年10月第1版
规　　格	170mm× 228mm　16开本		2015年7月第3版
印　　张	20.5 插页1	**印　　次**	2015年7月第1次印刷
字　　数	385 000	**定　　价**	36.00 元

版权所有　侵权必究　印装差错　负责调换

前 言

什么是复杂数据? 没有人能够确切定义. 本书将通常统计基本教科书中的例子所代表的数据称为简单数据, 例如普通最小二乘线性回归所能够圆满处理的独立同正态分布数据、用标准多元分析方法能够处理的具有多元正态分布的数据等. 其他本科教科书中能够相对圆满处理的数据应该不算复杂数据. 显然, 现实世界中遇到的绝大多数数据都不是标准教科书中所介绍的方法能够圆满处理的, 因此都应该被认为是复杂数据. 按照这个定义, 绝大多数真实数据都是复杂数据.

对于一个实际工作者来说, 拿到一个真实数据以后, 很可能需要查阅不少文献来寻找适合这个数据的几种可能模型(假定知道用什么模型可能解决问题), 再翻阅若干种软件手册来查阅这些文献所使用软件的计算方法(假定购买了这些软件). 造成这种情况的原因是, 多数统计教科书是以模型或方法为导向的, 内容也多是按照数学思维展开的.

以模型或方法为导向的教科书通常以介绍某种数学模型和方法为主, 同时说明这种模型适用于满足某些数学假定的数据, 最后说明该模型对于这些满足假定的数据拟合的优越性. 实际上, 任何一种真实数据是否满足某种数学假定几乎无法证明, 每一类数据都可能有不止一种现成的统计方法来处理, 还有无数的未知方法等待人们去开发. 以模型或方法为主导的方式往往让读者忽略了其他有关的方法, 而那些被忽略的方法在某种意义上很可能更有效, 或者更优越.

笔者认为, 现在需要一本具有以下特点的书:

- 用实际数据做案例.
  - 介绍的数据种类尽可能广泛;
  - 这些数据必须是真实的;
  - 这些数据必须不是简单平凡的教科书例子;
  - 每个数据都有理论及应用方面的背景;
  - 所有数据都能从网上下载.

- 对每种数据都介绍可能的方法.
  - 这些方法尽可能新;
  - 对各种方法进行比较;
  - 所有方法必须有计算支持.
- 全书使用一种软件.
  - 该软件必须是免费的, 可以从网上下载的;
  - 该软件必须能够包含尽可能多的最新统计方法;
  - 该软件必须不断更新;
  - 书中所有结论都可以通过运行该软件程序而得出, 并给出所有代码.
- 篇幅不能太大.
- 必须由浅入深, 对经典知识和模型进行必要的回顾.
- 不能有太多数学公式, 但至少必须能让读者直观理解各种方法的含义.
- 其宗旨是训练动手的能力, 而不是面面俱到地告诉人们所有细节.
- 不仅提供各种方法, 而且提醒人们使用各种方法存在的风险.

本书以数据形式为导向, 对应不同的数据形式介绍可能使用的一些方法. 首先引入某些感兴趣类型的数据, 再介绍并且对比可能适合这些数据的一些统计方法. 这些统计方法可能属于许多不同的模型, 属于不同的统计方向, 但只要适用于同一类数据, 我们就尽量将它们都予以介绍. 笔者觉得这种以数据为主导的学习方式有助于理解统计作为数据科学的本质, 有助于实际工作者通过数据学习多种统计方法的应用. 我们列举了可能用于同类数据的若干方法, 希望对创造新的数据分析方法有所启发并促使进一步探索, 同时也让读者免受查阅大量不同文献之苦. 本书不可能介绍所有的方法, 大量新方法在你阅读本书的时候正在诞生.

本书所有的分析都通过免费的自由软件R来实现. 读者可以毫不费力地重复本书所有的计算. R网站[1]拥有世界各地统计学家贡献的大量最新程序包(package), 这些程序包以飞快的速度增加和更新, 已从2009年底的大约1000个增加到2015年7月8日的6814个, 仅此之前的30天就增加了720个. 它们代表了统计学家创造的崭新的统计方法. 这些程序包的代码都是公开的.[2]

[1]网址: http://www.r-project.org/.

[2]除了极个别并非秘密的子程序之外, 因为它们很费时间, 用机器代码实行.

与此相对比, 所有商业软件远没有如此多的资源, 也不会更新得如此之快, 而且商业软件的代码都是保密的昂贵"黑匣子". 在发达国家, 不能想象一个统计研究生不会使用R软件. 那里很多学校都开设了R软件的课程. 今天, 任何一个统计学家想要介绍和推广其创造的统计方法, 都必须提供相应的计算程序, 而发表该程序的最佳地点就是R网站. 由于方法和代码是公开的, 这些方法很容易引起有关学者的关注, 这些关注对研究相应方法形成群体效应, 推动其发展. 不会编程的统计学家在今天是很难生存的.

在学校讲授任何一款商业软件都是为该公司做义务广告, 如果没有相关软件公司的资助, 就没有学校愿意花钱讲授商业软件. 在教学中使用盗版软件是违法行为, 绝对不应该或明或暗地鼓励师生使用盗版商业软件.

对R软件编程的熟悉还有助于学习其他快速计算的语言, 比如C++, FORTRAN和Python等, 这对于应对因快速处理庞大的数据集而面临的巨大的计算量有所裨益.

本书首先通过一些简单的统计和数学内容介绍R软件的基本知识, 然后介绍数据分析的一些基本逻辑和常识. 本书的主体则是根据不同数据形式介绍相应的方法. 本书以数据为主导, 各章都是完全独立的. 有一些统计基本知识的读者可以选读本书的任何一个完整的部分. 虽然本书介绍的方法涉及应用统计的各个方面, 但不可能介绍所有的数学和统计细节, 否则将会是一部巨型的百科全书. 笔者尽量用文字和少量数学公式对各种方法的原理予以直观介绍, 并引导读者做进一步的阅读.

由于本书没有按照数学模型的分类来编排, 因此对各种方法的介绍不可能满足数学上的系统、整洁和完美的要求, 但这正是对现实数据和现实世界的反映. 如果现实数据都像标准教科书例子那样"规范", 统计就没有存在和发展的必要了. 本书试图让读者理解世界是复杂的, 数据形式是多种多样的. 必须有超越书本、超越所谓权威的智慧和勇气, 才能充满自信地面对世界上出现的各种挑战.

由于统计正以前所未有的速度发展, R网站及其各个程序包也在不断更新, 因此, 笔者希望读者通过对本书的学习, 学会如何通过R不断学习新的知识和方法. "授人以鱼, 不如授之以渔", 成功的教师不是像百科全书那样告诉学生一些现成的知识, 而是让学生产生疑问和兴趣, 以促进其做进一步的探索.

本书绝大多数数据例子都可以从网上找到并且下载. 这些例子背后都有一些理论和应用的故事. 笔者并没有刻意挑选例子所在的领域, 这没有关系. 你学会了一加一等于二, 也就学会了一个苹果加一个苹果等于两个苹果, 或一个梨加一个梨等于两个梨这样的计算. 那个把作为科学的统计按照工种来划分(诸如工业统计、农业统计、劳动统计)的时代早已一去不复返了. 统计是为各个领域服务的, 我们想要得到的是到任何领域都能施展的能力, 而不是有限的行业培训. 如果你能够处理具有挑战性的数据, 那么无论该数据来自何领域, 你的感觉都会很好.

虽然本书冠以"复杂数据统计方法"之名, 但对"非复杂"数据的方法都有较完整的回顾, 并给出了相应的运算程序, 只不过没有像标准教科书那样详细地解释细节而已.

本书的适用范围很广, 其内容曾经在中国人民大学、首都经贸大学、中央财经大学、西南财经大学、云南财经大学、四川大学、哈尔滨理工大学、新疆财经大学、中山大学、内蒙古科技大学、云南师范大学讲授过，对象包括数学、应用数学、统计、精算、经济、旅游、环境等专业的本科生以及数学、应用数学、统计、计量经济学、生物医学、应用统计、经济学等专业的硕士和博士研究生. 作为成绩评定, 给每个学生分配若干网站上的实际数据, 并要求他们在学期末将他们分析处理这些数据的结果形成报告. 这些数据如何处理, 没有标准答案, 甚至有些必要的方法还超出了授课的范围, 需要学生做进一步的探索和学习. 笔者认为, 应用统计硕士所学的内容应该包括本书的大部分内容. 希望本书对于各个领域的教师以及实际工作者都有参考价值.

本书面世以来, 得到了广大读者的支持和鼓励. 目前的第三版对第二版做了相当大的增补, 并且重新安排了章节. 除了对前两版中发现的错误做出改正之外, 主要增加了贝叶斯网络、定序变量的比例优势模型、调查问卷的垃圾比例计算等内容. 总共增加将近1/3的篇幅. 和第二版一样, 第三版的排版是笔者通过LaTeX软件实现的.

在任何国家及任何制度下都能够生存和发展的知识和能力, 就是科学, 是人们在生命的历程中应该获得的.

吴喜之

教师教学服务说明

中国人民大学出版社工商管理分社以出版经典、高品质的工商管理、财务会计、统计、市场营销、人力资源管理、运营管理、物流管理、旅游管理等领域的各层次教材为宗旨。

为了更好地为一线教师服务，近年来工商管理分社着力建设了一批数字化、立体化的网络教学资源。教师可以通过以下方式获得免费下载教学资源的权限：

在“人大经管图书在线”（www.rdjg.com.cn）注册,下载“教师服务登记表”，或直接填写下面的“教师服务登记表”，加盖院系公章，然后邮寄或传真给我们。我们收到表格后将在一个工作日内为您开通相关资源的下载权限。

如您需要帮助，请随时与我们联络：

中国人民大学出版社工商管理分社
联系电话：010-62515735，62515749，62515987
传真：010-62515732，62514775　　电子邮箱：rdcbsjg@crup.com.cn
通讯地址：北京市海淀区中关村大街甲59号文化大厦1501室（100872）

教师服务登记表

姓名		□先生 □女士	职称		
座机/手机			电子邮箱		
通讯地址			邮编		
任教学校			所在院系		
	课程名称	现用教材名称	出版社	对象(本科生/研究生/MBA/其他)	学生人数
所授课程					
需要哪本教材的配套资源					
人大经管图书在线用户名					

院/系领导（签字）：

院/系办公室盖章

目 录

前 言 i

第一章 引 言 1

1.1 作为科学的统计 1

1.2 数据分析的实践 3

1.3 数据的形式以及可能用到的模型 4

1.3.1 横截面数据: 因变量为实轴上的数量变量 5

1.3.2 横截面数据: 因变量为分类变量、频数或定序变量 5

1.3.3 纵向数据、多水平数据、面板数据、重复观测数据 5

1.3.4 多元数据各变量之间的关系: 多元分析 6

1.3.5 抽样调查数据中垃圾比例的计算 6

1.3.6 路径模型/结构方程模型 6

1.3.7 贝叶斯网络 7

1.3.8 多元时间序列数据 7

1.4 R软件入门 7

1.4.1 简介 7

1.4.2 安装和运行小贴士 8

1.4.3 动手 9

1.5 国内统计教学(课本)的若干误区 10

1.5.1 假设检验的误区: 不能拒绝就接受 10

1.5.2 假设检验的误区: p值小于0.05就显著 11

1.5.3 置信区间的误区 12

1.5.4 最小二乘线性回归中的误区 12

1.5.5 样本量是多少才算大样本 15

1.5.6 用31个省、直辖市、自治区数据能做什么 16
1.5.7 汇总数据(比如部分均值)和原始观测值的区别 16

第二章 横截面数据回归: 经典方法 18

2.1 简单回归回顾 18
2.1.1 对例2.1数据的简单拟合 19
2.1.2 对例2.1数据的进一步分析 20
2.1.3 对简单线性回归的一些讨论 23
2.1.4 损失函数及分位数回归简介 24
2.2 简单线性模型中的指数变换 27
2.3 生存分析数据的Cox回归模型 31
2.4 数据出现多重共线性情况: 岭回归、lasso回归、适应性lasso回归、偏最小二乘回归 34
2.4.1 岭回归 35
2.4.2 lasso回归 36
2.4.3 适应性lasso回归 38
2.4.4 偏最小二乘回归 39

第三章 横截面数据回归: 机器学习方法 41

3.1 没有任何先验假定的数据: 机器学习回归方法 41
3.2 决策树回归(回归树) 43
3.2.1 拟合全部数据 43
3.2.2 交叉验证 45
3.3 boosting回归 46
3.4 bagging回归 47
3.5 随机森林回归 48
3.6 支持向量机回归 49
3.6.1 SVM 49

3.6.2 SVR 49
3.6.3 交叉验证 50
3.7 人工神经网络回归 51
3.7.1 确定合适的隐藏层节点数目 52
3.7.2 交叉验证 53
3.8 10折交叉验证结果汇总及方法稳定性讨论 54

第四章 横截面数据分类: 经典方法 57

4.1 logistic回归和probit回归 57
4.1.1 广义线性模型简单回顾 57
4.1.2 脊柱数据例子 58
4.1.3 logistic回归 59
4.1.4 probit回归 63
4.2 经典判别分析 64
4.2.1 关于例4.1 column.2C.csv数据的两分类判别 65
4.2.2 例4.1 column.2C.csv数据两分类判别诸方法的10折交叉验证结果 66
4.2.3 关于例4.1 column.3C.csv数据的三分类判别 68
4.2.4 关于例4.1 column.3C.csv数据的三分类判别的三种方法的10折交叉验证结果 70

第五章 横截面数据分类: 机器学习方法 72

5.1 对变量没有任何限制: 机器学习分类方法 72
5.1.1 概论和例子 72
5.1.2 产生交叉验证数据集 72
5.2 决策树分类(分类树) 74
5.2.1 拟合全部数据 74
5.2.2 交叉验证 76

5.3 adaboost分类 76
5.3.1 拟合全部数据 77
5.3.2 交叉验证 79
5.4 bagging分类 80
5.4.1 拟合全部数据 80
5.4.2 交叉验证 81
5.5 随机森林分类 82
5.5.1 拟合全部数据 82
5.5.2 交叉验证 85
5.6 支持向量机分类 85
5.6.1 拟合全部数据 85
5.6.2 交叉验证 86
5.7 最近邻方法分类 87
5.8 神经网络分类 88
5.8.1 拟合 88
5.8.2 神经网络对于不同参数的拟合效果分析 89
5.9 分类方法10折交叉验证结果汇总 90

第六章 横截面数据: 计数或有序因变量 92

6.1 概要和例子 92
6.2 经典的Poisson 对数线性模型回顾 93
6.3 使用Poisson对数线性模型时的散布问题 95
6.4 零膨胀时的Poisson回归 98
6.5 用机器学习的算法模型拟合计数因变量数据 101
6.5.1 随机森林拟合例6.1数据的10折交叉验证 101
6.5.2 决策树(回归树)拟合例6.1数据的10折交叉验证 101
6.5.3 支持向量机拟合例6.1数据的10折交叉验证 102

6.5.4 各种方法拟合例6.1数据的10折交叉验证结果比较和一些讨论 102
6.6 多项logit模型及多项分布对数线性模型回顾 104
6.6.1 多项logit模型回顾 104
6.6.2 多项分布对数线性模型回顾 107
6.7 有序变量的比例优势模型 110
第七章 纵向数据(多水平模型、面板数据) 119
7.1 线性随机效应混合模型 121
7.2 广义线性随机效应混合模型 125
7.3 决策树及随机效应模型 127
7.4 纵向生存数据分析 130
7.4.1 Cox 随机效应混合模型 130
7.4.2 分步联合建模 134
7.5 计量经济学家的视角: 面板数据 141
第八章 多元分析 147
8.1 实数轴上的数据: 经典多元分析内容回顾 147
8.1.1 主成分分析及因子分析 147
8.1.2 分层聚类及k均值聚类 156
8.1.3 典型相关分析 164
8.1.4 对应分析 168
8.2 非经典多元数据分析: 可视化 171
8.2.1 主成分分析 173
8.2.2 对应分析 174
8.2.3 多重对应分析 175
8.2.4 多重因子分析 176
8.2.5 分层多重因子分析 178

8.2.6 基于主成分分析的分层聚类 179

第九章 多元数据的关联规则分析 181

9.1 问题的提出 181

9.2 数据的初步处理 182

9.3 一些基本概念 183

9.4 频数和规则 185

第十章 调查数据中垃圾比例的计算 188

10.1 动机 188

10.2 我们的着眼点在于数据本身 189

10.3 解剖一个数据 190

10.4 计算例10.1Income2000.csv数据中各种比例及置信区间 193

10.4.1 一些计算准备工作 193

10.4.2 主要函数 194

10.4.3 更多的计算 196

10.5 通过一个简单人造数值例子解释本章代码的意义 201

第十一章 路径建模(结构方程建模)数据的PLS分析 207

11.1 路径模型概述 207

11.1.1 路径模型 207

11.1.2 路径模型的两种主要方法 208

11.2 PLS方法: 顾客满意度的例子 210

11.3 协方差方法简介 215

11.4 结构方程模型的一些问题 222

第十二章 贝叶斯网络 224

12.1 引言 224

12.2 贝叶斯网络涉及的一些基本概念 227

12.2.1 贝叶斯网络 227

12.2.2 贝叶斯网络中的条件独立性概念 227

12.3 通过贝叶斯网络做推断 229

12.4 学习贝叶斯网络 231

12.4.1 网络学习算法的种类 231

12.4.2 几种可能面对的问题 232

12.5 贝叶斯网络的数值例子及计算 236

12.5.1 全部变量是离散变量的情况 236

12.5.2 全部变量是连续变量的情况 240

12.5.3 连续变量和离散变量混合的情况 241

第十三章 多元时间序列数据 245

13.1 时间序列的基本概念及单变量时间序列方法的回顾 245

13.1.1 时间序列的一些定义和基本概念 245

13.1.2 常用的一元时间序列方法 252

13.2 单位根检验、协整检验及Granger因果检验 262

13.2.1 概述 262

13.2.2 单位根检验 264

13.2.3 协整检验 266

13.2.4 Granger 因果检验 272

13.3 VAR模型、VARX模型与状态空间模型 273

13.3.1 VAR模型的拟合与预测 273

13.3.2 VARX模型的拟合与预测 276

13.3.3 状态空间模型的拟合与预测 278

13.3.4 模型的比较 280
13.4 非线性时间序列 283
13.4.1 引言 283
13.4.2 线性AR模型 285
13.4.3 自门限自回归模型 286
13.4.4 Logistic 平滑过渡自回归模型 290
13.4.5 神经网络模型 293
13.4.6 可加AR模型 294
13.4.7 模型的比较 294
13.4.8 门限协整 295
附录 练习: 熟练使用R软件 300
参考文献 309

第一章　引 言

1.1　作为科学的统计

统计是科学(science), 而科学的基本特征是其方法论: **对世界的认识源于观测或实验的信息(或者数据), 总结信息时会形成模型(亦称假说或理论), 模型会指导进一步的探索, 直至遇到这些模型无法解释的现象, 这就导致对这些模型的更新和替代**. 这就是科学的方法. 只有用科学的方法进行的探索才能称为科学.

科学的理论完全依赖于实际, 统计方法则完全依赖于来自实际的数据. 统计可以定义为"收集、分析、展示和解释数据的科学", 或者称为**数据科学**(data science). 统计几乎应用于所有领域. 人们现在已经逐渐认识到, 作为数据科学的统计, 必须和实际应用领域结合, 必须和计算机科学结合, 才会有前途(见图1.1).

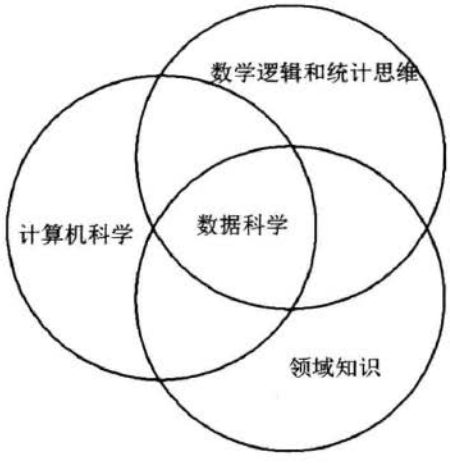


图 1.1　作为数学逻辑和统计批判性思维、计算机科学、实际领域知识之交的数据科学

统计的思维方式是归纳(induction), 也就是从数据所反映的现实得到比较一般的模型, 希望以此解释数据所代表的那部分世界. 归纳和以演绎(deduction)为主的数学思维方式相反, 演绎是在一些人为的**假定**(比如一个公理系统)之下推导出各种结论.

在统计科学发展的前期, 由于没有计算机, 不可能应付庞大的数据量[1], 只

[1]请想象一下用纸和笔得到简单线性回归所必须计算的预测矩阵$\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top$, 假定$\boldsymbol{X}$为30 × 5的数值矩阵.

能在对少量数据的背景分布做出诸如独立同正态分布之类的数学假定后, 建立一些假定的数学模型, 进行手工计算, 并推导出一些由这些模型所得结果的性质, 诸如置信区间、假设检验的p值、无偏性及相合性等. 在数据与数学假定相差较远的情况下, 人们又利用诸如中心极限定理或大样本定理得到当样本量趋于无穷时的一些类似的性质. 统计的这种发展方式, 给统计打上了很深的数学烙印.

统计发展的历史痕迹体现在很多方面, 特别是流行"模型驱动"的研究及教学模式. 各统计院系的课程大多以数学模型作为课程的名称和主要内容, 一些数理统计杂志也喜欢发表没有数据背景的关于数学模型的文章. 很多学生毕业后只会推导一些课本上的公式, 却不会处理真实数据. 一些人对于有穷样本, 也假装认为是大样本, 并且堂而皇之地用大样本性质来描述从有穷样本中得到的结论. 至于数据是否满足大样本定理的条件, 数据样本是不是"大样本"等关键问题尽量不谈或少谈. 按照模型驱动的研究方式, 一些学者不从数据出发, 而是想象出一些他们感觉很好的数学模型, 然后在世界上到处寻求"适合"他们模型的数据来"证明"自己的模型的确有意义. 这种自欺欺人的做法绝对是不科学的.

以模型而不是数据为主导的研究方式导致统计在某种程度上成为自我封闭、自我欣赏及自我评价的系统. 固步自封的后果是, 30多年来, 统计丢掉了许多属于数据科学的领域, 也失去了许多人才. 在现成数学模型无法处理大量复杂数据的情况下, 计算机领域的研究人员和部分概率论及统计学家开发了许多计算方法, 处理了传统统计无法解决的大量问题. 诸如人工神经网络、决策树、boosting、随机森林、支持向量机等大量算法模型的相继出现宣告了传统数学模型主导(如果不是垄断的话)数据分析时代的终结. 这些研究最初根本无法刊登在传统统计杂志上, 因此大多出现在计算机及各应用领域的杂志上.

模型驱动的研究方法在前计算机时代有其合理性. 但是在计算机快速发展的今天, 仍然固守这种研究模式, 就不会有前途了. 人们在处理数据时, 首先寻求现有的方法, 当现有方法不能满足他们的需求时, 往往会根据数据的特征创造出新的可以计算的方法来满足实际需要. 这就是统计科学近年来飞速发展的历程. 创建模型的目的是适应现实数据. 统计研究应该是由问题或者数据驱动的, 而不是由模型驱动的.

随着时代的进步, 各个统计院系现在也开始设置诸如数据挖掘、机器学习等课程, 统计杂志也开始逐渐重视这些研究. 这些算法模型大多不是用封闭的数学公式来描述, 而是体现在计算机算法或程序上. 对于结果的风险也不是用假定的分布(或渐近分布)所得到的p值来描述, 而是用没有参加建模训练的测试集的误差来描述. 这些方法发展很快, 不仅因为它们能够解决问题, 还因

为那些不懂统计或概率论的人也能够完全理解结果(这也是某些有"知识垄断欲"的传统统计学家不易接受的现实). 现在, 无论承认与否, 多数统计学家都明白, 如果不会计算机编程或者不与编程人员合作, 则不会产生任何有意义的成果.

1.2 数据分析的实践

数据收集 首先要根据实际目的收集数据. 有些数据是需要人工收集的, 比如通过普查、调查、实验、观察等手段得到; 另一些则来自各种与计算机联系的数据源, 比如遥感数据、网络数据、商务数据、远程上报数据等. 确定哪些变量的数据需要收集是非常关键的, 这个决策不是基于数学或统计的知识, 而是基于对相应领域的了解和经验. 有数据不一定能够得到需要的结论, 我们需要的是与所关心问题**充分有关**的变量的数据.

数据预处理 原始数据往往或多或少地存在各种缺失值, 还有不合逻辑或不一致等问题. 这需要进行预处理. 这些工作很可能非常费时而且极其琐碎, 但必须去做, 否则后续的分析是不可能展开的. 填补缺失值有很多方法, 比较简单的就是删除, 或者用同一变量其他值的均值或中位数填补, 或者在各个变量之间建立模型(比如线性模型、最近邻方法等)来填补.

寻找适合的模型 有了数据, 我们需要的是模型, 其目的或者是预测, 或者是理解产生数据的机制. 为了寻找模型, 首先要对数据进行探索性分析, 利用图形、各种统计量或者比较复杂的探索方法来查看数据的关联性、线性性、异方差性、多重共线性、聚类特征、平衡特征、分布形状等. 有了对数据的粗略认识之后, 就要寻找适合的模型, 无论是传统意义上的模型还是以算法为基础的模型. 首先寻找现成的模型, 比较各种模型的计算结果, 如果现有模型不能满足需要, 新的数据分析方法就应该产生了. 模型选择的过程贯穿整个数据分析过程.

比较模型的标准 在传统统计中, 通常要对分布和模型形式做出假定, 在这些假定下确定损失函数, 并依此得到各种判别准则, 这些准则包括各种检验、一些统计量的临界值等. 但绝对不要忘记, 所有这些都是在对数据分布及描述变量之间关系的模型所做的假定[1]之下得到的. 如果这些假定不满足, 则这些准则也没有什么意义了. 在使用算法模型时, 由于没有传统模型的那些假定, 判断模型的好坏通常都用交叉验证(cross validation)的方法, 也就是说, 拿一部分数据作为训练集(training set), 得到模型, 再用另一部分数据(称为测试集(testing set))来看误差是多少. 有时需要进行k折交叉验证(k-fold cross

[1]这些假定都是无法用确定性方法验证的, 最多只能尝试用显著性检验来拒绝它, 如果没有充分理由拒绝, 也不能"证明"这些假定是对的, 它们仍然属于**假定**.

validation), 即把数据分成k份, 每次拿$k-1$份作为训练集, 用剩下的一份作为测试集, 重复k次, 得到k个误差作出平均, 以避免仅用一个测试集可能出现的偏差. 显然, 交叉验证的方法也适用于传统模型之间以及传统模型和算法模型之间的比较.

对结果的解释 最终目的不是选择模型, 而是解释模型所产生的结果, 结果则必须是应用领域的结果, 必须有实际意义. 仅仅用统计术语说某个模型较好, 某个变量显著之类的话是不够的.

1.3 数据的形式以及可能用到的模型

数据形式多种多样, 但大部分可以转换成一个或一组二维数据文件, 比如, n维列联表在R中是一个n维数组, 但总可以转换成一个二维数据阵. 其行数为n个变量各个水平的全部组合数, 如果第i个变量有L_i个水平, 则行数为$\prod_{i=1}^{n} L_i$. 其列数为$n+1$: n个变量加上一个频数变量. 当维数较大时, n维列联表数据很可能有许多水平组合为空值(稀疏). 一些数据, 比如某些空间数据, 则可能需要用多个数据文件来描述, 而且数据格式也不一定是方阵. 本书所用的绝大多数数据都可以从网上下载, 对于单一文件的方阵型数据, 通常每一行代表一个观测值, 每一列代表一个变量.

根据研究目的, 数据中的一些变量可以被人为地作为因变量或自变量, 这样的情况往往出现在人们想要用一部分变量描述或预测另一部分变量的时候, 例如统计中的回归或分类(判别)问题. 在一些情况下, 人们不太关心预测, 但想知道数据变量之间的其他关系, 比如在多元分析课程中的聚类及因子分析等内容. 还有一些情况, 人们想用较长的历史值预测未来, 诸如时间序列数据的分析. 一些数据被称为横截面数据, 那里每个对象仅观测一次. 而有些时候一个对象观测若干次, 但又不同于时间序列的长期观测, 被称为纵向数据. 纵向数据的一些特例在某些领域中被称为面板数据. 在诸如医学、质量控制、心理学、社会学等领域, 人们关心某事件(如死亡、失效等)是否发生及什么时候发生的问题, 相应的数据则被称为生存数据. 调查问卷的结果往往做成列联表的形式, 也被称为列联表数据. 这些数据类型绝对不是排他的, 比如, 有些数据既是生存数据, 又是纵向数据.

对每一种数据类型, 人们根据目的以及数据的一些特殊性, 找到各种数学模型来处理. 下面就本书后面章节可能涉及的各种数据分析的内容加以简单介绍. 由于实际数据是复杂的, 可能必须经过多次尝试及对比才能确定什么模型不适用, 什么模型适用. 所有的传统方法都有一定的假定, 必须注意这些假定的合理性.

1.3.1 横截面数据: 因变量为实轴上的数量变量

当因变量为数量变量时, 人们首先想到的是回归, 教科书中最先介绍的是假定模型中误差项独立同正态分布的线性回归. 由于误差项很可能不满足这个假定, 还有可能有多重共线性等问题, 这样就产生了诸如加权回归、稳健回归、偏最小二乘回归、lasso回归、岭回归、主成分回归、Box-Cox变换(或其他变换)、多项式回归、分位数回归等模型, 还产生了相应的各种检验及判断方法, 诸如最优子集、逐步回归、回归诊断, 等等. 在线性或其他假定不满足时, 又出现非线性回归、非参数回归、广义线性模型、随机效应混合模型、可加模型、广义可加模型, 等等. 生存分析也包含了回归的内容. 近年来, 神经网络、决策树的回归树、boosting、bagging、最近邻方法、随机森林、支持向量机等算法模型也广泛应用于回归. 第二章将会介绍一些上面所述的经典方法部分, 而第三章将介绍机器学习回归方面的内容.

1.3.2 横截面数据: 因变量为分类变量、频数或定序变量

当因变量为分类变量时, 一般的回归方法就不能使用了. 在传统的统计领域, 如果因变量是二分变量, 则可以尝试用广义线性模型中的logistic回归和probit回归, 如果自变量是数量型的变量, 则可以尝试线性判别分析, 这里有可能要求自变量满足正态分布. 但最新的一些基于算法的模型则没有关于分布的假定, 比如决策树的分类树、神经网络、boosting、bagging、随机森林、最近邻方法、支持向量机等, 但能够更好地处理分类问题.

在许多情况下, 数据的形式是频数, 比如列联表数据, 这不是常规意义上的连续型数值变量. 如果把这些频数作为关注的因变量, 而把那些形成列联表的分类变量以及可能得到的数量变量作为自变量, 则可以应用多项分布对数线性模型、多项logit模型及Poisson对数线性模型(作为广义线性模型的特例). 在使用Poisson模型时, 往往可能出现Poisson模型假定不允许的过散布(overdispersion, 即方差大于均值)或欠散布(underdispersion, 即方差小于均值)情况. 这时就要进行矫正或者利用一些双参数模型. 当然, "非正统"的机器学习方法在这里也可以使用.

第四章将介绍分类和判别的经典方法, 而第五章将介绍机器学习的分类方法. 第六章将介绍因变量为频数的各种方法, 包括处理定序变量的比例优势模型.

1.3.3 纵向数据、多水平数据、面板数据、重复观测数据

当一个对象有多次重复观测时, 得到的就不是横截面数据, 而是纵向数据, 经济金融领域的这一类数据有时被称为面板数据, 属于纵向数据的特例. 应对

这种数据有许多不同方法, 这些方法也成为相应处理纵向数据的课程名称, 比如多层模型、随机效应混合模型等. 此外, 很多纵向数据还与生存数据有关, 很多广义线性模型可处理的数据也是纵向数据, 这就产生了更多的处理方法. 一些机器学习方法也被用到纵向数据的分析之中. 第七章将介绍线性随机效应混合模型、广义线性随机效应混合模型、决策树模型及纵向生存数据分析等内容.

1.3.4 多元数据各变量之间的关系: 多元分析

在没有预测任务, 对数据进行分析的目的仅在于确定各个变量之间的关系时, 就不需要确定自变量及因变量了. 这时, 如果变量全部都是服从正态分布的数量变量, 这个问题属于传统的多元分析的范畴, 主要方法包括主成分分析、因子分析、聚类分析、典型相关分析等. 有点另类的对应分析是一种图描述方法, 并不被认为是经典方法, 但也出现于一些教科书中. 对于复杂的分层数据, 或者包含分类(定性)变量的数据, 传统的多元分析方法就不能使用了, 新的基于算法的模型可以应对非常复杂的多元分析课题. 第八章首先回顾经典的多元分析方法, 然后介绍可视化很强的新发展的多元分析方法, 最后在第九章介绍没有丝毫经典统计味道的纯粹机器学习方法: 关联规则分析.

1.3.5 抽样调查数据中垃圾比例的计算

任何抽样调查都会产生大量的比例, 但很少有人计算过这些比例中有多少是根本没有意义的. 在第十章, 笔者介绍了如何计算问卷调查所产生的众多比例中有多少垃圾. 作者还给出了详尽的计算代码, 以及这些代码的含义和使用. 希望读者能够在此基础上开发出更加有效的程序.

1.3.6 路径模型/结构方程模型

还有一种调查是在已经确定的模型基础上设计的, 例如顾客满意度调查是依照顾客满意度模型设计的, 这些模型被称为路径模型, 因子分析是路径模型的一个特例. 处理路径模型数据的方法主要有两种, 一种是协方差方法; 另一种是偏最小二乘方法. 路径模型也称结构方程模型, 但由于一些关于结构方程模型的书仅介绍协方差方法, 许多人把结构方程模型和协方差方法等同起来, 这是不合适的. 协方差方法严格要求假定数据是正态的, 而偏最小二乘方法则没有对分布做要求. 路径模型分为结构模型和测量模型两部分, 其中结构模型涉及的都是不可观测的隐变量之间的关系, 测量模型则涉及隐变量和可观测变量之间的关系. 协方差方法无法计算隐变量的值, 因此只有偏最小二乘法才能应用于诸如满意度之类的模型. 第十一章首先介绍路径模型的基本概念和方

法, 然后主要介绍偏最小二乘方法, 最后简单介绍协方差方法.

1.3.7 贝叶斯网络

贝叶斯网络是一种用途很广的图模型方法, 贝叶斯网络看上去有些类似于第十一章的图模型(结构方程模型). 但不像完全主观构建而又很难更新或修改的图模型, 贝叶斯网络根据数据中变量之间的相关来构建图中的变量联系, 同时也可以加以主观的介入. 因此贝叶斯网络从某种意义上比图模型更加客观, 与数据所反映的现实世界的关系也更加密切. 第十二章介绍贝叶斯网络及如何构建贝叶斯网络的计算.

1.3.8 多元时间序列数据

一般国内时间序列教科书都以一元时间序列为主要内容, 这是因为一元时间序列的数学比较容易. 当然, 试图用时间序列做预测的人必须假定, 这些序列的发展不会受到其他变量的影响, 或者至少这些序列的未来可以由其历史来预测. 对于多元时间序列, 人们可能用VAR、VARX、状态空间模型等方法处理. 在计量经济学中有一套分析这些数据的检验方法. 第十三章首先回顾时间序列的基本概念, 然后介绍单位根检验、协整检验、Granger因果检验等方法及用VAR、VARX和状态空间模型来拟合数据. 最后介绍了在计量经济学中使用的自门限自回归模型(SETAR)、logistic平滑过渡自回归模型(LSTAR)、神经网络模型、可加AR模型、模型比较、门限协整等内容.

1.4 R软件入门

1.4.1 简介

R软件(R Development Core Team, 2011)用的是S语言, 其运算模式和C语言, basic, matlab, maple, gauss 等很类似. R软件是免费的自由软件, 它的代码公开, 可以修改, 十分透明和方便. 大量国外新出版的统计方法专著都附带有R程序. R软件有强大的帮助系统, 其子程序称为函数. 所有函数都有详细说明, 包括变元的性质、缺省值是什么、输出值是什么、方法的大概说明以及参考文献和作者地址. 大多数函数都有例子, 把这些例子的代码复制并粘贴到R界面就可以立即得到结果, 十分便捷. 反映新方法的各种程序包(package)可以从R网站上下载, 更方便的是联网时通过R软件菜单的"程序包-安装程序包"选项直接下载程序包.

软件必须在使用中学, 仅仅从软件手册中学习是不可取的, 正如仅仅用字典和语法书无法学会讲外语一样. 笔者用过众多的编程软件, 没有一个是从课

堂或者手册学的, 都是在分析数据的实践中学会的. 笔者在见到R软件时, 已经接近"耳顺"之年, 但在一天内即基本掌握, 一周内可以熟练编程和无障碍地实现数据分析目的. 古稀糊涂之翁尚能学懂, 何况年轻聪明的读者乎! 近年出现了大量关于R软件的英文参考资料, 很容易下载, 其代码可以复制和粘贴到R软件上(笔者当年可没有如此幸运). 其中有一个是Vincent Zoonekynd编写的*Statistics with R*, 实际上是网络书籍, 也可以下载其pdf版本[1], 很容易使用; 还有一本书是*Modern Applied Statistics with S* (Venables and Ripley, 2002), 也可以下载其第四版(2002年)的pdf版本[2], 该书通过R软件来介绍许多现代统计方法, 是非常好的一本书, 有一个R自带的程序包(`MASS`)就是以这本书的内容编写的(MASS为该书名字的缩写).

1.4.2 安装和运行小贴士

- 登录R网站(`http://www.r-project.org/`)[3], 根据说明从你所选择的镜像网站下载并安装R的所有基本元素.
- 向左边变元赋值语句可以用"="或者"<-"; 还可以用"->"向右赋值.
- 运行时可以在提示码">"后逐行输入指令. 如果回车之后出现"+"号, 则说明你的语句不完整(得在+号后面继续输入)或者已输入的语句有错误.
- 每一行可以输入多个语句, 之间用半角分号";"分隔.
- 所有代码中的标点符号都用半角格式(基本ASCII码). R代码对于字母的大小写敏感. 变量名字、定性变量的水平以及外部文件路径和名字都可以用中文.
- 不一定非得键入你的程序, 可以粘贴, 也可以打开或新建以R为扩展名的文件(或其他文本文件)作为运行脚本, 在脚本中可以用`Ctrl+R` 来执行(计算)光标所在行的命令, 或者仅运行光标选中的任何部分.
- 出现的图形可以用`Ctrl+W`或`Ctrl+C`来复制并粘贴(前者像素高), 或者通过菜单存成所需的文件格式.
- 如果在运行时点击`Esc`, 则会终止运行.
- 在运行完毕时会被问到"是否保存工作空间映像?" 如果选择"保存", 下次运行时, 这次的运行结果还会重新载入内存, 不用重复计算, 缺点是占用空间. 如果已经有脚本, 而且运算量不大, 一般都不保存. 如果你点击了"保存", 又没有输入文件名, 这些结果会放在所设或默认的工作目录下名为"`.RData`"的文件中, 你可以随时找到并删除它.

[1]网址为: http://zoonek2.free.fr/UNIX/48_R/all.html.
[2]网址为: http://www.planta.cn/forum/files_planta/modern_applied_statistics_with_s_192.pdf.
[3]网上搜索"R"即可得到其网址.

- 注意, 从PPT或Word文档之类非文本文件中复制并粘贴到R上的代码很可能存在由这些软件自动变换的首字母大写或者左右引号等造成的R无法执行的问题.
- R中有很多常用的数学函数、统计函数以及其他函数. 可以通过在R的帮助菜单中选择"手册(PDF文件)", 在该手册的附录中找到各种常用函数的内容.
- 在R界面, 你可以用问号加函数名(或数据名)来得到该函数或数据的细节, 比如用"`?lm`"可以得到关于线性模型函数"`lm`"的各种细节. 另外, 如果想查看在MASS程序包中的稳健线性模型"`rlm`", 在已经打开该程序包时(用`library(MASS)`打开, 用`detach(package:MASS)`关闭), 可用"`?rlm`"来得到该函数的细节. 如果MASS没有打开[1], 或者不知道`rlm`在哪个程序包, 可以用"`??rlm`"来得到其位置. 如果对于名字不清楚, 但知道部分字符, 比如"`lm`", 可以用"`apropos("lm")`"来得到所有包含"`lm`"字符的函数或数据.
- 如果想知道某个程序包有哪些函数或数据, 则可以在R的帮助菜单上选择"Html帮助", 再选择"Packages"即可得到你的R上装载的所有程序包. 这个"Html帮助"很方便, 可以链接到许多帮助(包括手册等).
- 有一些简化的函数, 如加减乘除乘方("`+`, `-`, `*`, `/`, `^`")等, 可以用诸如"`?"+""`"这样的命令得到帮助(不能用"`?+`").
- 你还可以写关于代码的注释: 任何在"`#`"号后面作为注释的代码或文字都不会参与运行.
- 你可能会遇到无法运行过去已经成功运行过的一些代码, 或者得到不同结果的现象. 原因往往是这些程序包经过更新, 一些函数选项(甚至函数名称和代码)都已经改变, 这说明R软件的更新和成长是很快的. 解决的办法是查看该函数, 或者查看提供有关函数的程序包来探索一下究竟.
- 有一个名为"RStudio"可以自由下载的软件能更方便地用几个窗口来展示R的执行、运行历史、脚本文件、数据细节等过程.

1.4.3 动手

如果你不愿意弄湿游泳衣, 即使你的教练是世界游泳冠军, 即使你在教室里听了几百小时的课, 你也永远学不会游泳. 如果你不开口, 即使你熟记了字典中所有英文单词的音标, 即使你完全明白英语语法, 你也永远学不会说英语. 软件当然要在使用中学. R软件的资源丰富, 功能非常强大, 我们不可能也没有必要把每一个细节都弄明白, 有很多功能笔者到现在也没有用到, 或者是因为没有需要, 或者是因为有替代方法. 我们都有小时候读书的经验, 能看懂多少就

[1]通常为了节省内存以及避免变量名称混杂, 应该在需要时打开相应的程序包, 不需要时关闭.

看懂多少, 很少查字典, 后来长大了, 在开始学外语时, 由于大量单词不会才对每个不认识的单词查字典. 实际上, 学外语时, 在有一定单词量的情况下, 能猜就不查字典可能是更好的学习方式.

本书最后的附录"练习: 熟练使用R软件"提供了一些笔者为练习而编的代码, 如果全部一次运行, 用不了一分钟. 但希望读者在每运行一行之后就思考一下. 一般人都能够在很短的一两天将这些代码完全理解. 如果在学习以后章节的统计内容时不断实践, R语言就会成为你自己的语言了.

建议初学R者, 在读本书之前, 务必花些时间, 运行一下这些代码!

1.5 国内统计教学(课本)的若干误区

1.5.1 假设检验的误区: 不能拒绝就接受

除了像两点分布那样的理论探讨之外, 在目前数理统计教科书的内容范畴中, 当p值被认为不够小而不能拒绝零假设时, 只能够说"目前没有足够证据拒绝零假设", 而绝对不能说"接受零假设". 在实际数据分析中, 人们只能够得到"拒绝零假设时可能犯错误的风险"(相应于p值[1]), 而得不到"接受零假设时可能犯错误的概率"(这是不可能得到的概率[2]). 国内某些"权威"教科书长期错误使用"接受零假设"的说法, 其原因可能是觉得任何统计分析一定要有结论, 实际上, 在给出任何统计结论时, 必须给出相应于该结论可能产生的风险. 提供决策建议而又不说明风险是不负责任的. 无法给出风险的"接受零假设"的决策是绝对不能做的. 无论"权威"如何说, 我们都应该运用自己的大脑来思考. "接受零假设"的说法已经成为中国特色, 近40多年来没有见到国外教科书有这种说法.

下面我们用一个数值例子来说明"接受零假设"说法的荒谬. 对自然数列(当然不是正态分布)做两种正态性检验: 一种是对从1到50的自然数列做Shapiro-Wilk正态性检验; 另一种是对从1到500的自然数列做Kolmogorov-Smirnov正态性检验. 它们的p值分别为0.05809及0.0667, 结果是两种正态性检验在0.05的显著性水平下都无法拒绝(正态性的)零假设.

```
> shapiro.test(1:50)
        Shapiro-Wilk normality test
data:  1:50
W = 0.9556, p-value = 0.05809
```

[1]即使是p值, 也仅仅是从一个样本得到的, 不能代替整个总体.

[2]在理论探讨时所用的零假设和备选假设分别都是一点时, 零假设和备选假设对称, 那时用"接受零假设"的说法似乎有些意义, 但这种"两点假设"的问题没有普遍的实际意义.

```
> x=1:500;ks.test(x,"pnorm", mean(x), sd(x), exact = T)

        One-sample Kolmogorov-Smirnov test

data:  x
D = 0.058, p-value = 0.0667
alternative hypothesis: two-sided
```

难道我们能说“接受1到50或者1到500的自然数来自正态分布”吗？很多人把很荒唐的事情当成理所当然的，这或者是因为不动大脑，或者是因为盲目迷信国内“权威”教科书.

读者可以试试，只要样本量小，就往往会得到不能拒绝零假设的结论，比如$x=(-2000,-5000), y=(20000,50000)$，我们设零假设为这两个数据所代表的均值相等，即$H_0:\mu_x=\mu_y$，并做两样本均值的双边$t$检验:

```
> x=c(-2000,-5000);y=c(20000,50000)
> t.test(x,y)

        Welch Two Sample t-test

data:  x and y
t = -2.5539, df = 1.02, p-value = 0.2336
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -221393.3  144393.3
sample estimates:
mean of x mean of y
    -3500     35000
```

从计算机输出可以看出，一个数据的均值为-3500，另一个为35000，但由于p值为0.2336，在一般的显著性水平下(只要小于0.2336)就无法拒绝零假设，难道我们应该说“接受”两个均值相等的零假设吗?

1.5.2 假设检验的误区: p值小于0.05就显著

“p值是多少才算小概率”取决于显著性水平的取值，也就是说，给定显著性水平α，当p值小于α时应拒绝零假设. α取多少完全依赖于问题本身. 对于80年前的Fisher时代，对于农业试验，概率为0.05可以认为很小，就算显著，这不会引起任何争议. 那么为什么不取0.048，或者0.053? 这可能是人们习惯的“洁

癖"所致, 其实没有多大区别, 不必太纠结.

关于显著性水平取多少, 看下面例子: 如果1000对父子(父女)做亲子鉴定, 有50个鉴定错了(误差0.05), 或者有5个鉴定错了(误差0.005), 你能按照显著性水平0.05来认为这是小概率事件吗? 显然不能, 即使是0.001, 也不能算是小概率事件, 鉴定机构肯定脱不了干系. 不能让"0.05"这个数字把自己的头脑禁锢了. 任何时候都要以问题的性质为出发点, 绝对不能盲目跟随某些定式思维教科书的并非负责任的暗示.

此外, p值仅仅是由一个样本计算出来的, 对其含义的任何夸大或过分理解都是不可取的.

1.5.3 置信区间的误区

如果从数据根据公式$\bar{x} \pm t_{\alpha/2}\frac{s}{\sqrt{n}}$算出来的均值$\mu$的置信区间为$(2.3, 4.5)$, 那么能不能说区间$(2.3, 4.5)$以概率$1-\alpha$覆盖$\mu$? 上述说法显然不对, 因为$\mu$和区间$(2.3, 4.5)$都是固定的数, 没有随机性可言, 不能出现任何概率. 区间$(2.3, 4.5)$是否覆盖μ, 不可能知道. 人们只能够说"对于无穷多个不同样本(样本量均为n), 根据公式$\bar{x} \pm t_{\alpha/2}\frac{s}{\sqrt{n}}$算出来的无穷多个区间中大约有$1-\alpha$比例的置信区间覆盖$\mu$, 但到底哪些覆盖, 谁也不知道". 或者能够说"对于随机变量X而言, 随机区间$\bar{X} \pm t_{\alpha/2}\frac{s}{\sqrt{n}}$以概率$1-\alpha$覆盖$\mu$", 但这个结论和用具体数据算出来的数字区间意义不一样, 和具体样本无关. 此外, 上面的置信区间论述是假定样本为独立同正态分布的. 你敢保证样本满足这个条件吗?

1.5.4 最小二乘线性回归中的误区

1. 大多是主观的假定

对于普通最小二乘线性回归:

- **往往假定了模型的线性形式**. 但是, 大家考虑一下, 世界上有多少关系是线性的呢? 这种线性假定不仅存在于回归当中, 而且几乎存在于所有统计方向. 这是因为数学对于线性假定的情况最有办法, 即使对应于非线性假定的情况, 也常常要利用诸如Taylor展开一类的方法把问题转换成线性的来处理. 这也说明了目前人们掌握的数学工具的局限性以及使用和开发机器学习一类方法的必要性.
- **往往假定了样本点是独立同分布的**. 人们不禁要问: 世界上有多少关系是独立的? 有多少变量在抽样时可以假定分布不变? 这种假定也大多是为了数学上的方便.

- **往往假定了样本点有正态分布或者样本量"足够大"**. 前面说过, 没有任何人能够证明某一实际数据来自于任何分布. 而假定"大样本"是为了使用需要正态假定的各种数学结论. 但谁又能说你的样本量足够大呢? (请看本节后面关于大样本的论述.)
- **采用"最小二乘法"本身意味着你选择的损失是对称的二次函数形式**. 请考虑下面问题的损失对称性: 一包西洋参的分量多一些或者少一些对于顾客和对于卖家的损失一样吗? 把一个癌症病人说成正常和把正常人说成患癌症损失一样吗? 为什么损失函数取二次函数, 而不是绝对值? 而不是非对称函数? (参见后面分位数回归部分关于损失函数的论述.)

2. **对回归结果缺乏根据的"解释"**

- **误区**: "**当其他变量不变时, 某变量系数的大小是该变量增加一个单位时因变量所增加的部分(对因变量的贡献)**". 这种说法仅仅在变量独立以及模型的一大堆假定正确时有效. 或者对在干预情况下的实验数据近似有效. 如果对于一般的观测数据你也非要这么说, 那么请问: 你能证明变量的独立性和模型假定正确吗? 永远无法证明, 也没有人能够证明! 不幸的是, 这种说法经常出现在"经济统计"的教科书中, 而各种经济指标恰恰大多是不独立的. 这和下面的不恰当叙述是等价的: "**做线性回归拟合可以从系数大小知道各个变量对因变量的影响大小**." 下面例1.1表明这种说法是谬误的.
- **误区**: "**进行t检验或F检验时, p值较小的变量就较显著.**" 你能证明数据是独立同正态分布的吗? 你能证明模型是线性的吗? 你能证明是"大样本"吗? 如果不能, 最好别这样说. (参见下面例1.1)
- **误区**: "**R^2越接近于1, 说明模型越合适.**" 你用铅笔信手把所有观测值用任何曲线(或折线)连接起来作为回归线时, 必然有$R^2 = 1$, 但这可能仅仅是过拟合的毫无意义的"回归".

例 1.1 经济数据(jingji.csv) 这是有"资本"、"就业"、"电力"和"GDP"四个变量的数据. 运行下面的代码下载数据并做各种回归:

```
z=read.csv("jingji.csv")
a=lm(GDP~就业,z);summary(a)
a=lm(GDP~资本+就业+电力,z);summary(a)
a=lm(GDP~就业+电力,z);summary(a)
a=lm(GDP~资本+就业,z);summary(a)
a=lm(GDP~资本+电力,z);summary(a)
```

这里有五个回归方程, 根据计算机输出, 得到五条拟合直线:

$$\text{GDP} = -85955.90 + 27.41 \times \text{就业}$$
$$\text{GDP} = 2842.35 + 1.06 \times \text{资本} - 1.21 \times \text{就业} + 3.78 \times \text{电力}$$
$$\text{GDP} = -11523.98 + 3.06 \times \text{就业} + 7.37 \times \text{电力}$$
$$\text{GDP} = 11725.73 + 2.05 \times \text{资本} - 3.72 \times \text{就业}$$
$$\text{GDP} = -1088.52 + 0.93 \times \text{资本} + 4.07 \times \text{电力}$$

而关于各个变量在各个模型中的t检验的p值列在表1.1中. 从表1.1可以看出, 这些模型的系数很难说明任何问题, 各个变量的系数在不同模型中变化很大, p值差别也很大. 比如变量"就业"在第一个模型中非常显著, 而在后面三个模型中就很不显著, 其系数甚至还有两次为负数. 这种现象在自变量之间不独立的情况下是普遍的, 这时, 最小二乘法所估计的个别回归系数很难说有什么意义.

表 1.1 例1.1 各个变量在不同模型下t检验的p值

模型	变量t检验的p值		
	资本	就业	电力
GDP ~ 就业	—	6.14×10^{-5}	—
GDP ~ 资本 + 就业 + 电力	0.0145	0.5602	0.0152
GDP ~ 就业 + 电力	—	0.198	1.74×10^{-6}
GDP ~ 资本 + 就业	1.65×10^{-6}	0.201	—
GDP ~ 资本 + 电力	0.00427	—	0.00456

有人认为, "经典回归的好处在于有可以用数学语言表示的'意义明确'的公式," 并且以此来否定"看不见、摸不着"的诸如决策树(决策树其实看得见)、随机森林等机器学习模型. 其实, 错误但又吸引眼球的东西危害最大. 相信这些人并不反对用电脑、智能手机等功能不能用数学公式表达的装置.

3. 增加无关的("错误的")自变量对预测会不会有影响

某国家机关科研所的一位负责人说: 即使自变量采取了与因变量无关的周期变量, 利用拟合数据所得到的模型做出的预测结果也会出现周期性. 是这样的吗? 我们模拟一个因变量(y)和自变量(x1)的线性模型数据, 然后再独立地模拟一个周期性自变量(x2), 最后拟合y~x1及y~x1+x2(请运行下面代码):

```
set.seed(10);x1=rnorm(2000)
y=3+5*x1+rnorm(2000)
a=lm(y~x1);summary(a)
x2=sin(rnorm(2000,10))#加上一个无关的周期变量
```

```
a2=lm(y~x1+x2);summary(a2)#再次拟合
```

这两次回归结果在表1.2和表1.3之中.

表 1.2 回归y ~ x1

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.9831	0.0230	129.82	0.0000
x1	4.9922	0.0226	221.14	0.0000

表 1.3 回归y ~ x1 + x2

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.9844	0.0262	114.09	0.0000
x1	4.9922	0.0226	221.08	0.0000
x2	0.0039	0.0382	0.10	0.9187

显然, 增加这个变量对方程没有什么影响, 两次拟合x1的系数都是4.9922, 非常显著, 而增加的周期变量x2完全不显著(p值为0.9187), 其拟合系数为0.0039. 实际上, 增加与因变量无关的自变量不会对回归结果有多大影响. 那个负责人完全凭自己的想象代替科学, 实在有些可悲, 这也说明职位不能代替知识.

1.5.5 样本量是多少才算大样本

"大样本"经常用来表示可以使用中心极限定理来说明线性回归的F检验和t检验有效(因为均值渐近正态). 但在实际数据分析中, 有的教科书说样本量是30就可以认为是大样本. 这种说法不负责任, 会严重误导读者. 我们看下面的模拟例子.

运行下面的模拟和检验程序会产生5000个样本量为100000的$t(2)$分布的样本均值, 并检验这些均值是否为正态分布.

```
set.seed(10);y=NULL;for(i in 1:5000)
{y=c(y, mean(rt(100000,2)))}
shapiro.test(y)# 得到W = 0.8309, p-value < 2.2e-16
```

得到的p值等于2.2×10^{-16}. 这说明, 即使是在完全符合中心极限定理的假定时, 即使样本量$n = 100000$, 所得到的均值正态性假设还会被拒绝(对于几乎任何显著性水平). 注意, 由于这里是模拟, 所以可以认为中心极限定理的条件满足, 但对于实际数据, 该定理的条件根本无法核实.

显然"$n = 30$就算是大样本"的断言是荒唐的. 实际上, 在证明各种"大样本定理"的统计学家中, 没有人愿意说多么大才算是大样本. 大样本定理的结论对于样本量$n \to \infty$是有意义的. 但你能够说清楚你的n与∞差多远吗?

1.5.6 用31个省、直辖市、自治区数据能做什么

在宏观经济统计领域中, 很多人都用中国统计局公布的31个省、直辖市、自治区的数据来做回归等统计分析, 但是究竟有多少人想过下面的问题?

- **31个省、直辖市、自治区数据仅仅是一些汇总数据, 根本不是样本**. 因此完全不满足任何适用于样本的统计推断方法.
- **如果一定要说31个省、直辖市、自治区数据是样本, 那么总体是什么?** 如果还有疑问, 好好温习一下初等概率统计知识就会明白了.
- **此外, 31个省、直辖市、自治区数据互相根本不独立.**
- **由于不是任何总体的样本, 更谈不上分布及正态性.**
- **对于31个省、直辖市、自治区数据, 使用任何超出描述性统计方法的做法都值得怀疑.**
- **有人还用31个省、直辖市、自治区数据做空间自回归, 这是非常不妥的.** 除了它不是样本之外, 做空间统计的单位太大, 比如内蒙古既和沿海省份相连, 又和最落后的省份相接, 任何空间距离的描述都很片面, 而且边境省份也没有和周边国家/地区相关, 似乎中国完全处于空间中自我封闭的状态.

1.5.7 汇总数据(比如部分均值)和原始观测值的区别

很多人喜欢用汇总数据, 比如用各地区变量的均值来做诸如回归那样的推断. 这时会出现很多问题. 下面把一个有600个观测值的原始数据分成4组、12组及30组, 然后求出每组的均值, 看各组的相关系数的差别, 并点出均值的散点图(图1.2). 图1.2显示, 原来基本独立的(相关系数为0.01)两组数据在分组并取均值之后, 它们的相关系数从4个均值的0.91, 12个均值的0.22到30个均值的-0.17. 事实上, 当只有两组时, 相关系数为1, 随着组的个数增加, 各组均值的相关系数应该接近原始数据的相关系数.

从这个例子可以看出, 无论在什么情况下, 都最好使用原始数据. 如果使用原始数据分组后的各组均值来替代原始数据, 则组的数目越少, 结果越不可靠.

产生图1.2的代码为:

```
set.seed(999);n=600
w=data.frame(x=rnorm(n),y=rnorm(n))
I=c(4,12,30);par(mfrow=c(2,2))
```

```
plot(w,main=paste("n=",n," r=",round(cor(w)[1,2],2)))
for(j in 1:length(I)){
z=NULL;u=I[j];for(i in 1:u)
z=rbind(z,apply(w[((i-1)*n/u+1):(i*n/u),],2,mean))
plot(z,main=paste("n=",u," r=",round(cor(z)[1,2],2)))}
```

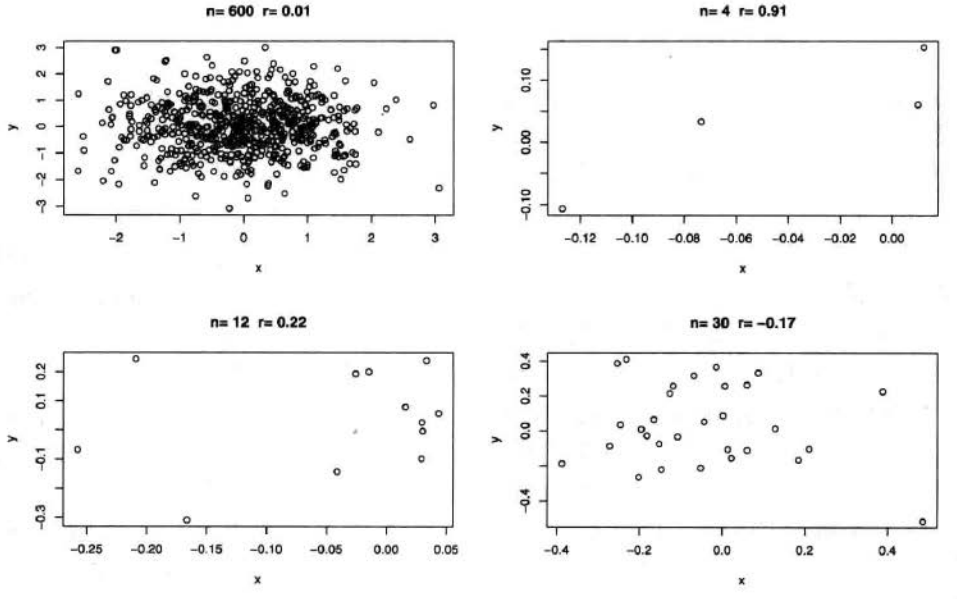


图 1.2 对样本量为600的数据分组求均值, 按照均值作散点图并求相关系数r

第二章　横截面数据回归: 经典方法

2.1　简单回归回顾

横截面数据是标准回归分析教科书的主要研究对象, 对这类数据最简单的假定是其因变量$\boldsymbol{Y}$和自变量$\boldsymbol{X}$满足线性回归模型[1], 其方程为:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

式中, 因变量$\boldsymbol{Y}$为n维向量; 自变量$\boldsymbol{X}$为$n \times p$矩阵; 误差项$\boldsymbol{\epsilon}$为n维向量. 这意味着一共有n个观测值, 有一个由向量$\boldsymbol{Y}$代表的因变量及由$\boldsymbol{X}$代表的p个自变量(如果$\boldsymbol{X}$矩阵包含全是1的代表截距的一列, 则有$p-1$个自变量).

在简单回归中, 误差项$\boldsymbol{\epsilon}$的元素一般要求是独立同分布零均值的, 而通常分布假定为正态的. 在最小二乘回归的标准输出中, 对系数的t检验和方差分析的F检验是教科书中必讲的内容, 而p值小就意味着"显著", 但很多教科书避而不谈误差项是否偏离正态性, 也很少谈到对残差的正态性的验证. 实际上, 如果正态性或渐近正态性不成立, 那些t检验和F检验就没有任何意义.

当然, 我们无法证明某个数据的确来自某一总体分布, 但可以利用假设检验来拒绝其来自某分布, 而且要给出相应的关于显著性的p值. 如果不能拒绝, 我们则称"没有足够证据拒绝零假设", 零假设可以在后续研究中作为一个假定, 而不是确定的结论, 仅此而已. 此外, 对不满足诸如正态性之类假定的模型也可以进行相互比较, 但所用的方法不是这些基于正态性的检验, 可以使用AIC之类的准则或者交叉验证来比较.

下面介绍一个横截面数据回归的例子, 首先试图用初等经典线性回归方法来拟合这个数据, 然后根据数据的一些特点对该数据的自变量做些变换,再重新拟合. 我们先利用这个例子来复习一下大家可能已经学过的线性回归的知识. 注意, 该数据并不是原始数据, 而是加工过的数据. 有24个数据观测值, 每个都是夏天所有工作日相应时段的平均值.

例 2.1　公路一氧化碳数据(cofreewy.txt)　这个来自Ledolter and Tiao (1979)和Hogg and Ledolter (1992)的数据[2]是洛杉矶一个站点从午夜开始的24小时中所测量的一氧化碳(变量CO)在整个夏天(周末除外)的平均值, 另外还有3个关于时间、地点及风速的变量, 因此数据为24×4的矩阵. 变量具体情况见表2.1.

---

[1]线性模型是最容易处理的模型, 但实际数据是否满足这个简单模型, 很难验证, 因此, 在回归中必须对各种模型进行比较, 才能找到**相对来说**较适合的模型.

[2]该数据可以从网站http://www.statsci.org/data/general/cofreewy.html下载.

表 2.1 例2.1 公路一氧化碳数据变量情况

变量名	描述	性质
CO	一氧化碳平均值	定量变量
Hour	时间点	定量变量
Traffic	交通密度(交通车辆数目除以交通速度)	定量变量
Wind	风速的竖直分量	定量变量

2.1.1 对例2.1数据的简单拟合

对这个数据的最简单的处理就是最小二乘回归. 也就是用一个以CO为因变量, 其他3个变量为自变量的回归模型来拟合数据:

$$\mathrm{CO} = \beta_0 + \beta_1 \mathrm{Hour} + \beta_2 \mathrm{Traffic} + \beta_3 \mathrm{Wind} + \epsilon.$$

这可以用下面(包括输入数据)的代码得到:

```
w=read.table("COfreewy.txt",header=T)#读入数据
a=lm(CO~.,w)#利用全部自变量做线性回归
summary(a)#展示结果
b=step(a,direction="backward")#逐步回归
summary(b)#展示逐步回归结果
shapiro.test(b$res)#做残差的正态性检验
```

其中, 我们应用了AIC [1] (Akaike information criterion)在逐步回归语句(函数`step()`)中选择变量. 结果舍弃了Hour变量. 回归结果见表2.2.

表 2.2 公路一氧化碳数据简单回归结果表

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.2745	0.1981	6.43	0.0000
Traffic	0.0183	0.0013	13.62	0.0000
Wind	0.1747	0.0568	3.08	0.0057

得到的确定系数[2] R^2及调整的R^2分别为0.950及0.945. 根据这种回归的结果, 拟合的回归直线为:

$$\mathrm{CO} = 1.275 + 0.018\mathrm{Traffic} + 0.175\mathrm{Wind}.$$

[1] AIC的一般公式为AIC $= 2k - 2\ln L$. 式中, k为参数个数; L为估计模型的最大化的似然函数值. 如果在回归中, 误差项是独立同正态分布的, n为样本量, RSS为残差平方和, 则AIC $= 2k + n\ln(RSS/n)$.

[2] 回归中的确定系数(可决系数)定义为$R^2 \equiv 1 - SS_{err}/SS_{tot}$, 这里$SS_{tot} = \sum_i (y_i - \bar{y})^2$; $SS_{err} = \sum_i (y_i - \hat{y}_i)^2$.

关于残差的Shapiro-Wilk正态性检验的p值为0.05601, 看上去似乎在0.05的显著性水平下不能拒绝残差来自正态总体的假定. 对于一般的回归教科书来说, 根据打印的结果, 这个问题似乎已经圆满解决, 不用再去怀疑了.

2.1.2 对例2.1数据的进一步分析

不断怀疑和探索是科学的态度, 首先查看例2.1简单回归残差的QQ图(见图2.1), 该图是用下面代码产生的:

```
qqnorm(b$res);qqline(b$res)
```

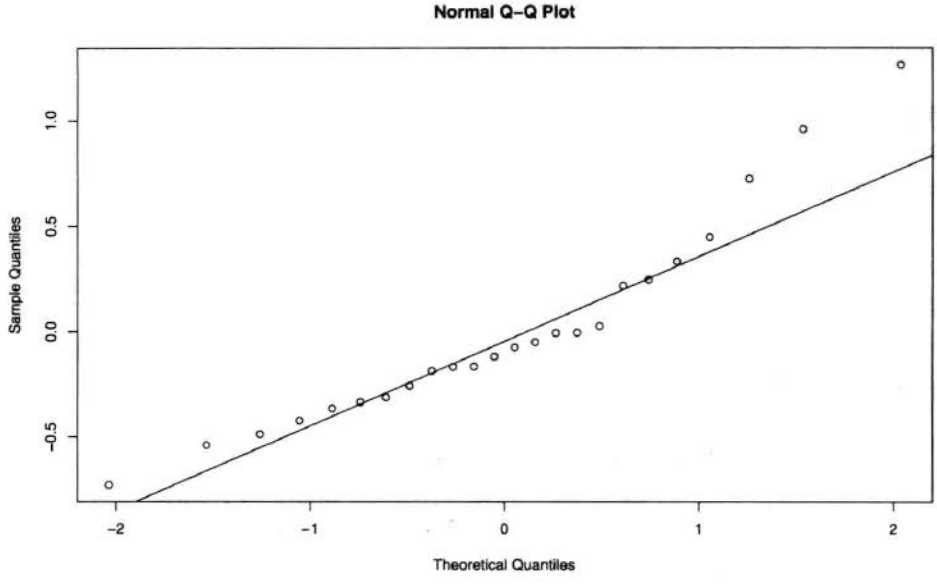


图 2.1 例2.1简单回归残差的QQ图

根据图2.1, 这个残差很难说是正态的. 虽然正态性检验在显著性水平0.05下无法拒绝零假设, 但一般来说, 数据量小的时候, 往往不易拒绝零假设. 不能拒绝零假设, 只能说明对于这个检验, 证据不足以拒绝零假设而已, 并不能说明零假设正确. 此外, 对于样本量如此小的数据来说, 0.05的显著性水平似乎也偏高.

对这个数据再作进一步探索. 首先可以把各对变量用散点图(图2.2)表示. (用代码`plot(w)`或者`pairs(w)`产生).

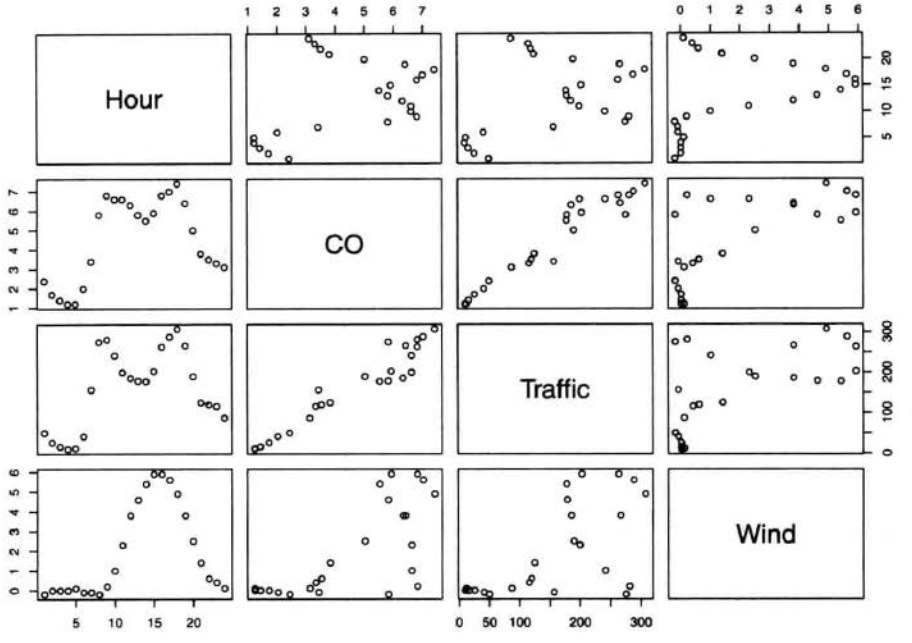


图 2.2 例2.1数据各变量之间的两两散点图

从这个散点图可以看出, CO和Traffic似乎有些线性关系, CO和Hour则有些类似于正弦曲线的关系, Hour和另外两个变量也有些周期形式的关系, 而CO和Wind的关系比较复杂, 很难用线性关系表示. 鉴于交通密度及风速都有某种随时间变量(Hour)变化的周期性, 借助时间序列分析所用的谐波分析(可参看Chatfield, 2004, p.126), 可以用有穷Fourier级数来代表时间序列$\{x_t\}$:

$$x_t = a_0 + \sum_{p=1}^{(N/2)-1} [a_p \cos(2\pi pt/N) + b_p \sin(2\pi pt/N)] + a_{N/2} \cos(\pi t),$$

这里的系数由x_t得出. 这个分析可以作为对Hour做变换的参考. 此外, 可以利用线性相关系数考察各变量的二次项和三次项对CO的影响(表2.3).

表2.3中, Tsq和Tcub代表Traffic的二次项和三次项, Hsq和Hcub代表Hour的二次项和三次项, Wsq和Wcub代表Wind的二次项和三次项. 从这里我们可以看出哪些变量的哪些高阶项可以放入候选模型. 先试一个较大的模型, 即把Traffic, Wind的一二三次项都包含进去, 再包含上面谐波分析中的$p = 1, 2$的项. 这个模型明显偏大, 我们可以根据AIC用逐步回归来选择变量.

表 2.3 公路一氧化碳数据线性相关系数表

	CO	Traffic	Tsq	Tcub	Hour	Hsq	Hcub	Wind	Wsq	Wcub
CO	1.00	0.96	0.90	0.83	0.43	0.24	0.10	0.71	0.63	0.57
Traffic	0.96	1.00	0.97	0.91	0.43	0.25	0.12	0.61	0.56	0.52
Tsq	0.90	0.97	1.00	0.99	0.30	0.14	0.02	0.56	0.53	0.50
Tcub	0.83	0.91	0.99	1.00	0.24	0.09	−0.01	0.50	0.49	0.47
Hour	0.43	0.43	0.30	0.24	1.00	0.97	0.92	0.42	0.33	0.29
Hsq	0.24	0.25	0.14	0.09	0.97	1.00	0.99	0.26	0.18	0.16
Hcub	0.10	0.12	0.02	−0.01	0.92	0.99	1.00	0.12	0.05	0.03
Wind	0.71	0.61	0.56	0.50	0.42	0.26	0.12	1.00	0.97	0.93
Wsq	0.63	0.56	0.53	0.49	0.33	0.18	0.05	0.97	1.00	0.99
Wcub	0.57	0.52	0.50	0.47	0.29	0.16	0.03	0.93	0.99	1.00

读入数据, 计算相关系数, 做这个较大模型的回归并采取逐步回归方法选择变量的代码为:

```
w=read.table("COfreewy.txt",header=T)#读入数据
attach(w) #把变量名字放入内存
cor(cbind(CO,Traffic,Tsq=Traffic^2,Tcub=Traffic^3,
    Hour,Hsq=Hour^2,Hcub=Hour^3,Wind,Wsq=Wind^2,Wcub=Wind^3))
a=lm(CO~Traffic+Wind+I(Wind^2)+I(Wind^3)+sin((2*pi/24)*Hour)+
 cos((2*pi/24)*Hour)+sin((4*pi/24)*Hour)+cos((4*pi/24)*Hour))
b=step(a) #逐步回归, 按照AIC选择变量
summary(b);anova(b);shapiro.test(b$res)
```

从输出的汇总结果(请读者自己实现)来看, 按照t检验(注意Shapiro-Wilk正态性检验的p值为0.8649), Wind的三次项和Hour的sin项的p值都大于0.2, 于是把它们去掉重新拟合, 代码为:

```
b1=lm(CO~Traffic+Wind+I(Wind^2)+
    cos((2*pi/24)*Hour)+cos((4*pi/24)*Hour))
summary(b1)#结果汇总
anova(b1)#方差分析表
shapiro.test(b1$res)#对残差的正态性检验
qqnorm(b1$res);qqline(b1$res)#作QQ图(请读者自己做, 这里不展示)
```

结果汇总及方差分析表分别显示在表2.4和表2.5中.

表 2.4 公路一氧化碳数据: 最终的回归结果

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.3984	0.1090	12.83	0.0000
Traffic	0.0172	0.0007	24.39	0.0000
Wind	0.4587	0.0933	4.92	0.0001
I(Wind^2)	−0.0573	0.0162	−3.55	0.0023
cos((2 * pi/24) * Hour)	−0.3668	0.0855	−4.29	0.0004
cos((4 * pi/24) * Hour)	0.3862	0.0698	5.53	0.0000

表 2.5 公路一氧化碳数据: 方差分析表

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Traffic	1	95.88	95.88	1926.53	0.0000
Wind	1	2.36	2.36	47.37	0.0000
I(Wind^2)	1	1.41	1.41	28.31	0.0000
cos((2 * pi/24) * Hour)	1	1.39	1.39	28.02	0.0000
cos((4 * pi/24) * Hour)	1	1.52	1.52	30.63	0.0000
Residuals	18	0.90	0.05		

最终选择出来的模型为

$$\begin{aligned}\mathrm{CO} = {} & \beta_0 + \beta_1 \mathrm{Traffic} + \beta_{21} \mathrm{Wind} + \beta_{22} \mathrm{Wind}^2 \\ & + \beta_{31} \cos\left(\frac{2\pi}{24}\mathrm{Hour}\right) + \beta_{32} \cos\left(\frac{4\pi}{24}\mathrm{Hour}\right) + \epsilon.\end{aligned}$$

式中, 系数$\beta_0, \beta_1, \beta_{21}, \beta_{22}, \beta_{31}, \beta_{32}$的估计值显示在表2.4的"Estimate"列, 而对系数t检验的p值在"Pr(>|t|)"那一列. 对残差的Shapiro-Wilk正态性检验的p值为0.6369, 没有足够证据拒绝其来自正态总体. 确定系数R^2和调整的R^2分别为0.9913 和0.9889. 这个结果似乎比2.1.1节中的好一些.

2.1.3 对简单线性回归的一些讨论

在上面的模型选择中既用了基于AIC的逐步回归, 又用了对系数的t检验及F检验. 到底依照什么标准来选择变量呢? 实际上, AIC和那些检验是从不同的角度来判断模型, 由于标准不同, 所选择的模型也可能有很大不同, 从而对数据的解释也不同, 这是经典回归的固有问题. 这正如在物理学中, 一个现象有多种假说来解释, 每一种假说都不是真理, 它们只不过是人们根据自己的标准从不同角度对客观世界的猜想而已.

另外, 诸如t检验和F检验那样的显著性检验的依据是正态性分布, 因此考察正态性是必不可少的. 如果没有了正态性或渐近正态性, 那么这些检验就没有多大意义了. 对模型的各种假定都是为了在判断模型是否适合时有所依据. 如果不想评价模型, 在散点图上信手画一条曲线(对于简单的一元回归), 也完全是一种回归, 只是很难说清楚你这条曲线比另一个人画的曲线更好而已.

在2.1.2节的回归中, 对自变量做了不同的变换, 但这还是简单线性模型, 因为"线性"是对未知参数而言的, 而不是对自变量而言的.

读者可能发现, 对例2.1数据的进一步回归似乎有些人工雕琢的痕迹. 原因之一是该数据的数据量不大, 变量有限又相关, 同时有一些与时间有关的周期性质. 该数据谐波分析的参与不是笔者想出来的, 是数据提供者的杰作, 虽然这里的最终模型和原文献稍有不同, 但并无显著区别. 对一个数据做出比较圆满的分析并不是必然的. 要用人类有限的脑细胞来应对无穷的宇宙, 大多数情况下的结果是束手无策.

即使一个回归的t检验和F检验全部显著, 正态性检验不显著(无法拒绝正态性), 而且R^2很接近1, 也不能说明该回归就有意义. 请试着运行下面的语句, 并且查看各种输出:

```
set.seed(441010)
x=c(rnorm(100),50);y=c(rnorm(100),-50)
a=lm(y~x);summary(a)
shapiro.test(a$res)
```

这是两个完全独立的变量(x和y)的回归, 分别由独立同标准正态分布的100对数字组成, 在它们之间进行回归没有任何意义, 但在各自仅仅增加了一个数之后, 就得到了非常"显著"的结果: R^2为0.9158, 调整的R^2为0.915, 斜率的t检验的p值为2×10^{-16}, Shapiro-Wilk正态性检验的p值为0.711. 但实际上, 这个回归是没有任何意义的, 是一个伪回归. 这个小例子说明, 任何检验都是为了否定(而不是为了肯定什么)而设立的, 你可能会通过检验发现一些问题, 但即使没有发现问题, 也不能说没有问题, 仅仅说明以你目前的手段没有发现问题而已. 统计有一个研究方向是回归诊断, 目的是寻找更多的手段来发现回归中的问题. 但有一点是肯定的, 就是人们永远也不能说"这个回归没有任何问题了".

2.1.4 损失函数及分位数回归简介

正如前面所说, 最小二乘回归使用了对称的二次损失函数. 一般来说, 带有可加误差项的回归模型可以写成下面的形式:

$$y_i=\mu(\boldsymbol{x}_i,\boldsymbol{\beta})+\epsilon_i,$$

式中, μ是一个一般的函数, 如果$\mu(\boldsymbol{x}_i, \boldsymbol{\beta}) = \boldsymbol{x}_i^\top \boldsymbol{\beta}$, 就是线性模型. 在拟合时, 总是希望找到使得残差$y_i - \mu(\boldsymbol{x}_i, \boldsymbol{\beta})$的某个凸函数[1]的和尽可能小的参数(向量)$\boldsymbol{\beta}$. 这就需要一个(凸)损失函数$\rho(\cdot)$. 更确切地说, 在我们选择了$\rho(\cdot)$之后, 希望寻找使得下式最小的$\boldsymbol{\beta}$.

$$\sum_{i=1}^{n} \rho(y_i - \mu(\boldsymbol{x}_i, \boldsymbol{\beta})).$$

对于线性回归模型$y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \epsilon_i$, 这就意味着寻找$\boldsymbol{\beta}$使得

$$\sum_{i=1}^{n} \rho(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})$$

最小. 如果损失函数为二次函数, 则$\rho(u) = u^2$. 这时, 对于线性模型来说, 就是要求使得残差平方和

$$\sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2$$

最小的$\boldsymbol{\beta}$, 也就是最小二乘回归. 如果损失函数为$\rho(u) = |u|$, 则称为最小一乘回归, 它使得残差绝对值的和最小. 最小一乘回归是分位数回归(quantile regression)的特例. 一般的τ分位数回归的损失函数为:

$$\rho_\tau(u) = u(\tau - I(u < 0)).$$

当$\tau = 0.5$时, 就是最小一乘回归.

最小二乘回归和最小一乘回归的损失函数是对称的, 而一般的τ分位数回归的损失函数不是对称的, 而是由两条从原点出发的分别位于第一和第二象限的射线组成, 它们的斜率之比为$\tau : \tau - 1$. 图2.3给出了最小二乘回归的损失函数u^2(左)及分位数回归的两个($\tau = 0.2$和$\tau = 0.6$)损失函数(右).

在实际应用中, 残差为正和残差为负的损失往往是不对称的. 比如, 用车床加工一个金属轴, 其直径如果大于要求, 可以返工, 而如果小于要求, 这个零件就作废了. 这时, 损失函数就不应该是对称的. 由于篇幅有限, 这里对分位数回归仅作简要介绍.

在R中可以用程序包`quantreg`[2]来做分位数回归. 那里有一个关于比利时工人阶级的收入和食品花费的例子, 数据名称为`engel`, 来自Koenker and Bassett(1982). 下面代码计算τ分别为$0.15, 0.25, 0.50, 0.75, 0.95, 0.99$的分位数回归, 同时点出了原始数据图(图2.4左)、做了(以10为底的)对数变换之后的图及6条分位数回归拟合直线(图2.4右). 这些代码均来自该程序包.

[1]这里所说的凸函数是指下凸(convex), 而不是上凸(concave).

[2]Roger Koenker (2011). quantreg: Quantile Regression. R package version 4.76. http://CRAN.R-project.org/package=quantreg.

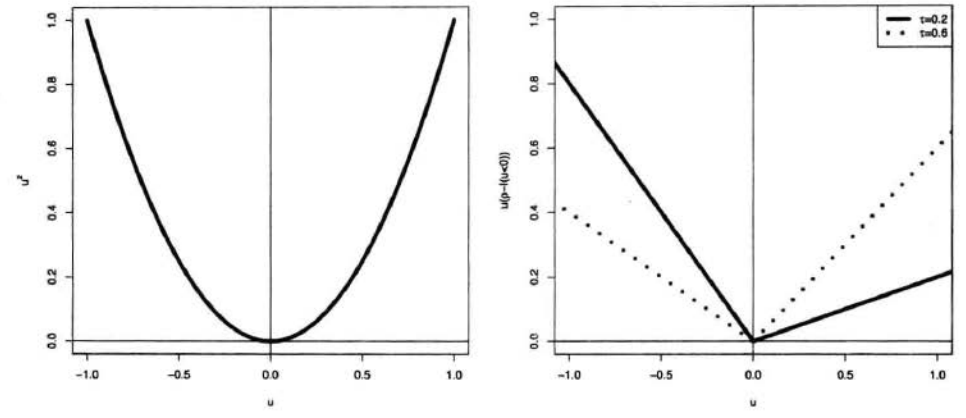


图 2.3 两类损失函数: $\rho(u) = u^2$(左)及$\rho_\tau(u) = u(\tau - I(u < 0))$(右)

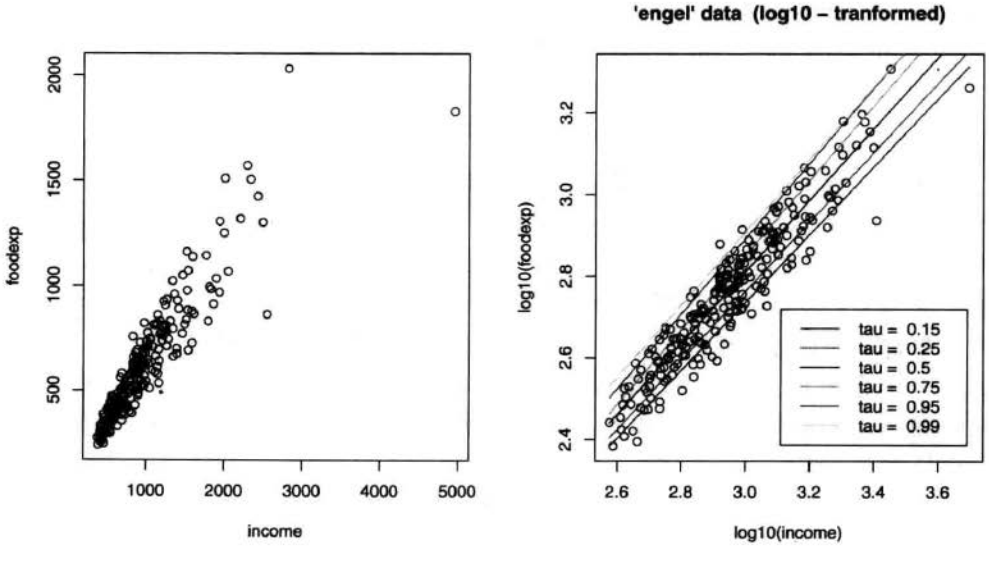


图 2.4 原始的engel数据点图(左)及对数变换之后的点图和6条分位数回归拟合直线(右)

```
library(quantreg)
par(mfrow=c(1,2))
data(engel);plot(engel)
```

```
plot(log10(foodexp) ~ log10(income), data = engel,
    main = "'engel' data  (log10 - tranformed)")
taus <- c(.15, .25, .50, .75, .95, .99)
rqs <- as.list(taus)
for(i in seq(along = taus)) {
    rqs[[i]]=rq(log10(foodexp)~log10(income),
    tau=taus[i],data=engel)
    lines(log10(engel$income),
    fitted(rqs[[i]]),col = i+1)}
legend("bottomright", paste("tau = ", taus), inset = .04,
    col = 2:(length(taus)+1), lty=1)
```

2.2 简单线性模型中的指数变换

在许多情况下, 为了使得经典线性模型所需要的正态性假定成立, 常常对因变量使用指数变换或对数变换(其统一形式为Box-Cox变换), 以使其接近正态形状. 虽然这种变换不一定奏效, 但不妨一试. 如果数据的分布为非单峰的分布, 或者是各种分布的混合, 或者完全没有确定的分布, 那么这种变换是很难奏效的. 有人认为, 多数可以写成数学表达式的分布都不难变换成正态分布, 因此变换是万能的. 但是在实际生活中, 你根本无法确定一个数据来自于何种分布. 一个数据很可能是由许多不同机制产生的混合体, 根本谈不上服从什么分布. 在应用中不能纸上谈兵.

为了说明本节及下一节的各种方法, 选择介绍另外一个横截面数据.

例 2.2 口咽癌数据(pharynx.csv, pharynx1.csv) 这个来自Kalbfleisch and Prentice (1980)的数据[1], 基于美国Radiation Therapy Oncology Group 的几个机构针对口咽若干位置的鳞状细胞癌的临床试验. 该试验分成两组, 其中一组仅仅使用放疗(TX=1), 而另一组则同时使用放疗和化疗(TX=2). 原始数据pharynx.csv为195 × 13的矩阵. 表2.6是变量情况.

在计算机上读取并查看数据pharynx.csv后, 读者可能会发现, 这个数据中的变量CASE和ENTRY.DT不能参与建模, 应该删去. 在数据探索分析中还发现有两个观测值有缺失, 也予以删除. 此外, COND有些水平记录太少, 予以合并. 这样整理过的数据就是193 × 11的矩阵. 下面的分析基于这个整理过的数据(pharynx1.csv).

读取以及整理数据的代码如下:

[1]可以从网站http://www.umass.edu/statdata/statdata/stat-nonlin.html 下载.

表 2.6 例2.2 口咽癌数据变量情况

变量名	描述	变量名	描述
CASE	编号	SITE	病变位置(哑元)
INST	机构代码	T.STAGE	癌症T分期(哑元)
SEX	性别(哑元:1,2)	N.STAGE	癌症N分期(哑元)
TX	试验代码(1:标准,2:处理)	ENTRY.DT	进入试验日期(整数)
GRADE	和正常细胞的区别度(哑元)	STATUS	删失(0:右删失, 1:死亡)
AGE	年龄(定量)	TIME	如未删失, 则是存活时间
COND	身体状况(哑元)		否则是最后记录时间(天数)

```
w=read.csv("pharynx.csv")#读取口咽癌数据
w=w[,-c(1,11)]#去掉CASE和日期
w=w[w$COND!=9&w$GRADE!=9,]#去掉两列NA(这两个变量的NA用9表示)
#把COND变成两个水平:
w$COND[w$COND==3|w$COND==4]=2
w$COND[w$COND==0]=1
#存成pharynx1.csv
write.csv(w,"pharynx1.csv",quote=F,row.names=F)
```

除了TIME和AGE之外都是分类(定性)变量. 变量TIME是这里关心的因变量. 这个数据是典型的生存分析数据, 这里尝试用回归分析加上变换的方法来处理, 也可以用下一节将介绍的生存分析的方法处理, 比如Cox比例危险回归模型(Cox proportional hazards regression model), 当然也可以用其他回归方法处理.

首先不妨对例2.2数据试一下简单的线性回归[1], 在诸如生存分析的Cox回归及Weibull回归等模型出现之前, 人们就是用简单回归来处理这类数据的. 回归的代码如下:

```
u=read.csv("pharynx1.csv")#读入数据
x=1:11;(x=x[-c(5,11)])#定性变量的下标
for(i in x)u[,i]=factor(u[,i])#把定性变量从数值型转换成因子型
a=lm(TIME~.,data=u);summary(a)
```

[1]在这里把STATUS也作为自变量引入, 但在一些生存分析模型中, 这样做可能并不被认可. STATUS描述了是否删失, 删失的原因很多, 它不仅对生存时间的计算有影响, 而且说明患者为了某种原因不愿意继续治疗了. 在生存分析中仅仅考虑了对生存时间的影响, 没有把它作为反映患者状况的一种指标.

我们发现R^2只有0.577, 而调整的R^2为0.528. 此外, Shapiro-Wilk正态性检验的p值只有1.569×10^{-6}. 这个结果显然不理想. 由于正态性条件不满足, 因此那些检验的p值没有多大意义.

变换能不能解决问题呢? 我们可以试试对TIME做指数变换. 利用程序包MASS[1]中的boxcox()函数可以找到回归中的Box-Cox变换的参数λ. 原始的Box-Cox变换(Box & Cox. 1964)的公式为

$$y^{(\lambda)} = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \lambda \neq 0; \\ \ln(y), & \lambda = 0. \end{cases}$$

这个公式是为了统一指数变换和对数变换而设计的. 在实际应用中, 如果确定了λ, 可以直接用指数或对数变换, 不必套用这个公式, 回归效果是一样的. 利用boxcox()函数寻找λ的R代码如下:

```
library(MASS)
b=boxcox(TIME~.,data=u)
I=which(b$y==max(b$y))
b$x[I]#lambda=0.38我们取0.4
```

得到对应不同λ值的对数似然曲线, 并且标出了使得对数似然最大的λ的位置(图2.5), 据此, 可以试试用TIME的0.4次方作为因变量来拟合数据.

对变换后的因变量做逐步回归的代码如下:

```
a=lm(TIME^0.4 ~ INST + SEX + TX + AGE + COND + T.STAGE +
    N.STAGE + STATUS, data = u);b=step(a)
summary(b);anova(b)
shapiro.test(b$res)
```

下面是输出的回归系数(及t检验)和方差分析结果:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.17554    1.18836  10.246  < 2e-16
INST2        0.53984    0.68852   0.784  0.43402
INST3        2.10842    0.68621   3.073  0.00245
INST4        0.87160    0.74709   1.167  0.24488
INST5        0.47751    0.73099   0.653  0.51443
INST6        1.64965    0.72982   2.260  0.02498
SEX2         0.98381    0.44128   2.229  0.02701
```

[1]Venables, W. N. & Ripley, B. D. (2002) *Modern Applied Statistics with S.* Fourth Edition. Springer, New York. ISBN 0-387-95457-0.

```
TX2          -0.62554    0.37581  -1.664  0.09773
AGE           0.04621    0.01703   2.714  0.00728
COND2        -2.93016    0.45307  -6.467 8.92e-10
STATUS1      -4.71723    0.43531 -10.837  < 2e-16

Residual standard error: 2.564 on 182 degrees of freedom
Multiple R-squared:  0.5618,    Adjusted R-squared:  0.5377
F-statistic: 23.33 on 10 and 182 DF,  p-value: < 2.2e-16
```

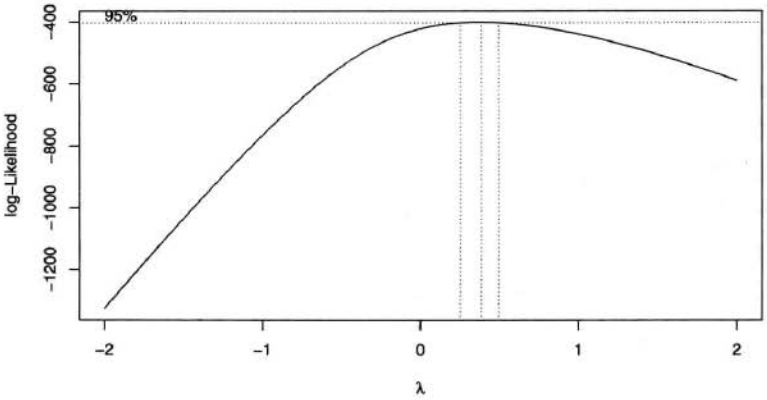


图 2.5 例2.2口咽癌数据Box-Cox变换参数λ的搜寻

```
Analysis of Variance Table

Response: TIME^0.4
           Df  Sum Sq Mean Sq  F value    Pr(>F)
INST        5  148.57   29.71   4.5183 0.0006604
SEX         1   17.88   17.88   2.7189 0.1008922
TX          1   42.20   42.20   6.4163 0.0121524
AGE         1    1.49    1.49   0.2260 0.6350440
COND        1  552.13  552.13  83.9576 < 2.2e-16
STATUS      1  772.26  772.26 117.4304 < 2.2e-16
Residuals 182 1196.89    6.58
```

结果R^2只有0.5618, 而调整的R^2为0.5377. 此外, 关于残差的Shapiro-Wilk正态性检验的p值为0.1473. 这个结果似乎不能否认正态性. 如果认为正

态性条件满足, 那么根据t检验和根据方差分析结果得到的显著的变量不尽相同. 虽然方差分析结果中TX的p值不算太小(0.012), 但似乎还算显著, 这是否说明用放化疗并不比仅用放疗活得更长, 而且删失的也活得长些? 请读者自己计算并解释输出. 虽然关于拟合的F检验很显著(p值小于2.2×10^{-16}), 但从R^2来看, 拟合得并不好, 因此这里所给结论的说服力需要打折扣.

2.3 生存分析数据的Cox回归模型

我们知道, 前一节引进的例2.2是生存分析数据. 一般来说, 生存数据是为特殊目的而收集的, 生存分析的方法广泛应用于生物学、医学、保险学、社会学、心理学、教育学等众多领域. 这里首先简单介绍生存分析的一些基本概念, 然后引进处理横截面生存数据的Cox比例危险回归模型, 这是生存分析的一种标准方法. 在生存分析中, 研究的主要对象是寿命超过某一时间的概率. 当然, 生存分析模型不仅和寿命有关, 还可以描述其他一些事件发生的概率, 例如产品的失效、出狱犯人第一次犯罪、失业人员第一次找到工作、青少年第一次吸毒, 等等.

如果令T为表示寿命的随机变量, 其累积分布为$F(t) = P(T \leqslant t)$, 其分布密度为$f(t)$, 而活过时间t的概率为:

$$S(t) = P(T > t) = 1 - P(T \leqslant t), \quad t > 0.$$

式中, $S(t)$称为生存函数(survival function). 除了生存函数之外, 还有危险函数(hazard function)的概念, 用$h(t)$表示, 它实际上是$-\ln S(t)$关于t的导数:

$$\begin{aligned} h(t) &= \lim_{\Delta t \to 0} \frac{P(t \leqslant T \leqslant t + \Delta t | T \geqslant t)}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{[1 - F(t)]\Delta t} = \frac{f(t)}{S(t)} = \frac{\mathrm{d}}{\mathrm{d}t}[-\ln S(t)]. \end{aligned}$$

累积危险函数为$H(t) = \int_0^t h(u)\mathrm{d}u$. 生存函数、危险函数、累积危险函数在数学上是等价的, 知道其中之一, 就可以推导出其他函数. 如果仅考虑变量TX而不考虑其他各因素的影响, 则本例可以画出根据数据得到的综合生存函数图(图2.6), 画该图使用了程序包survival[1], 代码如下:

```
library(survival)
fit <- survfit(Surv(TIME, as.numeric(STATUS)) ~ TX, data=u)
plot(fit,lty=1:2, ylab="S(t)",xlab="t",
```

[1]Terry Therneau and original Splus− >R port by Thomas Lumley (2011). survival: Survival analysis, including penalised likelihood. R package version 2.36-10. http://CRAN.R-project.org/package=survival.

```
    main='Survival Functions')
legend(1500,1,c("TX=1","TX=2"),lty=1:2)#图例
```

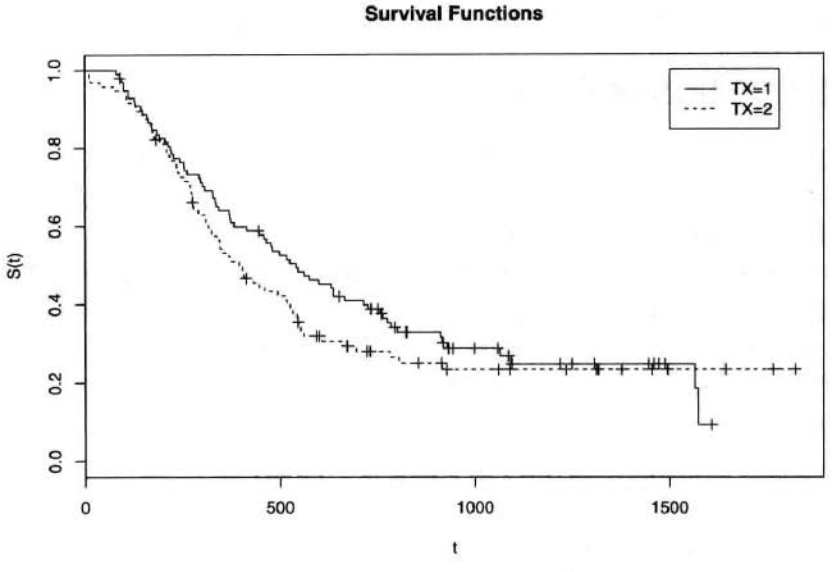


图 2.6 例2.2口咽癌数据的综合生存函数

生存函数并不是一成不变的, 它可能是一些自变量的函数. Cox比例危险回归模型把生存函数的某种变换形式描述成自变量的线性函数. 如果用$\boldsymbol{X}$代表自变量向量, 而用危险函数或其对数作为响应变量, 则Cox比例危险回归模型可写成

$$\ln h(t) = \ln h_0(t) + \boldsymbol{X}^\top \boldsymbol{\beta}$$

或者

$$h(t) = h_0(t) \exp(\boldsymbol{X}^\top \boldsymbol{\beta}),$$

式中, $h_0(t)$表示待估计的基本危险函数, 它与自变量$\boldsymbol{X}$无关.

现在用Cox比例危险回归模型来拟合口咽癌数据. 具体代码如下:

```
library(survival)
fit=coxph(Surv(TIME,as.numeric(STATUS))~.,data=u)#Cox回归模型
plot(survfit(fit)) #拟合的生存函数
summary(fit)#回归结果
```

输出结果表明R^2为0.248, 对模型的似然比检验的p值为2.383×10^{-5}, Wald检验的p值为1.197×10^{-5}, Score检验的p值为1.439×10^{-6}. 除了R^2稍小之外, 似乎拟合得还可以. 当然, 从输出结果来看, 许多变量并不显著. 因此可以减少一些变量, 这就留给读者去做了.

通常, 生存数据分析的主要目的是比较不同的处理方法以及各种因素(变量)对生存函数的影响, 而不是单纯地寻找拟合的模型.

下面我们用程序包mfp[1]来尝试关于Cox比例危险模型的多重分数多项式模型(multiple fractional polynomial model). 这种方法类似于多项式回归, 但更加灵活, 而且自动进行逐步回归及变量选择. 参见Ambler and Royston (2001). 代码为:

```
library(mfp)
f=mfp(Surv(TIME,as.numeric(STATUS))~fp(AGE, df=4,select=0.05)
    +INST+SEX+TX+GRADE+COND+SITE+T.STAGE+N.STAGE,
    family=cox,data=u)
print(f)#输出结果
(rsq=1-sum((f$residuals)^2)/sum((u$TIME-mean(u$TIME))^2))#R^2
```

输出的结果列在下面. 此外, 似然比检验的p值为1.327×10^{-5}, R^2为0.999, 比前面的拟合结果要好.

	coef	exp(coef)	se(coef)	z	p
COND2.1	1.18540	3.2720	0.2139	5.54227	2.99e-08
SEX2.1	-0.27923	0.7564	0.2207	-1.26519	2.06e-01
T.STAGE2.1	-0.31705	0.7283	0.5000	-0.63416	5.26e-01
T.STAGE3.1	-0.23767	0.7885	0.4718	-0.50373	6.14e-01
T.STAGE4.1	0.16458	1.1789	0.4955	0.33214	7.40e-01
N.STAGE1.1	-0.28642	0.7509	0.3134	-0.91404	3.61e-01
N.STAGE2.1	-0.04994	0.9513	0.3410	-0.14644	8.84e-01
N.STAGE3.1	0.22688	1.2547	0.2567	0.88399	3.77e-01
GRADE2.1	0.09141	1.0957	0.2253	0.40576	6.85e-01
GRADE3.1	-0.30871	0.7344	0.3064	-1.00764	3.14e-01
INST2.1	0.31219	1.3664	0.3456	0.90327	3.66e-01
INST3.1	-0.12545	0.8821	0.3615	-0.34706	7.29e-01
INST4.1	-0.22755	0.7965	0.3872	-0.58775	5.57e-01
INST5.1	-0.22744	0.7966	0.3861	-0.58911	5.56e-01

[1]Original by Gareth Ambler and modified by Axel Benner (2010). mfp: Multivariable Fractional Polynomials. R package version 1.4.9. http://CRAN.R-project.org/package=mfp.

```
INST6.1      0.03597    1.0366   0.3696  0.09731 9.22e-01
TX2.1        0.08314    1.0867   0.1829  0.45452 6.49e-01
SITE2.1     -0.13228    0.8761   0.2413 -0.54817 5.84e-01
SITE4.1     -0.02888    0.9715   0.2326 -0.12420 9.01e-01

Likelihood ratio test=54.9  on 18 df, p=1.327e-05 n= 193
```

2.4 数据出现多重共线性情况: 岭回归、lasso回归、适应性lasso回归、偏最小二乘回归

针对使得传统回归无法正常运行的多重共线性问题, 我们在这里介绍岭回归、lasso回归、适应性lasso回归、偏最小二乘回归, 并顺便介绍一点主成分回归.[1] 请读者注意, 对于共线性问题, 机器学习方法往往优于本节介绍的传统方法. 请看后面的有关内容.

在多元回归中, 当两个或更多的自变量有些相关的时候, 就有可能出现多重共线性(multicollinearity)的情况. 当多重共线性严重时, 模型或数据的微小变化有可能造成系数估计的较大变化, 这使得结果模型不稳定, 也不易解释. 一般来说, 只要不存在严重共线性, 对预测不会有较大影响. 但高度的多重共线性会造成计算困难, 比如矩阵的逆可能会很不稳定.

有一些关于多重共线性的度量, 其中之一是容忍度(tolerance)或(等价的)方差膨胀因子(variance inflation factor, VIF), 而另一个是条件数(condition number), 常用κ表示. 其中容忍度与VIF的定义为:

$$\text{tolerance} = 1 - R_j^2, \quad \text{VIF}_j = \frac{1}{1 - R_j^2},$$

式中, R_j^2是第j个变量在所有其他变量上回归时的确定系数. 容忍度太小(按照一些文献, 比如小于0.2或0.1)或VIF太大(比如大于5或10), 则有多重共线性问题. 而条件数的定义为:

$$\kappa = \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}},$$

式中, λ为$\boldsymbol{X}^\top \boldsymbol{X}$的特征值($\boldsymbol{X}$代表自变量矩阵). 显然, 当自变量矩阵正交时, 条件数κ为1. 一些研究者认为, 当$\kappa > 15$时, 有共线性问题; 而$\kappa > 30$时, 说明共线性问题严重. 当然, 这些判断准则可能不一致, 或者不很准确, 但不失为一种参考.

[1] 主成分回归的结果一般不优于偏最小二乘回归.

这里介绍几种常用的处理多重共线性的方法, 包括岭回归(ridge regression)、lasso回归、适应性lasso回归及偏最小二乘回归(partial least squares regression, PLSR)等, 附带介绍一下比较经典的主成分回归(principal component regression, PCR). 这几种方法通过一个例子来介绍.

例 2.3 糖尿病数据(diabetes.csv) 这个数据来自Efron et al. (2004), 包含在R程序包`lars`[1]中. 该数据除了因变量y之外, 还有两个自变量矩阵, x及x2, 前者是标准化的, 为442×10矩阵, 后者为442×64矩阵, 包括前者及一些交互作用. 该数据是关于糖尿病的血液等化验指标.

首先, 我们来看共线性问题, VIF可以通过R程序包`car`[2]的函数`vif()`得到, 条件数κ则可从R固有的函数`kappa()`得到. 下面在计算VIF时使用数据x2. 与这些计算有关的代码如下:

```
w=read.csv("diabetes.csv")
#w[,1:10]为x,w[,11]为y,而w[,12:75]为x2
kappa(w[,12:75])#x2的条件数
library(car)#包含vif的程序包
vif(lm(y~.,w[,11:75]))
```

计算结果表明, 数据x2的条件数$\kappa = 11427.09$, 而x2最大的5个VIF依次为1295001.21, 1000312.11, 180836.83, 139965.06, 61177.87. 看来共线性问题很严重. 我们就不尝试简单回归了.

下面分别介绍几种应对具有共线性问题数据的回归方法. 这些方法中大多都是参数模型方法, 但从模型的预测效果来说, 一般不如非参数的偏最小二乘方法.

2.4.1 岭回归

假定自变量数据矩阵$\boldsymbol{X} = \{x_{ij}\}$为$n \times p$的, 普通最小二乘回归(ordinary least squares, OLS) 寻求那些使得残差平方和最小的系数β, 即

$$(\hat{\alpha}^{(ols)}, \hat{\beta}^{(ols)}) = \arg\min_{(\alpha,\beta)} \sum_{i=1}^{n} \left(y_i - \alpha - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 .$$

[1]Trevor Hastie and Brad Efron (2011). lars: Least Angle Regression, Lasso and Forward Stagewise. R package version 0.9-8. http://CRAN.R-project.org/package=lars.

[2]John Fox and Sanford Weisberg (2011). *An R Companion to Applied Regression.* Second Edition. Thousand Oaks CA: Sage. URL: http://socserv.socsci.mcmaster.ca/jfox/Books/Companion.

岭回归则需要一个惩罚项来约束系数的大小, 其惩罚项就是在上面的公式中增加一项$\lambda\sum_{j=1}^{p}\beta_j^2$, 即岭回归的系数既要使得残差平方和小, 又不能使得系数太膨胀:

$$(\hat{\alpha}^{(ridge)},\hat{\beta}^{(ridge)})=\arg\min_{(\alpha,\beta)}\sum_{i=1}^{n}\left[\left(y_i-\alpha-\sum_{j=1}^{p}x_{ij}\beta_j\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2\right],$$

这等价于在约束条件$\sum_{j=1}^{p}\beta_j^2\leqslant s$下, 满足

$$(\hat{\alpha}^{(ridge)},\hat{\beta}^{(ridge)})=\arg\min_{(\alpha,\beta)}\sum_{i=1}^{n}\left(y_i-\alpha-\sum_{j=1}^{p}x_{ij}\beta_j\right)^2.$$

显然这里有确定λ或者s的问题, 一般都用交叉验证或Mallows C_p等准则通过计算来确定. 可以用程序包MASS中的函数lm.ridge()来实现. 这里采用更方便的可以自动选择岭回归参数的程序包ridge[1] 中的函数linearRidge(). 代码如下:

```
w=read.csv("diabetes.csv")
library(ridge)
a=linearRidge(y ~ ., data = w[,11:75])
summary(a)
```

计算结果包括估计的岭回归参数(5.363159)以及各个变量的系数. 由于输出很长, 这里就不显示了.

2.4.2 lasso回归

在原理上和岭回归的想法有些类似, 但惩罚项中不是系数的平方而是其绝对值, 即在约束条件$\sum_{j=1}^{p}|\beta_j|\leqslant s$下, 系数需要满足下面的条件:

$$(\hat{\alpha}^{(lasso)},\hat{\beta}^{(lasso)})=\arg\min_{(\alpha,\beta)}\sum_{i=1}^{n}\left(y_i-\alpha-\sum_{j=1}^{p}x_{ij}\beta_j\right)^2.$$

出于绝对值的特点, lasso回归不像岭回归那样把系数缩小, 而是筛选掉一些系数. 这里的计算主要使用R程序包lars中的函数lars(), 该程序包除了lasso方法之外, 还有最小角度回归具有针对共线性的功能, 请读者自己学习. 这个软件对于系数的选择有k折交叉验证(k-fold CV)及C_p两种方法. k折交叉验证是评价模型的一种常用方法, 它把所有的数据观测值大致分成k等份(可以

[1] Erika Cule (2012). ridge: Ridge Regression with automatic selection of the penalty parameter. R package version 2.1-1. http://CRAN.R-project.org/package=ridge.

随机分), 然后轮流以其中的所有可能的$k-1$份为训练集, 用来拟合数据, 剩下1份为测试集, 即测试集与训练集的观测值数目之比约为$1:(k-1)$, 一共计算k次, 得到拟合测试集时的均方误差(或其他指标, 如标准化均方误差)那样的k个指标, 再做平均. 对于每个模型都做一遍, 然后选择平均均方误差最小的模型. Mallows C_p 统计量也是用来评价回归的一个准则. 如果从k个自变量中选取p个$(k>p)$参与回归, 那么C_p统计量的定义为

$$C_p=\frac{SSE_p}{S^2}-n+2p;\ SSE_p=\sum_{i=1}^{n}(Y_i-Y_{pi})^2.$$

据此, 选取C_p最小的模型. 对于糖尿病数据, 计算代码如下(得到表2.7和图2.7):

```
library(lars)
#由于lars函数只用于矩阵型数据
#下面就把数据中的自变量和因变量变为矩阵形式
x=as.matrix(w[,1:10]);y=as.matrix(w[,11])
x2=as.matrix(w[,12:75]);laa=lars(x2,y)
plot(laa) #绘出图2.7
summary(laa)#给出Cp值(表2.7)
cva=cv.lars(x2,y,K=10) #10折交叉验证
best=cva$index[which.min(cva$cv)]#选适合值(随机性使得结果不同)
coef=coef.lars(laa,mode="fraction",s=best)#使CV最小步时的系数
min(laa$Cp)#哪个Cp最小, 结果是第15个
coef1=coef.lars(laa,mode="step",s=15)#使laa$Cp最小的step的系数
```

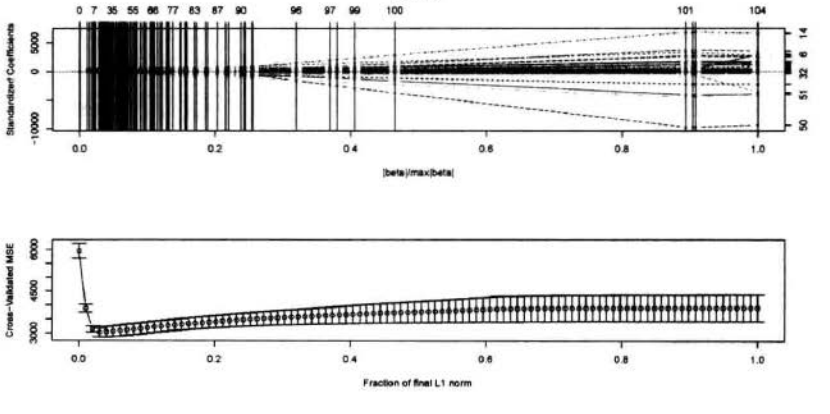


图 2.7 糖尿病数据在lasso回归中系数随参数的变化(上)以及CV的变化(下)

表2.7给出不同情况下C_p统计量的值(一共100多次尝试, 这里只给了13到17步的结果), 使其值最小的为第15步($C_p = 18.20$). 图2.7(上)给出了在不同的步数下系数增减的情况, 最左边是只有截距, 最右边是保持所有变量. 图2.7(下)给出了CV的变化图, 可以看出在什么时候达到极小值. 注意, 由于交叉验证的随机性等原因, 用CV和C_p所选择的结果可能会有所不同. 本例用CV选择了13个变量(用`coef[coef!=0]`查看), 而用C_p选择了14个变量(用`coef1[coef1!=0]`查看).

表 2.7 糖尿病数据在lasso回归中C_p值的变化

step	Df	Rss	Cp
13	14	1234993.28	21.86
14	15	1225552.04	20.53
15	16	1213288.85	18.20
16	17	1212253.39	19.83
17	18	1210148.58	21.09

2.4.3 适应性lasso回归

适应性lasso(adaptive lasso, alasso)回归是lasso回归的改进型. 与lasso回归和岭回归类似, 其系数β要满足下面条件:

$$(\hat{\alpha}^{(alasso)}, \hat{\beta}^{(alasso)}) = \arg\min_{(\alpha,\beta)} \sum_{i=1}^{n} \left(y_i - \alpha - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 .$$

但惩罚项是系数绝对值的加权平均, 即约束条件为$\sum_{j=1}^{p} w_i|\beta_j| \leqslant s$. 式中, $w_i = 1/(\hat{\beta}_i)^\gamma$. 而$\gamma > 0$为一个调整参数. 这实际上是Friedman (2008)的方法的特例, 适用于很大范围的损失函数及惩罚条件. 因此, 前面说过的岭回归和lasso回归仅仅是其方法的特例. 这里使用的是程序包**msgps**[1], 其中不仅包括适应性lasso(alasso)回归, 还包括弹性网络(elastic net)及广义弹性网络(generalized elastic net)等方法. 该程序包寻求最优参数是基于广义路径搜索方法(generalized path seeking algorithm). 而确定最优所根据的准则包括Mallows C_p、偏差纠正的AIC (AICc)、广义交叉验证(generalized cross validation, GCV)及BIC. 这里仅介绍alasso, 希望读者能够继续学习Friedman (2008)的其他方法.

对于糖尿病数据的例子, 计算代码为:

[1]Kei Hirose (2011). msgps: Degrees of freedom of elastic net, adaptive lasso and generalized elastic net. R package version 1.1. http://CRAN.R-project.org/package=msgps.

```
w=read.csv("diabetes.csv")
x=as.matrix(w[,1:10]);y=w[,11];x2=as.matrix(w[,12:75])
library(msgps)
#adaptive lasso
al=msgps(x2,y,penalty="alasso",gamma=1,lambda=0)
summary(al)
plot(al)
```

输出结果显示调整参数选的是$\gamma = 45.41$[1], 这时各个准则的值及自由度为:

```
ms.tuning:
        Cp  AICC   GCV   BIC
[1,] 7.987 7.987 8.433 5.111

ms.df:
        Cp  AICC   GCV   BIC
[1,] 18.68 18.68 19.47 13.27
```

同时输出的还有各个参数. 由于参数很多, 这里仅以图2.8表示在不同情况下参数的变化.

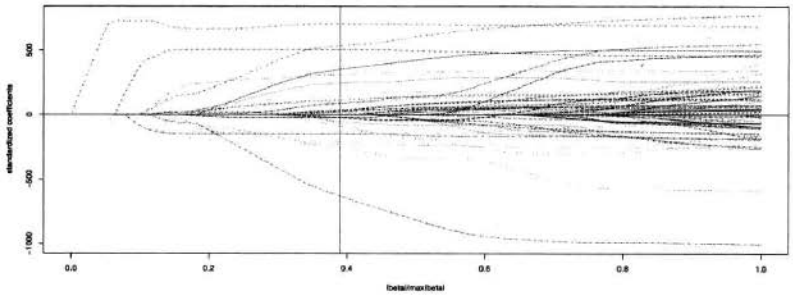


图 2.8 糖尿病数据在适应性lasso回归中系数随参数的变化

2.4.4 偏最小二乘回归

如果说lasso回归与岭回归在思想上有类似之处, 那么偏最小二乘回归完全是另类. 但偏最小二乘回归有些类似于主成分回归, 主成分回归是在自变量和因变量(如果也有多个变量) 中各自找到一些互相独立的主成分, 然后按照计算主成分时得到的特征值的大小(特征值较大的主成分对原来变量的代表性也较

[1]代码中的选项lambda大于0时, 相应于$\hat{\beta}_i$为岭回归参数估计.

强)来选取部分主成分, 主成分是独立的, 用这些主成分代替原来的变量进行回归, 共线性问题就解决了.

偏最小二乘回归则在因变量(如果也有多个变量组成)和自变量中先各自寻找一个因子(成分), 条件是这两个因子在其他可能的成分中最相关, 然后在选中的这一对因子的正交空间中再选一对最相关的因子, 如此下去, 直到这些对因子有充分代表性为止(可以用交叉验证).

对糖尿病数据使用程序包pls.[1] 该程序包也可做主成分回归. 因为一些研究发现(Wold et al., 1984, Garthwaite, 1994)偏最小二乘回归在预测上优于普通最小二乘回归及主成分回归, 在观测值数目相对较少(甚至少于变量数目)时, 偏最小二乘回归仍然可以使用. 所以这里仅做偏最小二乘回归的计算, 主成分回归的计算完全类似, 只需把下面的函数plsr()换成pcr()即可.

对糖尿病数据例子进行偏最小二乘回归并绘制偏最小二乘回归中CV的RMSEP变化图(图2.9)的计算代码如下:

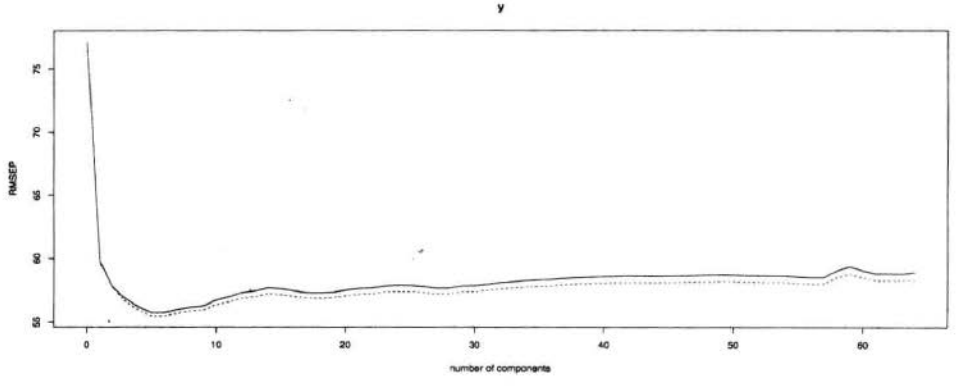


图 2.9 糖尿病数据在偏最小二乘回归中CV的RMSEP变化

```
library(pls)
ap=plsr(y ~ x2, 64, validation = "CV") #求出所有可能的64个因子
ap$loadings #看代表性, 前28个因子可以代表76.4%的方差
ap$coef #看各个因子作为原变量的线性组合的系数
validationplot(ap)#此图可用来根据CV所用准则最优的原则挑选因子数量
RMSEP(ap);MSEP(ap);R2(ap)#不同准则(MSEP,R2)在不同因子数量时的值
```

图2.9可以用来根据CV中的RMSEP最小的原则挑选因子数量. 可以看出, 5个因子时RMSEP最小. 用其他准则, 比如MSEP及R2都选择了5个因子.

[1] Bjørn-Helge Mevik, Ron Wehrens and Kristian Hovde Liland (2011). Partial Least Squares and Principal Component regression, GPL-2. URL: http://mevik.net/work/software/pls.html.

第三章　横截面数据回归: 机器学习方法

3.1　没有任何先验假定的数据: 机器学习回归方法

对于无法做任何分布假定的横截面数据, 本节将介绍算法模型, 即机器学习或数据挖掘中常用的方法, 包括决策树回归、boosting回归、bagging回归、随机森林回归、支持向量机回归、人工神经网络回归. 这些方法体现了统计的最新发展.

为了比较各种回归的预测效果, 介绍了交叉验证的方法, 也讨论了各种方法的稳定性及过拟合现象.

前面介绍的统计模型大多对数据有一定的要求或假定, 而且模型本身也可以有较明确的数学形式. 关于模型或拟合优劣, 大多根据对数据的分布假定得到的检验来判断. 但是, 在大多数情况下, 人们根本无法对真实世界数据的分布做任何假定, 同时, 很难想象复杂的现实世界能够用有限的数学公式来描述. 后面几节要介绍的机器学习方法(即数据挖掘方法或算法模型)对数据没有任何假定, 而产生的结果也用交叉验证的方法来判断, 脱离了假定分布⇒用明确数学模型来拟合⇒假设检验⇒p值的经典统计过程. 这种基于算法或程序的模型预测效果相当好, 而且交叉验证的结果也容易被广大实际工作者理解和接受. 下面的例子贯穿此小节的始终.

例 3.1　mg数据(mg.csv)　这个数据来自Flake and Lawrence(2002), 可从网站下载.[1] 只知道该数据有6个自变量和1个因变量, 一共有1385个观测值, 没有关于该数据的任何其他信息. 之所以采用这个数据是因为这里所有变量都是数量变量, 因此, 也可以用经典线性回归来做, 从而可以对各种方法进行比较.

首先试着对全部数据做一个简单的线性回归. 发现结果不太理想($R^2 = 0.5868$). 计算各个变量之间的相关系数, 可得到表3.1, 所用代码如下:

```
w=read.csv("mg.csv")#读入数据
a=lm(y~.,w)#简单线性回归
cor(w)#相关系数表
```

从相关系数和两两散点图(该散点图可用代码`pairs(w)`或者`plot(w)`得到, 这里不显示)看不出各个变量之间的任何模式. 通过这个数据, 介绍几种算法建模方法, 同时用交叉验证(利用均方误差或标准化均方误差)对各种方法进行比较.

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression/mg.

表 3.1 mg数据各个变量之间的相关系数

	y	x1	x2	x3	x4	x5	x6
y	1.00	0.26	0.54	−0.50	−0.44	0.55	0.15
x1	0.26	1.00	−0.15	−0.65	0.43	0.45	−0.63
x2	0.54	−0.15	1.00	−0.15	−0.65	0.44	0.45
x3	−0.50	−0.65	−0.15	1.00	−0.15	−0.65	0.44
x4	−0.44	0.43	−0.65	−0.15	1.00	−0.15	−0.65
x5	0.55	0.45	0.44	−0.65	−0.15	1.00	−0.15
x6	0.15	−0.63	0.45	0.44	−0.65	−0.15	1.00

下面我们对各种方法都用10折交叉验证的方法来判断其结果的可靠性. 对于每种方法, 按照随机建立的10个训练集通过计算建立10个模型, 对测试集分别得到10个标准化均方误差(NMSE), 再得出10次平均的NMSE. 令$\bar{y}$为因变量均值, $\hat{y}$为从训练集得到的模型对一个测试数据集的预测值, 这里NMSE的定义为

$$\text{NMSE}=\overline{(y-\hat{y})^2}\Big/\overline{(y-\bar{y})^2}=\frac{\sum(y-\hat{y})^2}{\sum(y-\bar{y})^2}.$$

显然, 如果什么模型都不用, 仅仅用均值来做预测, 那么NMSE应该等于1. 所以, 如果在回归时得到NMSE大于1, 这个模型就很糟糕了, 还不如没有模型. 仅仅对于训练集来说, 其NMSE等于$1-R^2$, 这里R^2为回归的确定系数. 但是对于测试集来说, 其NMSE与测试集回归的R^2没有什么关系. 交叉验证主要关心测试集的NMSE.

下面构造为交叉验证服务的10个训练集和测试集, 办法是: 随机把下标分配给$1,2,...,10$十个数字, 也就是把数据下标随机分成10份, 然后每次提取一份作为测试集, 其他9份放在一起作为训练集, 用模型进行拟合, 记下结果和误差, 如此下去, 一共做10次, 最后把误差平均起来. 下面一直到本章结尾对各种方法都用这套训练集和测试集. 这里所用的随机选Z折下标集的函数为:

```
CV=function(n,Z=10,seed=888){
z=rep(1:Z,ceiling(n/Z))[1:n];set.seed(seed);z=sample(z,n)
mm=list();for (i in 1:Z) mm[[i]]=(1:n)[z==i];return(mm)}
```

这里变元n为样本量, Z为折的数目(这里缺省值为10), seed为随机种子(这里缺省值为888). 输出为Z个下标集; mm[[i]]为第i $(i=1,2,...,Z)$个下标集. 根据这个函数对这个mg数据求10个下标集的代码为:

```
w=read.csv("mg.csv");n=nrow(w);Z=10;mm=CV(n,Z);D=1
```

这里mm存储了10个下标集, 而D说明因变量是第一个变量, 这些值在后面对于每种方法都要用.

为了和其他方法进行比较, 先对数据mg求简单线性回归的10折交叉验证的测试集的NMSE, 代码如下:

```
MSE=rep(0,Z) #建立一个向量储存结果
for(i in 1:Z)#对每一组训练集和测试集做1次, 共Z=10次
{m=mm[[i]];M=mean((w[m,D]-mean(w[m,D]))^2)
a=lm(y~.,w[-m,])#简单线性回归, 这里[-m]为训练集下标集合
MSE[i]=mean((w[m,D]-predict(a,w[m,]))^2)/M}#求测试集NMSE
mean(MSE) #测试集的NMSE
```

得到测试集的NMSE为0.4193126.

3.2 决策树回归(回归树)

3.2.1 拟合全部数据

决策树(decision tree)用于回归时称为回归树(regression tree), 用于分类时则称为分类树(classification tree). 先用mg的全部数据做一棵回归树, 然后解释其意义. 我们所用的是程序包rpart[1]中的函数rpart(), 执行下面的代码(这里的rpart.plot包[2]是基于rpart的画图程序包, 打开rpart.plot程序包就自动打开rpart程序包了):

```
library(rpart.plot)#同时自动打开rpart
a=rpart(y~.,w);a #计算决策树并输出决策树的细节
rpart.plot(a,type=2,faclen=T) #画出决策树的图
```

得到如下输出, 这个输出给出了决策树的细节.

```
n= 1385

node), split, n, deviance, yval
      * denotes terminal node

 1) root 1385 70.9545100 0.9301440
```

[1]Terry Therneau and Beth Atkinson. R port by Brian Ripley (2011). rpart: Recursive Partitioning. R package version 3.1-50. http://CRAN.R-project.org/package=rpart.

[2]Stephen Milborrow (2011). rpart.plot: Plot rpart models. An enhanced version of plot.rpart. R package version 1.2-2. http://CRAN.R-project.org/package=rpart.plot.

```
2) x2< 0.9894983 736 31.9699900 0.8017888
  4) x3>=0.8423859 551 16.4573000 0.7288982
    8) x2>=0.5178963 483 13.3868900 0.7068909
     16) x2< 0.7999875 263  4.7064590 0.6422662
       32) x3< 1.092533 185  3.3567710 0.6073434 *
       33) x3>=1.092533 78  0.5889238 0.7250958 *
     17) x2>=0.7999875 220  6.2689860 0.7841468 *
    9) x2< 0.5178963 68  1.1749050 0.8852150 *
  5) x3< 0.8423859 185  3.8660570 1.0188850
   10) x4>=1.062353 71  1.8511830 0.9239636 *
   11) x4< 1.062353 114  0.9767496 1.0780020 *
3) x2>=0.9894983 649 13.1077400 1.0757060
  6) x3>=1.056185 206  3.5797200 0.9644841
   12) x4>=0.9107852 23  0.5746155 0.7823564 *
   13) x4< 0.9107852 183  2.1462980 0.9873744 *
  7) x3< 1.056185 443  5.7947760 1.1274250 *
```

有了上面的打印输出以及与其等价的图3.1, 就可以了解决策树的构造及解释了. 决策树就像一棵倒长的树, 有很多分叉, 分叉点叫节点(node). 打印输出的1号节点为根节点(图中最高的一个), 这里的信息是全部数据的信息, 比如$n = 1385$, 偏差[1](deviance)=70.9545100, 在该节点的因变量的均值=0.9301440. 这些信息部分显示在图3.1中. 然后, 计算机程序在每个自变量中选择一个数值分割点, 使得在该节点的整个数据在这个分割点比在其他分割点分成两部分之后的偏差之和都小, 对每个自变量都选择这样一个分割点, 并对不同自变量的分割结果做比较, 得到一个使得分割后总偏差最小的变量, 这个变量就是该节点的分割变量(或称为拆分变量).

根据打印输出, 本例的根节点的分割变量为x2, 其分割点为0.9894983, x2小于该值的观测值分到左边(节点2), 大于该值的则分到右边(节点3). 在每个节点都展示出观测值数目、偏差及因变量均值这三项内容(图中仅有均值), 这样数据就分成了两部分. 然后, 对每一部分数据, 选择分割变量及其分割点的程序重新开始. 如此下去, 根据一些按照拟合程度和避免决策树无限制长下去的准则, 到一定节点就不再分割了, 这些节点就是终节点(亦称叶节点), 在打印输出中有星号(*)的为叶节点.

[1]这里在每个节点的偏差定义为在该节点的数据中的因变量与其均值之差的平方和, 即$\sum_i (y_i - \bar{y})^2$. 这里回归树用偏差大小来选择变量, 也可以用其他诸如残差平方和等作为选择变量的准则.

使用决策树很简单, 如果有新的(只知道自变量的)观测值, 就可以根据决策树以及各个变量的大小, 从根节点一直走到一个终节点, 那里因变量的均值就是该观测值的因变量预测值. 决策树本身受数据影响较大, 但许多决策树结合起来就可以构成很优秀的方法, 比如后面介绍的boosting、bagging及随机森林等组合方法就是以决策树作为基本决策单元构造出来的.[1] 另外, 决策树使用简单, 容易解释, 完全不用考虑诸如线性性、共线性之类的约束或干扰, 很受实际工作者的青睐.

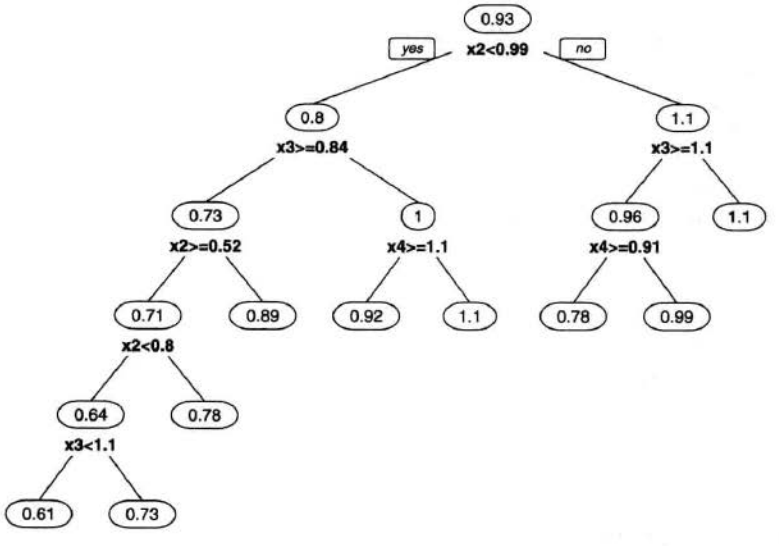


图 3.1 关于mg数据的回归树

3.2.2 交叉验证

下面对mg数据通过10个训练集构建10棵决策树, 每次由相应的训练集求误差, 最终得到10折交叉验证的测试集的NMSE:

```
MSE=rep(0,Z);for(i in 1:Z)
{m=mm[[i]];M=mean((w[m,D]-mean(w[m,D]))^2)
a=rpart(y~.,w[-m,]) #决策树回归
MSE[i]=mean((w[m,D]-predict(a,w[m,]))^2)/M}
```

[1]组合方法的基本决策单元也可以是决策树之外的其他模型.

```
mean(MSE)
```

得到测试集的NMSE为0.3624986. 这比简单线性回归的结果有所改进. 但前面说过, 决策树并不十分稳定(即可能受到数据变化的影响), 属于一种弱学习器, 而下面要介绍的boosting、bagging及随机森林属于由弱学习器组合成的较强的学习器.

3.3 boosting回归

有各种各样的boosting方法, 它们都是组合方法, 其共同特点就是把弱学习器组合起来形成强学习器. 首先, 为什么要组合? 假定一个弱学习器犯错误的概率是0.49, 即每次决策都有49%的可能出错, 但是如果用1001个出错概率相同的独立弱学习器来做等权重投票, 那么, 利用简单的二项分布可以算出, 按照服从简单多数的原则, 犯错误的概率就减少到0.26(计算代码`1-pbinom(500,1001,0.49)`). 如果用10001个弱学习器, 则出错概率减少到0.02(计算代码`1-pbinom(5000,10001,0.49)`). 当然, 决策树的出错概率不会这样固定不变, 但只要平均少于50%, 组合起来就会很有利了.

组合会增强学习能力. 但是, 一个数据只能产生一个学习器, 比如决策树, 怎样才能产生那么多学习器呢? 这就要靠再抽样, 比如自助法(bootstrap)抽样, 它在数据中反复**放回地**抽取样本量大小相同的样本, 这样就产生了许多很难相同的自助法样本, 由此得到许多弱学习器. boosting方法在每次再抽样时还会自动调整, 比如, 在前一次抽样时, 某些样本点在产生弱学习器的决策时会有较大的误差, 这可能是由它们在样本中缺乏代表性所致, 于是在第二次抽样时就增加其被抽中的概率以改进结果, 在其产生的诸学习器投票(或平均)时, 也不一定是等权重投票, 可根据其错误率或误差大小来加权投票(或加权平均), 也可以用其他方法来组合这些弱学习器. 这就是有如此多不同的boosting方法的原因.

这里处理mg数据时使用基于模型的boosting(model-based boosting)方法, 程序包为**`mboost`**.[1] 如果把需要的回归模型记为$f(x)$, 这里仅考虑可加模型$f(x)=\sum_{j=1}^{p} f_j(x^{(j)})$, 比如$f_j(x^{(j)})$可以是由第$j$组自变量$x^{(j)}$及某基本学习器$f_j$的结合. 如果定义了损失函数$\rho(\cdot,\cdot)$, 则模型需要按照下面公式拟合:

$$(\hat{f}_1,\hat{f}_2,...,\hat{f}_p)=\underset{(f_1,f_2,...,f_p)}{\arg\min}\sum_{i=1}^{n} w_i\rho\left(y_i,\sum_{j=1}^{p} f_j(x^{(j)})\right),$$

[1] T. Hothorn, P. Buehlmann, T. Kneib, M. Schmid, and B. Hofner (2011). mboost: Model-Based Boosting. R package version 2.1-1, http://CRAN.R-project.org/package=mboost.

式中, n为训练样本量; w_i为某种权重; $x^{(j)}$为自变量观测值; y_i为相应的因变量观测值. 其方法的细节请参阅Buehlmann and Hothorn(2007). 按照我们的数据, $p=6$, 对每个变量的回归模型$f_j(x^{(j)})\,(j=1,2,...,p)$在下面的程序中都选为决策树, 用btree()表示.

对数据mg做基于模型的boosting回归, 直接求10折交叉验证测试集的NMSE, 代码如下:

```
library(mboost)
MSE=rep(0,Z)
set.seed(1010);for(i in 1:Z)
{m=mm[[i]];M=mean((w[m,D]-mean(w[m,D]))^2)
a=mboost(y~btree(x1)+btree(x2)+btree(x3)+btree(x4)+
    btree(x5)+btree(x6),data =w[-m,])
MSE[i]=mean((w[m,D]-predict(a,w[m,]))^2)/M}
mean(MSE)
```

得到测试集的NMSE为0.3407539. 这个结果比单个决策树又有改进.

3.4 bagging回归

bagging是一种比boosting简单的组合方法. 在bagging中, 就是不断放回地对训练样本进行再抽样(自助法抽样), 每次再抽样的样本量和原来样本量一样. 对每个自助法样本, 都建立一棵回归树, 最终, 对于任何一个观测值, 每棵树都给出一个预测值, 最终的预测值为这些值的简单平均. 这里用的是程序包ipred [1]中的函数bagging().

对数据mg做bagging回归, 直接求10折交叉验证测试集的NMSE, 代码如下:

```
library(ipred)
MSE=rep(0,Z)
set.seed(1010);for(i in 1:Z)
{m=mm[[i]];M=mean((w[m,D]-mean(w[m,D]))^2)
a=bagging(y~.,data =w[-m,])
MSE[i]=mean((w[m,D]-predict(a,w[m,]))^2)/M}
mean(MSE)
```

得到测试集的NMSE为0.3164777. 这个结果比boosting稍有改进.

[1] Andrea Peters and Torsten Hothorn (2012). ipred: Improved Predictors. R package version 0.8-13. http://CRAN.R-project.org/package=ipred.

3.5 随机森林回归

随机森林(random forest)是另一种组合方法, 也是由随机放回地再抽样的样本形成的决策树组成的, 其特点是这些决策树的每一节点的分割变量不是由所有的自变量竞争产生, 而是由随机选取的少数变量产生, 因此不仅产生每棵决策树的样本是随机的, 每棵树的每个节点的产生也是随机的. 这些随机产生的决策树数目很大, 因此称为随机森林. 结果的投票(或平均)是等权的. 这里用的是程序包**randomForest**.[1] 其细节可参看Breiman (2001).

对数据mg做随机森林回归、计算10折交叉验证测试集NMSE的代码为:

```
library(randomForest); MSE=rep(0,Z)
set.seed(1010);for(i in 1:Z)
{m=mm[[i]];M=mean((w[m,D]-mean(w[m,D]))^2)
a=randomForest(y~.,data=w[-m,])
MSE[i]=mean((w[m,D]-predict(a,w[m,]))^2)/M }
mean(MSE)
```

得到测试集的NMSE为0.2789902. 这个结果比前面方法又有改进. 注意, 对于不同的随机种子, 运行所得到的随机森林的结果也不同. 随机森林不会有过拟合现象. 随机森林还可以输出自变量重要性的度量. 对mg全部数据做随机森林回归,利用下面的代码可得到重要性度量(见表3.2).

```
library(randomForest);set.seed(1010)
SS=randomForest(y ~ ., data=w, importance=TRUE,proximity=TRUE)
SS$importance
```

表 3.2 随机森林对各个变量的重要性度量

	%IncMSE	IncNodePurity
x1	0.0106	7.0304
x2	0.0233	17.8357
x3	0.0198	13.3297
x4	0.0196	11.6248
x5	0.0147	13.9375
x6	0.0068	5.2385

表3.2的第一列数字是从替换该变量而致精确度平均递减的角度来衡量变量重要性, 第二列数字则是从以该变量为拆分变量所造成的均方误差的平均递

[1] A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* 2(3), 18–22.

减的角度来衡量变量重要性, 两列都是数字越大, 说明变量越重要.

3.6 支持向量机回归

支持向量机(support vector machine, SVM)是一种分类方法, 本书后面将要用到, 而由SVM发展出来的回归方法称为支持向量回归(support vector regression, SVR). 由于支持向量机方法发展很快, 因此有各种不同的后续方法, 但这里仅仅介绍支持向量机分类和回归的基本原理.

3.6.1 SVM

假定其目的是把空间中的两类点($y=-1$或$y=1$)用超平面$\boldsymbol{w}^\top\boldsymbol{x}+b=0$分开(在严格线性可分的情况下, 存在这样的超平面), 而且希望这个超平面与两类点的距离最大, 也就是说, 使得隔离带宽$\rho=2/\|\boldsymbol{w}\|$最大. 这等价于用Lagrange乘子法求下式的极小值.

$$L(\boldsymbol{w},b,\boldsymbol{\alpha})=\frac{1}{2}\|\boldsymbol{w}\|^2-\sum_{i=1}^{n}\alpha_i[y_i(\boldsymbol{w}^\top\boldsymbol{x}_i+b)-1].$$

根据得到的解$\boldsymbol{w}^*,b^*,\boldsymbol{\alpha}^*$得到最优分割超平面方程$\boldsymbol{w}^{*\top}\boldsymbol{x}+b^*=0$. 任意点$(\boldsymbol{x})$的函数值$\boldsymbol{w}^{*\top}\boldsymbol{x}+b^*$的符号确定了该点的分类, 或者说判别函数为$\mathrm{sgn}(\boldsymbol{w}^{*\top}\boldsymbol{x}+b^*)$.

上面介绍的是严格线性可分的情况. 如果允许一些错误存在, 则称为近似线性可分问题, 结果与此有同样的形式. 可以注意到, 这些结果有下面依赖于内积的对偶(duel)性质: 首先, 这里的建模过程仅仅依赖于训练集点对的内积; 其次, 判别过程仅仅依赖于未知点和训练集中支持向量的内积. 这种依赖于内积的独特性质使得我们能够解决线性不可分的问题. 对于线性不可分问题, 可能需要一些变换$\boldsymbol{x}\mapsto\boldsymbol{\Phi}(\boldsymbol{x})$, 这些变换是很难猜到的, 但基于对偶性质, 可以猜测较灵活的核函数$K(\boldsymbol{x}_i^\top,\boldsymbol{x}_j)=\boldsymbol{\Phi}(\boldsymbol{x}_i)^\top\boldsymbol{\Phi}(\boldsymbol{x}_j)$而非$\boldsymbol{\Phi}(\cdot)$本身.

3.6.2 SVR

在回归问题中, y不仅仅是-1和1. 令$f(\boldsymbol{x})=\boldsymbol{w}^\top\boldsymbol{x}+b$, 希望$y$与$f(\boldsymbol{x})$的离差越小越好, 问题还是归结于求使得$\|\boldsymbol{w}\|^2/2=\boldsymbol{w}^\top\boldsymbol{w}/2$最小的$\boldsymbol{w}$, 但约束条件是$|y_i-f(\boldsymbol{x}_i)|\leqslant\epsilon$, 这里$\epsilon$为某目标值. 类似于SVM, 允许一些误差, 这样就可以把上面的约束放宽为(对于大于0的ξ_i,ξ_i^*) $y_i-f(\boldsymbol{x}_i)\leqslant\epsilon+\xi_i$及$f(\boldsymbol{x}_i)-y_i\leqslant\epsilon+\xi_i^*$, 即得到Lagrange函数

$$L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\eta})=\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_i(\xi_i+\xi_i^*)-\sum_i(\eta_i\xi_i+\eta_i^*\xi_i^*)$$

$$-\sum_i \alpha_i(\epsilon+\xi_i-y_i+\boldsymbol{w}^\top\boldsymbol{x}_i+b)$$
$$-\sum_i \alpha_i^*(\epsilon+\xi_i^*+y_i-\boldsymbol{w}^\top\boldsymbol{x}_i-b).$$

需要在约束条件$\boldsymbol{\alpha},\boldsymbol{\eta}>\mathbf{0}$下, 解$\min_{\boldsymbol{w},b,\boldsymbol{\xi}}\{\max_{\boldsymbol{\alpha},\boldsymbol{\eta}} L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\eta})\}$问题. 对于非线性问题, 和前面SVM一样, 利用核函数来解决.

3.6.3 交叉验证

对数据mg, 使用程序包rminer[1]来实现SVR, 并求10折交叉验证测试集的NMSE, 代码如下:

```
library(rminer);set.seed(1010)
MSE=rep(0,Z);for(i in 1:Z)
{m=mm[[i]];M=mean((w[m,D]-mean(w[m,D]))^2)
a=fit(y~.,w[-m,],model="svm")
MSE[i]=mean((w[m,D]-predict(a,w[m,]))^2)/M}
mean(MSE)
```

得到测试集的NMSE为0.2937321.

还有其他一些程序包可以实现SVR, 一个是e1071.[2] 求mg数据的10折交叉验证测试集NMSE的代码如下:

```
library(e1071);set.seed(1010)
MSE=rep(0,Z);for(i in 1:Z)
{m=mm[[i]];M=mean((w[m,D]-mean(w[m,D]))^2)
a=svm(y~.,w[-m,])
MSE[i]=mean((w[m,D]-predict(a,w[m,]))^2)/M }
mean(MSE)
```

得到测试集的NMSE为0.295695.

[1]Paulo Cortez (2011). rminer: Simpler use of data mining methods (e.g. NN and SVM) in classification and regression. R package version 1.1. http://CRAN.R-project.org/package=rminer.

[2]David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2014). e1071: Misc Functions of the Department of Statistics (e1071), TU Wien. R package version 1.6-4. http://CRAN.R-project.org/package=e1071.

另一个可以实现SVR的程序包是**kernlab**.[1] 求mg数据的10折交叉验证测试集NMSE的代码如下:

```
library(kernlab);set.seed(1010)
MSE=rep(0,Z);for(i in 1:Z)
{m=mm[[i]];M=mean((w[m,D]-mean(w[m,D]))^2)
a=ksvm(y~.,w[-m,],model="svm")
MSE[i]=mean((w[m,D]-predict(a,w[m,]))^2)/M }
mean(MSE)
```

得到测试集的NMSE为0.2937321.

3.7 人工神经网络回归

人工神经网络(artificial neural networks)是对自然神经网络的模仿, 它可以有效地解决很复杂的有大量互相相关变量的回归和分类问题. 在训练神经网络时可能很费时间, 但一旦训练完毕, 代入新数据进行计算时则很快. 图3.2就是一个有6个自变量(输入)和1个因变量(输出)的神经网络的示意图. 左边代表自变量的6个节点形成输入层(input layer), 中间8个节点形成隐藏层(hidden layer), 最右边的一个节点属于输出层(output layer), 代表因变量. 这些节点按照箭头连接. 箭头旁的数字标出权重.

神经网络的因变量可以有多个, 隐藏层也可以有多个, 但一般一个隐藏层就够了. 隐藏层的节点可多可少, 太多可能导致过拟合, 太少则可能拟合不好, 一般由交叉验证来选择隐藏层的节点数目. 神经网络的原理是把上层节点的值加权平均送到下层节点, 最终到输出层节点, 然后根据误差大小反馈回前面的层, 再重新加权平均, 如此反复训练, 直到误差在允许范围之内. 下面的公式可以说明一般神经网络的加权过程.

$$y_j = f^*\left(\sum_k w_{kj} z_{jk} + w_{0j}\right) = f^*\left\{\sum_k w_{kj}\left[f\left(\sum_i w_{ik} x_i + w_{0k}\right)\right] + w_{0j}\right\},$$

式中, w_{ik}是自变量x_i在隐藏层第k个节点的权重; w_{kj}是隐藏层第k个节点对于第j个因变量的权重; z_k是在隐藏层第k个节点的值. 这里的f和f^*为激活函数, 通常定义为S形的logistic函数

$$f(x) = \frac{1}{1+\mathrm{e}^{-x}}.$$

[1] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, Achim Zeileis (2004). kernlab - An S4 Package for Kernel Methods in R. *Journal of Statistical Software* 11(9), 1-20. URL http://www.jstatsoft.org/v11/i09/.

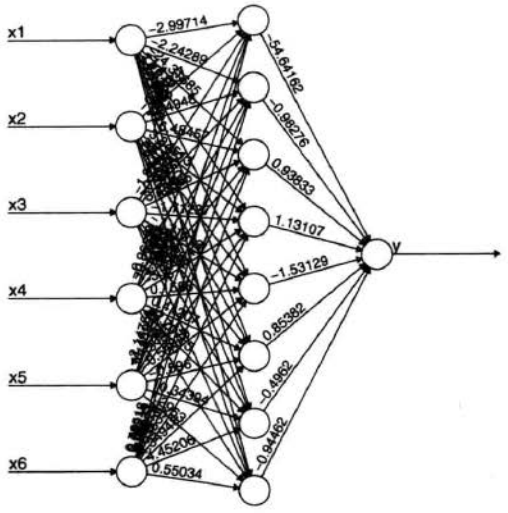


图 3.2 具有1个因变量、6个自变量及8个隐藏层节点的神经网络

3.7.1 确定合适的隐藏层节点数目

我们利用程序包nnet[1]来对数据mg做神经网络回归. 为了寻找合适的隐藏层节点数目及权重衰减参数decay, 我们使用程序包caret的函数train()来做格子点搜索, 下面是代码:

```
require(nnet);require(caret);set.seed(1010)
mygrid=expand.grid(.decay=c(0.5, 0.1), .size=c(4,5,6))
nnetfit=train(y/max(w[,D])~.,data=w, method="nnet",
maxit=1000, tuneGrid=mygrid, trace=F)
print(nnetfit)
```

部分输出为:

```
Resampling results across tuning parameters:

  decay  size  RMSE       Rsquared   RMSE SD      Rsquared SD
  0.1    4     0.1100032  0.5977453  0.002911433  0.02292053
  0.1    5     0.1100067  0.5977069  0.002829709  0.02250380
```

[1]Venables, W. N. & Ripley, B. D. (2002) *Modern Applied Statistics with S*. Fourth Edition. Springer, New York. ISBN 0-387-95457-0.

```
  0.1     6      0.1103432  0.5952075  0.002855127  0.02155031
  0.5     4      0.1156483  0.5795990  0.003321476  0.01872365
  0.5     5      0.1155982  0.5794548  0.003319754  0.01873370
  0.5     6      0.1155841  0.5793906  0.003319525  0.01873787

RMSE was used to select the optimal model using  the smallest
value. The final values used for the model were size = 4 and
decay = 0.1.
```

输出说明隐藏层节点数目为4, decay选择0.1. 人们可能注意到, 这里的因变量通过除以其最大值而被标准化了(y/max(w[,D]). 这是因为神经网络程序一般处理在[0, 1]区间的数据, 而在最后预测结果也要把结果还原(乘上最大值).

3.7.2 交叉验证

可以利用上面选择的参数来求10折交叉验证测试集的NMSE, 代码如下:

```
MSE=rep(0,Z);set.seed(1010);for(i in 1:Z)
{m=mm[[i]];M=mean((w[m,D]-mean(w[m,D]))^2)
a=nnet(y/max(w[,D])~ ., data=w[-m,], size=4, decay=0.1)
MSE[i]=mean((w[m,D]-predict(a,w[m,])*max(w[,D]))^2)/M}
mean(MSE)
```

得到测试集的NMSE为0.4142710.

其他程序包也可以做神经网络回归, 比如neuralnet.[1] 实施的代码如下:

```
#由于neuralnet函数中的公式不能有运算符, 因此把w转换成v
ny=(1:ncol(w))[-D]
nn1=paste(names(w)[D],"~",names(w)[ny[1]],sep="")
for(i in (1:ncol(w))[-D][-1])nn1=paste(nn1,"+",
    names(w)[i],sep="");v=w;v[,D]=v[,D]/max(w[,D])
#上面的nn1为对数据v的公式y ~ x1 + x2 + x3 + x4 + x5 + x6
#实施Z=10折交叉验证
MSE=rep(0,Z);set.seed(1010);for(i in 1:Z)
{m=mm[[i]];M=mean((w[m,D]-mean(w[m,D]))^2)
a=neuralnet(nn1,data=v[-m,],err.fct="sse",hidden=4,
```

[1]Stefan Fritsch, Frauke Guenther and following earlier work by Marc Suling (2012). neuralnet: Training of neural networks. R package version 1.32. http://CRAN. R-project.org/package=neuralnet.

```
    linear.output=FALSE)
MSE[i]=mean((w[m,D]-
    compute(a,v[m,-D])$net.result*max(w[,D]))^2)/M}
mean(MSE)
```

得到测试集的NMSE为0.3032879715.

3.8 10折交叉验证结果汇总及方法稳定性讨论

根据上述模型用决策树、boosting、bagging、随机森林、三种SVR方法、两种神经网络方法以及传统的线性模型来拟合mg数据的10折交叉验证, 得到的测试集的平均NMSE展示在表3.3中. 显然, 就这个数据而言, 随机森林表现最佳, 最差的是传统的线性模型. 注意, 笔者在使用这些方法时, 并未刻意在函数的各种选项中挑选最优的(除了神经网络), 用的都是默认值, 因此, 如果改变一些选项, 很可能会使得结果有所改进. 当然, 由于交叉验证的各个数据子集的选择有随机性, 改变随机种子会使得结果不同, 但差别不会太大.

表 3.3 各种方法关于训练集和测试集的NMSE

方法	平均NMSE
决策树	0.3625
boosting	0.3408
bagging	0.3165
随机森林	0.2790
SVR(rminer)	0.2937
SVR(e1071)	0.2957
SVR(kernlab)	0.2937
net(nnet)	0.4143
net(neuralnet)	0.3033
线性模型	0.4193

对于mg数据, 下面用不同数目(1到100棵)树的随机森林及不同数目(1到40个)隐藏层节点的神经网络来拟合, 绘出训练集和测试集的NMSE图(见图3.3), 以研究NMSE的走向. 所用代码如下(注意, 这里测试集和训练集各随机取一半观测值; 神经网络用的是neuralnet包, 而且利用了前面代码定义的v, nn1等结果):

```
w=read.csv("mg.csv")
m=sample(1:1385,ceiling(1385/2))#随机抽取的测试集下标
library(randomForest);n=100; NMSE=rep(0,n)->NMSE0
```

```
set.seed(1010);for (i in 1:n){
A=randomForest(y ~ ., data=w[-m,],  ntree=i)
y0=predict(A,w[-m,]);y1=predict(A,w[m,])
NMSE0[i]=mean((w$y[-m]-y0)^2)/mean((w$y[-m]-mean(w$y[-m]))^2)
NMSE[i]=mean((w$y[m]-y1)^2)/mean((w$y[m]-mean(w$y[m]))^2)}
library(neuralnet);set.seed(1010);n1=40;nmse=rep(0,n1)->nmse0
for(i in 1:n1){
a=neuralnet(nn1,data=v[-m,],err.fct="sse",hidden=i,
linear.output=FALSE)
y0=compute(a,v[-m,-D])$net.result*max(w[,D])
y1=compute(a,v[m,-D])$net.result*max(w[,D])
nmse0[i]=mean((w$y[-m]-y0)^2)/mean((w$y[-m]-mean(w$y[-m]))^2)
nmse[i]=mean((w$y[m]-y1)^2)/mean((w$y[m]-mean(w$y[m]))^2)}
par(mfrow=c(1,2))
plot(1:n,NMSE,type="l",ylim=c(min(NMSE,NMSE0),
  max(NMSE,NMSE0)),xlab="Number of trees",ylab="NMSE",
  main="Random Forests: NMSE",lty=2);lines(1:n,NMSE0)
legend("topright", lty=1:2,c("Training Set","Testing Set"))
plot(1:n1,nmse,type="l",ylim=c(min(nmse,nmse0),
  max(nmse,nmse0)),xlab="Number of units in the hidden layer",
  ylab="NMSE",main="NNET: NMSE",lty=2);lines(1:n1,nmse0)
legend("topright", lty=1:2,c("Training Set","Testing Set"))
```

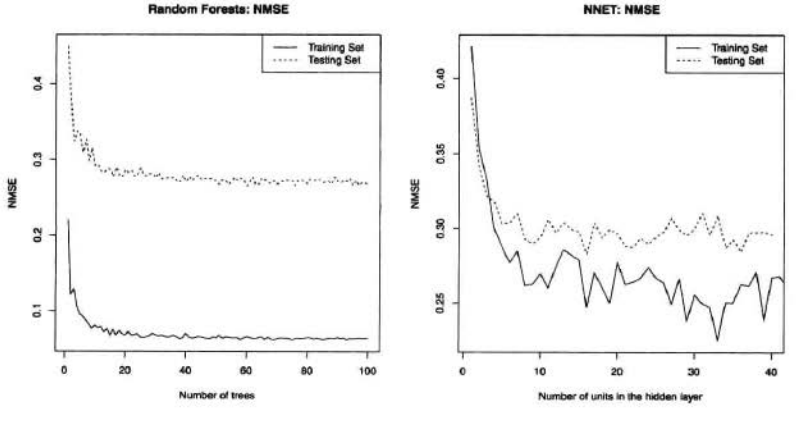


图 3.3 不同数目树的随机森林(左)及不同数目隐藏层节点的神经网络(右)对训练集和测试集的NMSE的影响

从图3.3可以看出, 随着树的数目的增加, 随机森林训练集和测试集的NMSE都稳定地下降, 而且最终稳定在各自的一个水平上. 而随着隐藏层节点的增加, 人工神经网络训练集和测试集的NMSE总体上也在下降, 但没有随机森林那么稳定.

有人看到随机森林训练集误差很小而测试集误差相对较大, 就以为有过拟合, 实际上, 只要测试集误差不随模型复杂性的增加而增长就不是过拟合. 那么什么是过拟合呢? 如果训练集的误差在下降, 而测试集的误差在下降到一定程度后反而上升, 就是过拟合. 随机森林没有过拟合现象, 这里的神经网络还没有出现过拟合现象. 但通常相信, 当神经网络的隐藏层中的节点增加到一定数目时, 就可能产生过拟合现象. 神经网络越复杂, 参数也越多, 当参数的数目接近观测值的数目时, 训练集可能会产生完全拟合, 测试集的误差就会很大.

第四章　横截面数据分类: 经典方法

分类(classification)就是经典多元统计中所说的判别(discrimination). 分类的目的与回归相似, 但这时因变量是有若干水平的分类变量(定性变量). 当因变量只有两个水平, 而自变量没有太多分类变量时, 可以采用logistic回归及probit回归, 也可以用经典判别分析. 在因变量有多于两个水平, 而且自变量没有太多的分类变量时, 可以用经典多元分析中的判别分析. 但无论在何种因变量水平数量的情况下, 如果自变量包含较多分类变量, 判别分析就不能用了, 这时可以尝试采用机器学习的各种方法, 比如决策树的分类树、神经网络、boosting、bagging、随机森林、最近邻方法、支持向量机及神经网络等. 而且机器学习方法往往比经典方法预测得更加准确.

4.1　logistic回归和probit回归

由于logistic回归和probit回归都属于广义线性模型的特例, 后面要介绍的Poisson对数线性模型也是广义线性模型的特例, 这里首先简单回顾一下广义线性模型的基本概念.

4.1.1　广义线性模型简单回顾

下面对广义线性模型的基本内容做一简单介绍. 记自变量的线性表示式为$\eta = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p = \boldsymbol{x}^\top \boldsymbol{\beta}$, 这里$x_1$可以是代表常数项的1.

首先回顾线性模型

$$Y = \boldsymbol{x}^\top \boldsymbol{\beta} + \epsilon,$$

这里假定$\epsilon \sim N(0, \sigma^2)$. 这时, 由于

$$Y \sim N(\boldsymbol{x}^\top \boldsymbol{\beta}, \sigma^2),$$

可以通过最大似然法得到参数$\boldsymbol{\beta}$及σ^2的估计, 而且这个估计在Y的正态假定下等价于最小二乘估计. 在取期望之后, 得到

$$\mu = E(Y) = \boldsymbol{x}^\top \boldsymbol{\beta} = \eta \quad \text{或者} \quad \mu = \eta.$$

如果μ和η之间用一个函数连接起来, 比如

$$\ell(\mu) = \eta \quad \text{或者} \quad \mu = \ell^{-1}(\eta) = m(\eta),$$

而且Y服从(正态分布也在其中的)指数族分布, 则我们就有了广义线性模型(generalized linear model, GLM), 这时, 作用在均值μ上的变换函数$\ell(\cdot)$称为连接函数(link function), 而其逆函数函数$m(\cdot)$称为均值函数. 由于Y的分布属于已知的指数族分布, 完全可以通过最大似然法得到参数$\boldsymbol{\beta}$及分布参数的估

计. 这完全类似于正态情况. 当然, 对于广义线性模型, 最大似然估计必须通过计算机的迭代算法得到, 不像正态情况时有封闭的数学表达式.

如果因变量观测值Y来自指数族分布, 那么它的密度函数的形式为:

$$f_Y(y;\theta,\phi) = a(y,\phi)\exp\frac{b(\theta)T(y)-\kappa(\theta)}{\phi},$$

式中所有函数都是已知的, 如果$b(\theta)=\theta$, 则称为典则形式(所有分布都可通过变换成典则形式); 再者, 如果$T(y)=y$, 则θ称为典则参数. 假定$\mu \equiv E(Y)$为Y的期望. 这时, 均值等于分布中$\kappa(\theta)$的一阶导数, 而方差又和其二阶导数有关:

$$\mu = E(Y) = \kappa'(\theta);\ Var(Y) = \kappa''(\theta)\phi.$$

连接函数$\theta=\eta$称为典则连接函数(canonical link function), 它使得数学推导简单很多, 虽然在数学上有方便之处, 但没有任何证据说明典则连接函数在拟合实际数据时比其他连接函数要好. 由于R中的函数`glm()`对某分布的默认连接函数是其典则连接函数, 表4.1列出了某些指数族分布的典则连接函数.

表 4.1 某些指数族分布的典则连接函数

分布	连接函数在R中的名字	连接函数$\ell(\mu)$	均值函数$m(\eta)$
正态(高斯)	`identity`	$\boldsymbol{x}^\top\boldsymbol{\beta}=\mu$	$\mu=\boldsymbol{x}^\top\boldsymbol{\beta}$
指数	`inverse`	$\boldsymbol{x}^\top\boldsymbol{\beta}=-\mu^{-1}$	$\mu=-(\boldsymbol{x}^\top\boldsymbol{\beta})^{-1}$
Gamma	`inverse`	$\boldsymbol{x}^\top\boldsymbol{\beta}=-\mu^{-1}$	$\mu=-(\boldsymbol{x}^\top\boldsymbol{\beta})^{-1}$
逆高斯	`1/mu^2`	$\boldsymbol{x}^\top\boldsymbol{\beta}=-\mu^{-2}$	$\mu=(-\boldsymbol{x}^\top\boldsymbol{\beta})^{-1/2}$
Poisson	`log`	$\boldsymbol{x}^\top\boldsymbol{\beta}=\ln(\mu)$	$\mu=\exp(\boldsymbol{x}^\top\boldsymbol{\beta})$
二项	`logit`	$\boldsymbol{x}^\top\boldsymbol{\beta}=\ln\left(\frac{\mu}{1-\mu}\right)$	$\mu=\frac{\exp(\boldsymbol{x}^\top\boldsymbol{\beta})}{1+\exp(\boldsymbol{x}^\top\boldsymbol{\beta})}$

4.1.2 脊柱数据例子

先引进一个数据. 该数据有两个版本, 两者的区别仅仅在于一个版本的因变量有2个水平, 另一个有3个水平. 两个版本的自变量是相同的, 都是定量变量. 可以尝试用经典的logistic回归、probit回归和判别分析来拟合这个数据. 当然, 经典的logistic回归、probit回归及判别分析并不要求自变量完全是数量变量, 可以包含少数定性变量. 定性自变量或者变量水平多了, 这些方法就不能用了.

例 4.1 脊柱数据(column.2C.csv, column.3C.csv) 这个数据[1]来源于Frank and Asuncion (2010). 该数据的自变量(V1,V2,..., V6)为6个生物力学特征, 都是数量. 研究目的之一是根据这些特征把患者划分为3类(因变量V7): 正常(100人, 代码为NO: normal)、椎间盘突出(60人, 代码为DH: disk hernia)或腰椎滑脱(150人, 代码为SL: spondilolysthesis); 另一个目的是把患者分为两类: 正常(100人, 代码为NO: normal), 不正常(210人, 代码为AB: abnormal). 在原始数据中变量V6的第116个观测值是明显的异常值, 会影响拟合运算, 因此, 我们对其进行了修正(第一个数据把原来的418.54换成了46.7093, 第二个数据换成49.5902), 这里的数据集是修正后的.

后面分别对脊柱数据(column.2C.csv)的logistic和probit回归方法及脊柱数据(column.3C.csv)的经典判别分析方法进行讨论.

4.1.3 logistic回归

对于一系列有两个结果的随机试验, 最简单的概率模型就是Bernoulli试验及Bernoulli分布, 那里假定成功的概率为p, 失败的概率为$1-p$. 二项分布就是由多次Bernoulli试验导出的. 在实际生活中, 有各种不同的其他因素干扰试验结果, 这样成功和失败的概率就不是固定的, 而是其他变量的一个函数. 假定自变量向量为$\boldsymbol{X}$, 那么一个简单的函数为:

$$\ln\frac{p}{1-p}=\boldsymbol{X}^{\top}\boldsymbol{\beta} \quad \text{或者} \quad p=\frac{\boldsymbol{X}^{\top}\boldsymbol{\beta}}{1+\boldsymbol{X}^{\top}\boldsymbol{\beta}} \tag{4.1}$$

式中, $\boldsymbol{\beta}$为待估计系数向量. 注意, Bernoulli分布均值$\mu=p$, 这里的连接函数为$\ell(\mu)=\ell(p)=\ln[p/(1-p)]=\boldsymbol{X}^{\top}\boldsymbol{\beta}$.

在简单回归的期望表达式$\mu=E(Y)=\boldsymbol{X}^{\top}\boldsymbol{\beta}$中, 左边不是观测变量, 而是均值参数$\mu$, 方程(4.1)左边的$\ln[p/(1-p)]$也不是可观测的变量, 而是假定背景分布(Bernoulli分布)一个参数的函数, 因为函数$\ln[p/(1-p)]$被称为logit函数, 所以这个模型叫做logistic回归模型, 为广义线性模型的一个特例.

广义线性模型在R中可用函数`glm()`来处理. 和logistic回归类似, 结果也往往类似的一个回归模型为probit回归, 该模型也是基于成功概率为p的Bernoulli试验, 其形式为:

$$\Phi^{-1}(p)=\boldsymbol{X}^{\top}\boldsymbol{\beta} \quad \text{或者} \quad p=\Phi(\boldsymbol{X}^{\top}\boldsymbol{\beta}), \tag{4.2}$$

式中, p为Bernoulli试验的成功概率; Φ为标准正态累积分布函数. 显然, logistic回归和probit回归都是试图把取值范围为整个实数轴的$\boldsymbol{X}^{\top}\boldsymbol{\beta}$和取值为区间$[0,1]$的$p$联系起来: logistic模型(4.1)左边的$\ln[p/(1-p)]$和右边的$\boldsymbol{X}^{\top}\boldsymbol{\beta}$取值

[1]可从网站http://archive.ics.uci.edu/ml/datasets/Vertebral+Column下载.

范围一样, 都是整个实数轴, 而probit模型(4.2)左边的$\Phi^{-1}(p)$和右边的$\boldsymbol{X}^{\top}\boldsymbol{\beta}$也都是取值于整个数轴的. 这里的probit模型的连接函数$\Phi^{-1}(p)$不是典则连接函数, 因此在使用函数`glm()`时要标明, 不能用缺省值.

1. **拟合**

下面的代码用logistic回归来拟合column.2C.csv数据(包括读入数据):

```
w2=read.csv("column.2C.csv")
a=glm(V7~.,w2,family="binomial")#默认logit连接函数
b=step(a)  #做逐步回归筛选变量
summary(b) #结果汇总
```

logistic回归的估计系数及近似正态z检验的结果和分类结果展示在表4.2和表4.3中. 变量V4在逐步回归中被淘汰. 此外, 在输出中AIC=191.32. 由表4.2可知训练出来的模型为:

$$\begin{aligned}\ln\frac{p}{1-p} =& -14.5946 + 0.0789 \times \mathrm{V1} - 0.1588 \times \mathrm{V2} \\ &+ 0.0298 \times \mathrm{V3} + 0.1042 \times \mathrm{V5} - 0.1704 \times \mathrm{V6}. \end{aligned} \tag{4.3}$$

表 4.2 逐步回归筛选变量之后的logistic回归结果输出

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-14.5946	3.1850	-4.58	0.0000
V1	0.0789	0.0231	3.42	0.0006
V2	-0.1588	0.0353	-4.50	0.0000
V3	0.0298	0.0193	1.55	0.1214
V5	0.1042	0.0226	4.60	0.0000
V6	-0.1704	0.0235	-7.26	0.0000

2. **预测**

建立模型的一个重要目的是预测, 我们现在来讨论预测. logistic回归的预测不像线性回归那么简单. 根据拟合方程(4.3), 把变量(这里是V1, V2, V3, V5)的观测值(或者是新数据, 或者是拟合时用的训练集数据)代入, 并不能得到因变量Y的拟合值或预测值, 而只能得到相应于每个观测值的p的估计值, 这样的$\hat{p}_i$值的个数和数据样本量相同.

另外, 值得注意的是, 利用逐步回归或其他方法挑选的模型可能有助于解释模型, 但不一定对预测有帮助. 我们下面的分析还是利用逐步回归之前的拟合模型.

这时, 我们会面对若干问题:

- 如何从每个估计的$\hat{p}_i$来预测因变量的值(对于例4.1是NO和AB). 这时就要选择一个$\hat{p}_i$的阈值, 根据这个阈值来区分是NO还是AB. 最常用的省事的阈值为0.5, 但这不见得是预测时误判率最小的阈值.
- 这个阈值的选择原则是什么? 这至少又有两种选择:
  - 按照总的误判率. 这时认为把NO判断成AB与把AB判断成NO是同等风险的.
  - 按照其他原则, 比如降低最大风险. 在例4.1中, 可以认为把AB判断成NO比把NO判断成AB更会误人治疗.
- 如何在R中实现我们的目标.

首先, 在R中得到的估计的p值是关于NO的概率还是AB的概率(哪个代表"成功")? 由于在拟合时我们并没有明确地指明这一点, R自动把p作为排序靠后的水平的概率. 也就是说, 运行代码`levels(w2$V7)`, 得到 `"AB" "NO"`. 这意味着p是NO的概率(R自动按照字典顺序排列).

下面是一个按照总误判率来寻找p的阈值的函数:

```
BI=function(D,w,ff,fm="binomial"){a=glm(ff,w,family=fm)
z=predict(a,w,type="response");ee=NULL
for(p in seq(.1,.9,.01)){u=rep(levels(w[,D])[2],nrow(w))
u[!(z>p)]=levels(w[,D])[1];e=sum(u!=w[,D])/nrow(w)
ee=rbind(ee,c(p,e))};I=which(ee[,2]==min(ee[,2]))
P=ee[min(I),1];return(ee[min(I),])}
```

其中, `D`说明因变量是第几个变量(例4.1中`D=7`); `w`是数据名字, `ff`是拟合公式; `fm`是模型. 这里的p值是从0.1到0.9按照0.01的间隔搜索的. 对于例4.1数据, 实行的代码为:

```
(p.2C=BI(7,w2,V7~.))
```

得到阈值为0.67, 这时对于训练集的误判率为0.1322581. 为了得到对训练集是如何误判的, 运行下面代码:

```
D=7;p=p.2C[1];a=glm(V7~.,w2,family="binomial")
z=(predict(a,w2,type="response")>p)
u=rep(levels(w[,D])[2],nrow(w))
u[!(z>p)]=levels(w[,D])[1];(zz=table(w2[,7],u))
(sum(zz)-sum(diag(zz)))/sum(zz)#计算总误判率
```

得到打印结果, 即表4.3的结果, 这里行代表正确的类, 列代表模型判断的类, 对角线外为错判个数. 误判率为0.1322581.

表 4.3 logistic回归分类结果

	AB	NO
AB	201	9
NO	32	68

这种矩阵称为混淆矩阵(confusion matrix). 下面再考虑按照把AB判断成NO的误判率减少最多的原则来寻找p的阈值的函数:

```
BI2=function(D,w,ff,fm="binomial"){a=glm(ff,w,family=fm)
z=predict(a,w,type="response")
ee=NULL;for(p in seq(.1,.9,.01))
{u=rep(levels(w[,D])[2],nrow(w));u[!(z>p)]=levels(w[,D])[1]
e=sum(u=="NO"&w[,D]=="AB")/sum(w[,D]=="AB")
ee=rbind(ee,c(p,e))}
I=which(ee[,2]==min(ee[,2]));P=ee[min(I),1]
return(ee[min(I),])}
```

执行这个函数, 得到

```
> (p.2C2=BI2(7,w2,V7~.))
[1] 0.9 0.0
```

这意味着把阈值提高到0.9, 就不会有AB到NO的误判了(误判率为0). 自然, 总的误判率会提高. 表4.4为混淆矩阵结果, 这里行代表正确的类, 列代表模型判断的类, 对角线外为错判个数. 总误判率为0.2387097, 比前面结果大, 这是因为把AB到NO的误判减少为0造成把NO到AB的误判增加. 大家还可以根据具体的问题来平衡这两种误判, 这就留给读者了. 相信读者会写出更有效的函数.

表 4.4 logistic回归分类结果

	AB	NO
AB	210	0
NO	74	26

得到判别结果的代码为:

```
D=7;p=p.2C2[1];a=glm(V7~.,w2,family="binomial")
z=(predict(a,w2,type="response")>p)
u=rep(levels(w[,D])[2],nrow(w));u[!(z>p)]=levels(w[,D])[1]
(zz=table(w2[,7],u));(sum(zz)-sum(diag(zz)))/sum(zz)
```

4.1.4 probit回归

下面用probit回归来处理例4.1 column.2C.csv数据. 在正常情况下, probit回归应该有选项family=binomial(link = "probit"), 但是由于该数据的变量V6无法进入计算(有出错信息), 我们又不愿去掉该变量, 因此在代码中改用准probit选项family=quasibinomial(link = "probit"). 这种选项不能计算AIC, 也就无法用逐步回归来选择变量了. 首先还是用前面的BI()函数选择阈值:

```
D=7;w=w2;ff=V7~.;fm=quasibinomial(link = "probit")
(pp.2C=BI(D,w2,ff,fm))
```

得到阈值为0.66, 这时总误判率为0.1290323. 为了看判别结果, 运行

```
D=7;p=pp.2C[1];a=glm(V7~.,w2,family=fm);summary(a)
z=(predict(a,w2,type="response")>p)
u=rep(levels(w[,D])[2],nrow(w));u[!(z>p)]=levels(w[,D])[1]
(zz=table(w2[,7],u));(sum(zz)-sum(diag(zz)))/sum(zz)
```

得到的probit回归的估计系数及近似正态z检验的结果和分类结果展示在表4.5和表4.6中.

表 4.5 准probit回归输出结果

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-8.3586	1.4318	-5.84	0.0000
V1	7.1004	16.5392	0.43	0.6680
V2	-7.1485	16.5407	-0.43	0.6659
V3	0.0196	0.0092	2.12	0.0348
V4	-7.0566	16.5400	-0.43	0.6699
V5	0.0597	0.0102	5.87	0.0000
V6	-0.0987	0.0105	-9.36	0.0000

由表4.5可知训练出来的模型为:

$$\begin{aligned}\Phi^{-1}(p) = &-8.3586 + 7.1004 \times \mathrm{V1} - 7.1485 \times \mathrm{V2} + 0.0196 \times \mathrm{V3} \\ &- 7.0566 \times \mathrm{V4} + 0.0597 \times \mathrm{V5} - 0.0987 \times \mathrm{V6}.\end{aligned}$$

表4.6中混淆矩阵的行代表正确的类, 列代表模型判断的类, 对角线外为误判个数. 误判率为0.1290323, 和表4.3比较, probit回归仅比logistic回归少判错1个观测值. 类似于logistic回归, 也可以运行使得AB到NO的误判尽量减少的程序:

表 4.6 准probit回归分类结果

	AB	NO
AB	201	9
NO	31	69

```
(pp.2C2=BI2(7,w2,V7~.,fm))
D=7;p=pp.2C2[1];a=glm(V7~.,w2,family=fm);summary(a)
z=(predict(a,w2,type="response")>p)
u=rep(levels(w[,D])[2],nrow(w));u[!(z>p)]=levels(w[,D])[1]
(zz=table(w2[,7],u));(sum(zz)-sum(diag(zz)))/sum(zz)
```

得到和logistic回归同样的结果. 这里就不重复了.

4.2 经典判别分析

例4.1的脊柱数据column.2C.csv和column.3C.csv均可以用判别分析的方法来处理. 经典判别分析的原理并不复杂, 简单地描述如下: 如果作为因变量的分类变量有k个取值(或水平、类), 而自变量包含p个变量, 那么每一个观测值就是p维空间的一个点. 整个训练集的各个点(假定有n个点)就按照已知的类别在p维空间中形成共有n个点的k个点群. 那么, 对于一个未知类别的点, 离哪个群近, 就可以分到那个群. 关键的细节问题是如何定义"远近"或距离. 还要确定一些准则和标准, 目的是使得每一点群内部的点尽量接近, 同时使各群之间尽量分开. 不同的准则造就了不同的判别或分类方法.

下面要用程序包`MASS`[1]的R函数`lda()`所基于的Fisher判别法使得每一点群内部的点尽可能接近, 而使各群之间尽量分开, 它利用了线性代数中的特征值和特征向量工具, 把原先的p维空间投影到能够把各群分得最开的低维空间上. 这使得分类更加有道理. Fisher判别法没有像其他线性判别方法那样明确要求假定数据服从多元正态分布. 除了`lda()`函数之外, 我们还使用程序包`mda`[2]中的`mda()`函数(这是一种混合的较灵活的判别分析方法), 以及程序包`fda`[3]中的`fda()`函数(这也是一种灵活的判别分析方法). 由于篇幅有限, 关于这些判别方法的细节就不做详细介绍了, 请参看有关文献.

[1] Venables, W. N. & Ripley, B. D. (2002) *Modern Applied Statistics with S*. Fourth Edition. Springer, New York. ISBN 0-387-95457-0.

[2] S original by Trevor Hastie & Robert Tibshirani. Original R port by Friedrich Leisch, Kurt Hornik and Brian D. Ripley (2013). mda: Mixture and flexible discriminant analysis. R package version 0.4-4. http://CRAN.R-project.org/package=mda.

[3] J. O. Ramsay, Hadley Wickham, Spencer Graves and Giles Hooker (2014). fda: Functional Data Analysis. R package version 2.4.4. http://CRAN.R-project.org/package=fda.

4.2.1 关于例4.1 column.2C.csv数据的两分类判别

1. 线性判别分析

先用的是程序包MASS中的函数lda()(linear discriminant analysis—线性判别分析). 关于例4.1 column.2C.csv数据, 程序代码(包括读入数据)如下:

```
library(MASS)
w2=read.csv("column.2C.csv")
a=lda(V7~.,data=w2)
b=predict(a,w2)$class
(zz=table(w2[,7],b))
(sum(zz)-sum(diag(zz)))/sum(zz)
```

得出了对column.2C.csv数据的判别情况(表4.7混淆矩阵), 这里行代表正确的类, 列代表模型判断的类, 对角线外为错判个数. 得到对训练集(全部数据)的误判率为0.1483871, 不如选择最优阈值的logistic回归及probit回归. 这个结果是由输出中的判别函数(用print(a)可得到, 这里没有显示)决定的, 该函数是自变量的线性组合, 是六维空间中的超平面.

表 4.7 函数lda对例4.1 column.2C.csv数据的判别结果

	AB	NO
AB	188	22
NO	24	76

2. 混合线性判别分析

用函数mda()对例4.1 column.2C.csv数据进行分析的代码如下:

```
library(mda);set.seed(1010)
a=mda(V7~.,data=w2)
b=predict(a,w2)
(zz=table(w2[,7],b));(sum(zz)-sum(diag(zz)))/sum(zz)
```

表4.8为函数mda()的判别结果(混淆矩阵). 得到对训练集(全部数据)的误判率为0.1354839, 比前面结果略有改进.

3. 灵活线性判别分析

用函数fda()对例4.1 column.2C.csv数据进行分析的代码如下:

```
library(fda)
```

表 4.8 函数mda对column.2C.csv数据的判别结果

	AB	NO
AB	187	23
NO	19	81

```
a=fda(V7~.,data=w2)
b=predict(a,w2)
(zz=table(w2[,7],b));(sum(zz)-sum(diag(zz)))/sum(zz)
```

表4.9为函数fda()的判别结果(混淆矩阵). 得到对训练集(全部数据)的误判率为0.1451613, 介于前面两个结果之间.

表 4.9 函数fda对例4.1 column.2C.csv数据的判别结果

	AB	NO
AB	188	22
NO	23	77

4.2.2 例4.1 column.2C.csv数据两分类判别诸方法的10折交叉验证结果

为了把例4.1 column.2C.csv数据随机分为Z份, 我们使用下面函数:

```
Fold=function(Z=10,w,D,seed=7777){
n=nrow(w);d=1:n;dd=list()
e=levels(w[,D]);T=length(e)#因变量T类
set.seed(seed)
for(i in 1:T){
d0=d[w[,D]==e[i]];j=length(d0)
ZT=rep(1:Z,ceiling(j/Z))[1:j]
id=cbind(sample(ZT,length(ZT)),d0);dd[[i]]=id}
#上面每个dd[[i]]是随机1:Z及i类的下标集组成的矩阵
mm=list()
for(i in 1:Z){u=NULL;
for(j in 1:T)u=c(u,dd[[j]][dd[[j]][,1]==i,2])
mm[[i]]=u} #mm[[i]]为第i个下标集: i=1,2,...,Z
return(mm)}#输出Z个下标集
```

函数Fold()中的变元Z为折数(缺省值为10), 变元w为数据名, 变元D为因变量的位置(第几个变量), 变元seed为随机种子(缺省值为7777). 该程序输出10个下标集合.

这个函数和回归时所用的CV()函数的区别在于, 函数Fold()尽量使得因变量的各个水平在各个折中的数目大体相同. 这里对这个函数不做详细解释, 请读者自己理解. 相信读者可以写出更好、更漂亮的程序. 后面还要多次用到函数Fold(). 函数Fold()的不足之处在于它仅仅平衡了因变量的各个水平, 如果自变量有多个定性变量, 则无法考虑它们水平的平衡(例4.1没有这个问题, 因为自变量都是定量变量).

运行下面代码就产生了数据column.2C.csv的10个随机子集的下标集, 作为10折交叉验证的基础.

```
w=read.csv("column.2C.csv")
D=7;Z=10;n=nrow(w);mm=Fold(Z,w,D,8888)
```

然后运行logistic回归、lda判别分析、mda判别分析、fda判别分析的10折交叉验证, 代码如下:

```
ff=paste(names(w)[D],"~.",sep="");ff=as.formula(ff)
p=BI(D,w,ff)[1] #logistic回归用最优阈值
ERROR=matrix(0,Z,4);J=1
for(i in 1:Z){m=mm[[i]]
a=glm(V7~.,w2,family="binomial")
z=(predict(a,w2[m,],type="response")>p)#测试集预测
u=rep(levels(w[m,D])[2],length(m));u[!z]=levels(w[m,D])[1]
ERROR[i,J]=sum(w[m,D]!=u)/length(m)}
J=J+1;for(i in 1:Z)
{m=mm[[i]]
a=lda(ff,w[-m,])
ERROR[i,J]=sum(w[m,D]!=predict(a,w[m,])$class)/length(m)}
J=J+1;for(i in 1:Z)
{m=mm[[i]]
set.seed(1010);a=mda(ff,w[-m,])#混合线性判别
ERROR[i,J]=sum(w[m,D]!=predict(a,w[m,]))/length(m)}
J=J+1;for(i in 1:Z)
{m=mm[[i]]
a=fda(ff,w[-m,])
ERROR[i,J]=sum(w[m,D]!=predict(a,w[m,]))/length(m)}
```

得到结果, 再运行下面代码以打印出结果:

```
ERROR=data.frame(ERROR)
names(ERROR)=c("logistic","lda","mda","fda")
ERROR;(ME=apply(ERROR,2,mean))
```

得到各种方法在各折的误判率为:

```
    logistic        lda        mda        fda
1  0.09677419 0.19354839 0.12903226 0.19354839
2  0.12903226 0.29032258 0.25806452 0.29032258
3  0.06451613 0.12903226 0.06451613 0.12903226
4  0.16129032 0.16129032 0.19354839 0.16129032
5  0.12903226 0.09677419 0.06451613 0.09677419
6  0.16129032 0.16129032 0.09677419 0.16129032
7  0.09677419 0.16129032 0.09677419 0.16129032
8  0.22580645 0.12903226 0.16129032 0.12903226
9  0.06451613 0.12903226 0.16129032 0.12903226
10 0.19354839 0.19354839 0.19354839 0.19354839
```

平均起来, 得到若干方法对例4.1 column.2C.csv数据的测试集误判率表(见表4.10). 显然, 选择最优阈值的logistic模型预测的误判率最低, mda方法其次, lda和fda结果差不多. 当然, 交叉验证的各个数据子集的选择有随机性, 但差别不会太大. 一般来说, 在因变量为二分变量的情况下, logistic分类结果要优于线性判别分析, 如果不选择最优阈值原则, 则不一定是这样的.

表 4.10 若干方法对例4.1 column.2C.csv数据的测试集误判率

方法	误判率
logistic	0.1322581
lda	0.1645161
mda	0.1419355
fda	0.1645161

4.2.3 关于例4.1 column.3C.csv数据的三分类判别

关于例4.1 column.3C.csv数据, 也用线性判别分析(lda), mda, fda三种方法来分类. 程序代码和前面类似.

1. 线性判别分析

先用的是程序包MASS中的函数lda(). 关于例4.1 column.3C.csv数据, 程序(包括读入数据)代码如下:

```
library(MASS)
w3=read.csv("column.3C.csv")
a=lda(V7~.,data=w3)
b=predict(a,w3)$class
(zz=table(w3[,7],b))
(sum(zz)-sum(diag(zz)))/sum(zz)
```

得出了对column.3C.csv数据的判别情况(见表4.11), 这里行代表正确的类, 列代表模型判断的类, 对角线外为错判个数. 得到对训练集(全部数据)的误判率为0.1612903.

表 4.11 函数lda对例4.1 column.3C.csv数据的判别结果

	DH	NO	SL
DH	40	20	0
NO	12	85	3
SL	4	11	135

2. 混合线性判别分析

用函数mda()对例4.1 column.3C.csv数据进行分析的代码如下:

```
library(mda)
w3=read.csv("column.3C.csv")
set.seed(1010);a=mda(V7~.,data=w3)
b=predict(a,w3)$class
(zz=table(w3[,7],b))
(sum(zz)-sum(diag(zz)))/sum(zz)
```

得到误判率为0.1612903, 和前面结果完全一样, 对训练集(全部数据)误判的结果也和表4.11完全一样.

3. 灵活线性判别分析

用函数fda()对例4.1 column.3C.csv数据进行分析的代码如下:

```
library(fda)
a=fda(V7~.,data=w3)
```

```
b=predict(a,w3)
(zz=table(w3[,7],b))
(sum(zz)-sum(diag(zz)))/sum(zz)
```

得到误判率为0.1612903, 和前面结果完全一样, 对训练集(全部数据)误判的结果也和表4.11完全一样.

经典的线性判别分析有很多限制, 其理论推导需要一些无法验证的假定. 下一节我们通过一个数据来引进其他的分类方法, 特别是在机器学习或数据挖掘中常用的方法. 这些方法对分布不做任何假定.

4.2.4 关于例4.1 column.3C.csv数据的三分类判别的三种方法的10折交叉验证结果

运行下面代码就产生了数据column.3C.csv的10个随机子集的下标集, 以其作为10折交叉验证的基础.

```
w=read.csv("column.3C.csv")
D=7;Z=10;n=nrow(w);mm=Fold(Z,w,D,8888)
```

然后运行lda判别分析、mda判别分析、fda判别分析的10折交叉验证, 代码如下:

```
ff=paste(names(w)[D],"~.",sep="");ff=as.formula(ff)
ERROR=matrix(0,Z,3);J=1
for(i in 1:Z)
{m=mm[[i]]
a=lda(ff,w[-m,])
ERROR[i,J]=sum(w[m,D]!=predict(a,w[m,])$class)/length(m)}
J=J+1;for(i in 1:Z)
{m=mm[[i]]
set.seed(1010);a=mda(ff,w[-m,])#混合线性判别
ERROR[i,J]=sum(w[m,D]!=predict(a,w[m,]))/length(m)}
J=J+1;for(i in 1:Z)
{m=mm[[i]]
a=fda(ff,w[-m,])
ERROR[i,J]=sum(w[m,D]!=predict(a,w[m,]))/length(m)}

ERROR=data.frame(ERROR)
names(ERROR)=c("lda","mda","fda")
```

```
(ME=apply(ERROR,2,mean));ERROR
```

得到各种方法在各折的误判率为:

```
          lda       mda       fda
1  0.12903226 0.09677419 0.12903226
2  0.09677419 0.09677419 0.09677419
3  0.16129032 0.16129032 0.16129032
4  0.19354839 0.16129032 0.19354839
5  0.22580645 0.12903226 0.22580645
6  0.16129032 0.12903226 0.16129032
7  0.25806452 0.19354839 0.25806452
8  0.09677419 0.09677419 0.09677419
9  0.16129032 0.16129032 0.16129032
10 0.22580645 0.19354839 0.22580645
```

其平均数为:

```
      lda       mda       fda
0.1709677 0.1419355 0.1709677
```

显然, lda和fda完全相同, 而平均起来, mda方法比lda和fda好些.

第五章　横截面数据分类: 机器学习方法

5.1　对变量没有任何限制: 机器学习分类方法

5.1.1　概论和例子

在因变量为分类变量而自变量含有多个分类变量或分类变量水平较多的情况下, 一般本科统计教科书所讲的经典统计基本上都无能为力, 但这些教科书都不介绍这一类数据, 似乎自变量与因变量都是分类变量的数据根本不存在一样, 这很难让人理解. 实际上, 这一类数据非常普遍, 涉及自然科学和社会科学的各个方面. 一个教师不能仅仅介绍自己知道的或者教科书范围内的东西, 还应该引导学生增加对广大未知世界的好奇心及继续探索的动力. 科学是在对未知世界不断进行探索和开拓的过程中发展的. 只停留在接受和背诵前人所积累的知识上是没有前途的.

下面我们将通过一个数值例子介绍各种在机器学习或数据挖掘中常用的方法.

例 5.1　胎心宫缩监护(cardiotocography, CTG)**数据**(CTG.xls)　这个数据[1]来源于Frank and Asuncion (2010). 该数据有2129个观测值及23个变量, 包含了致命心律(fetal heart rate, FHR)的各种度量以及基于监护记录的由专家分类的宫缩(uterine contraction, UC)特征. 这些变量情况列示于表5.1. 表中最后三个分类变量的水平为: Tendency(FHR直方图的趋势)有三个水平: −1 =左不对称, 0=对称, 1=右不对称; CLASS(FHR分类代码)用1∼10表示从平静睡眠到可疑的10种活动状况; NSP(胎儿状态分类代码)有三个水平: 1=正常, 2=疑似, 3=病态.

我们尝试用前面的头22个变量作为自变量来预测作为因变量的(最后一个变量)NSP的类别. 在查看了数据之后, 我们删去了3个有缺失值的观测值, 形成新的数据文件CTG.NAOMIT.csv. 下面将介绍几种分类方法, 包括决策树(分类树)、adaboost、bagging、随机森林、SVM、最近邻方法等. 其中有些已经在前面介绍过, 后面只需要介绍它们的分类功能.

5.1.2　产生交叉验证数据集

为了比较这些方法, 这里还将采取10折交叉验证的方法. 由于因变量有三个类别(1=正常, 2=疑似, 3=病态), 它们分别有1655, 295, 176个观测值, 为了

[1]该数据可从网站http://archive.ics.uci.edu/ml/datasets/Cardiotocography下载.

表 5.1 例5.1 胎心宫缩监护数据变量情况

变量名	描述	性质	变量名	描述	性质
LB	FHR基线(心跳数/分)	定量	Min	FHR直方图的最小值	定量
AC	每秒钟加速数目	定量	Max	FHR直方图的最大值	定量
FM	每秒钟胎儿动作数目	定量	Nmax	FHR直方图的峰值数	定量
UC	每秒钟子宫收缩数目	定量	Nzeros	FHR直方图的零值数	定量
DL	每秒钟轻微减速数目	定量	Mode	FHR直方图的众数	定量
DS	每秒钟严重减速数目	定量	Mean	FHR直方图的均值	定量
DP	每秒钟持续减速数目	定量	Median	FHR直方图的中位数	定量
ASTV	短期变异的时间百分比	定量	Variance	FHR直方图的方差	定量
MSTV	短期变异的平均值	定量	Tendency	FHR直方图的趋势	分类
ALTV	长期变异的时间百分比	定量	CLASS	FHR分类代码	分类
MLTV	长期变异的平均值	定量	NSP	胎儿状态分类代码	分类
Width	FHR直方图的宽度	定量			

平衡, 把这三种类型的每一种都随机地分为10份. 读入数据及平衡地把观测值随机分为10份的函数就是在4.2.2节已经引进的函数Fold():

```
Fold=function(Z=10,w,D,seed=7777){
n=nrow(w)
d=1:n;dd=list()
e=levels(w[,D])
T=length(e);set.seed(seed)
for(i in 1:T){
d0=d[w[,D]==e[i]];j=length(d0)
ZT=rep(1:Z,ceiling(j/Z))[1:j]
id=cbind(sample(ZT,length(ZT)),d0);dd[[i]]=id}
mm=list();for(i in 1:Z){u=NULL;
for(j in 1:T)u=c(u,dd[[j]][dd[[j]][,1]==i,2])
mm[[i]]=u}
return(mm)}#输出Z个下标集
```

下面代码读入数据, 并把用哑元表示的定性变量因子化, 还利用函数Fold()来确定10个交叉验证数据集:

```
w=read.csv("CTG.NAOMIT.csv") #读入数据
F=21:23;for(i in F)w[,i]=factor(w[,i])#因子化最后三个哑元变量
D=23;Z=10;n=nrow(w);mm=Fold(Z,w,D,8888)#D是因变量位置; Z是折数
```

5.2 决策树分类(分类树)

5.2.1 拟合全部数据

前面介绍过回归树, 这里的分类树和回归树基本相同, 只是一些细节有所区别. 决策树数字意义的说明是在输出的第二行中(n=2126下面的一行). 为了描述分类树, 先运行胎心宫缩监护例子的全部数据. 仍使用前面用过的程序包rpart中的函数rpart(). 把全部数据作为训练集来估计决策树的代码如下:

```
library(rpart.plot)
(a=rpart(NSP~.,w))#用决策树拟合全部数据并打印输出
rpart.plot(a,type=2,extra=4)#画决策树
```

这产生了下面的打印结果和图5.1所示的决策树.

```
n= 2126

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 2126 471 1 (0.77846 0.13876 0.08278)
  2) CLASS=1,2,3,4,6,7 1681  29 1 (0.98275 0.01725 0.0000) *
  3) CLASS=5,8,9,10 445 179 2 (0.00674 0.59775 0.39551)
    6) CLASS=5,10 269   4 2 (0.01115 0.98513 0.00372) *
    7) CLASS=8,9 176   1 3 (0.00000 0.00568 0.99432) *
```

容易发现, 整个决策树一共有两次分叉, 两次都用了变量CLASS. 从打印结果可以看出, 在根节点(1)上的观测值数目为2126个, 由于NSP三个水平的比例为0.78:0.14:0.08, NSP=1为多数, 按照少数服从多数的原则, 如果这个节点就是终节点, 那么决策为NSP=1, 损失(loss)为471个分错. 在这个节点, 被选中的拆分变量为CLASS, 当CLASS=1,2,3,4,6,7时, 走向左边节点(2), 观测值数目为1681, NSP三个水平之比为0.98:0.02:0.00, 如果立即做决策, 则为NSP=1, 有29个分错(loss), 这是第一个终节点; 而在根节点, 当CLASS=5,8,9,10时, 走向右边节点(3), 观测值数目为445, NSP三个水平之比为0.01:0.60:0.40, 如果立即做决策, 则为NSP=2, 有179个分错. 节点(3)的拆分变量还是CLASS, 当CLASS=5,10时, 走向左边节点(6), 有269个观测值, NSP三个水平之比为0.01:0.99:0.00, 因此决策为NSP=2, 损失为4个分错, 这是第二个终节点; 在节点(3), 当CLASS=8,9时, 走向右边, 有176个观测值, NSP三个水平之比为0.00:0.01:0.99, 因此决策为NSP=3, 损失为1个分错, 这

是第三个终节点. 图5.1除了没有给出每个节点的观测值数目及分错的数目外, 包括了大部分打印信息.

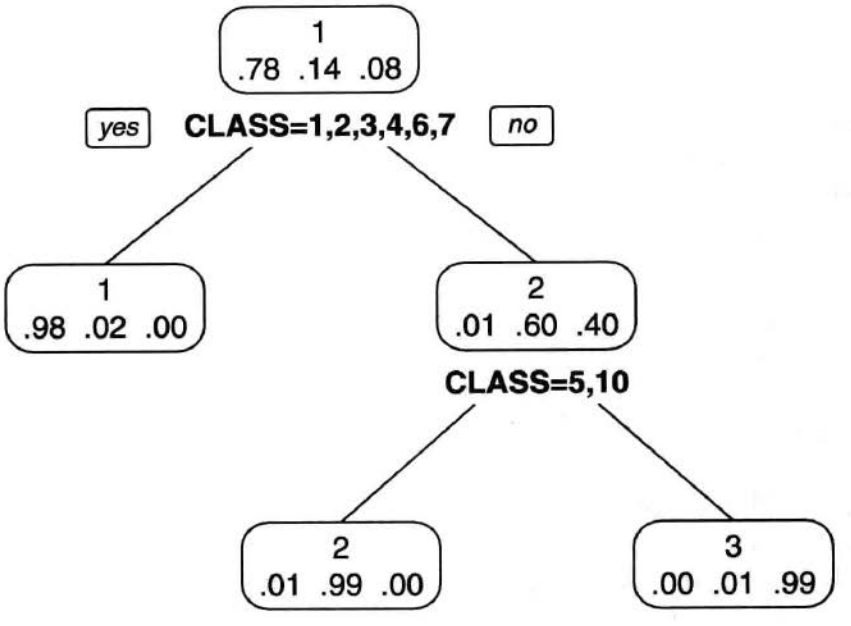


图 5.1 胎心宫缩监护数据的决策树

为了得到对训练集(全部数据)的分类误差, 执行下面的代码:

```
wp=predict(a,w,type="class")
(z=table(w[,D],wp))
sum(w[,D]!=wp)/nrow(w)
```

得到表5.2中决策树对例5.1数据的分类结果(混淆矩阵), 行是真实类, 列是预测类. 一共有34个观测值被分错. 误判率为0.01599247.

表 5.2 决策树对胎心宫缩监护数据的分类结果

	NSP=1	NSP=2	NSP=3
NSP=1	1652	3	0
NSP=2	29	265	1
NSP=3	0	1	175

5.2.2 交叉验证

下面就前面利用函数Fold()所产生的10个随机数据集做10折交叉验证, 这时会产生10棵决策树, 这里就不逐个把树打印出来了, 仅给出测试集的分类平均误判率. 相关的代码如下:

```
library(rpart)
E=rep(0,Z)
for(i in 1:Z){m=mm[[i]];
    n1=length(m);a=rpart(NSP ~ .,w[-m,])
    E[i]=sum(w[m,D]!=predict(a,w[m,],type="class"))/n1}
mean(E)
```

计算结果表明, 10折交叉验证得到的用分类树对例5.1数据做分类的测试集的平均误判率为0.01601179.

5.3 adaboost分类

这里介绍的adaboost是boosting的一种, 是一种组合方法, 这里用分类树作为基本学习器. adaboost (adaptive boosting的简写)可以译为"自适应助推法", 但我们更愿意用简明的英文缩写adaboost. adaboost是一种迭代式的组合算法, 目的是分类. 所用的基础分类器(这里用决策树)一开始可能较弱(即误判率较高), 但随着迭代的进行, 不断通过自助法(bootstrap)加权再抽样, 根据产生的新样本来改进分类器, 每一次迭代都针对前一个分类器对某些观测值的误判缺陷加以修正, 通常的做法是在(放回)抽取样本时对那些误判的观测值增加权重(相当于对正确分类的减少权重), 这样在新的样本中就可能有更多的前一次分错的观测值, 再形成一个新的分类器进入下一轮迭代, 而且在每轮迭代中都对这一轮产生的分类器给出误判率, 最终结果由各个阶段的分类器按照误判率加权(设置权重目的是惩罚误判率大的分类器)投票产生. 这就是"自适应".

adaboost的缺点是对奇异点或离群点可能比较敏感, 但其优点是对过拟合不那么敏感. 这里用的程序包是**adabag**[1], 该程序包包含了adaboost的**boosting()**函数(也就是老版本的**adaboost.M1()**函数, "adaboost.M1"是方法的名称)及下一节要介绍的**bagging()**函数. 函数**boosting()**的默认迭代次数为100次, 即产生100棵决策树.

[1] Alfaro-Cortes, Esteban; Gamez-Martinez, Matias, Garcia-Rubio and Noelia (2011). adabag: Applies AdaBoost.M1, AdaBoost-SAMME and Bagging. R package version 3.0. http://CRAN.R-project.org/package=adabag.

5.3.1 拟合全部数据

为了描述这种方法, 先用adaboost对全部数据进行拟合来介绍一下输出. 用下面的代码:

```
library(adabag)
set.seed(4410)
a=boosting(NSP~.,w)
wp=predict(a,w)$class
(z=table(w[,D],wp))
sum(w[,D]!=wp)/nrow(w)
barplot(a$importance,cex.name=.6)#画出变量重要性图
```

这给出了表5.3展示的分类结果, 行是真实类, 列是预测类, 并得到误判率0.0004703669. 只有一个观测值, 是从NSP=2错分到NSP=1. 上面代码还给出了变量重要性图(见图5.2).

表 5.3 adaboost对胎心宫缩监护数据的分类结果

	NSP=1	NSP=2	NSP=3
NSP=1	1654	1	0
NSP=2	0	295	0
NSP=3	0	0	176

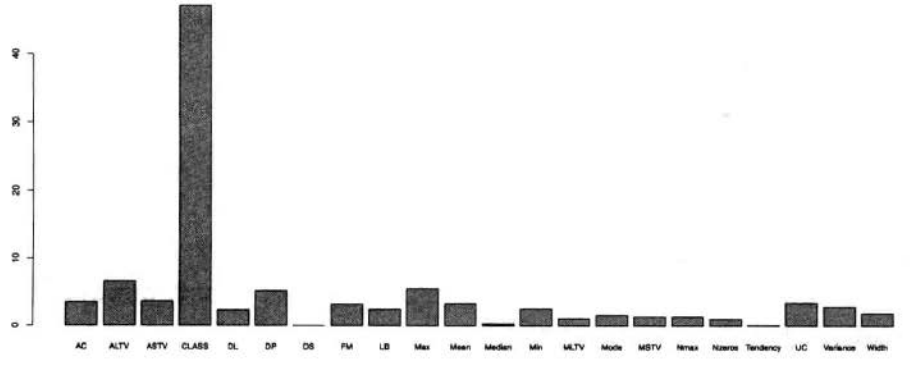


图 5.2 adaboost拟合例5.1数据时的变量重要性图

如果使用a$trees命令, 会打印出boosting过程决策树产生的所有树, 按照函数boosting()参数的默认值会打印出100棵树, 相信你不想这样做.

如果要打印个别的树则可用诸如a$trees[[1]], a$trees[[2]]等代码, 也可以用rpart.plot(a$trees[[100]],type=2,extra=4)画出某一棵(比如第100棵)树(事先需载入画图程序包rpart.plot).

输出中还有许多其他信息, 最实用的是各个变量的相对重要性, 可通过a$importance来得到, 它显示了每个变量被选为拆分变量的次数. 这相应于前面的变量重要性图(图5.2). 图中凸显了变量CLASS的重要性. 画图程序中的变元cex.name控制着条形图下面的变量名字的字体大小. 有时, 一些变量名字不显示, 把图放大后就可以显示了.

关于使用程序包adabag来预测的一个编程缺陷

在回归和分类中, 一般预测, 只要输入类似格式的数据自变量, 就可以得到相应因变量的拟合值, 比如, 在用决策树的时候, 首先用程序包rpart的函数rpart()拟合出模型, 然后对新数据(这里用原来数据中的第11个观测值去掉因变量作为新数据: newdata=w[11,])做预测, 即用newdata[-D]来做预测:

```
> newdata=w[11,]
> a=rpart(NSP ~ .,w)
> predict(a,newdata[-D],type="class")
11
 2
Levels: 1 2 3
```

得到的结果是"2", 和原来w[11,D]的一样, 这说明一切正常.

现在用程序包adabag的boosting()函数做预测, 如果放入完整的newdata=w[11,](命名为newdata)则不成问题:

```
> b=boosting(NSP~.,w)
> predict(b,newdata)$class
[1] "2"
```

但是, 如果在这里用newdata[-D], 则出现问题:

```
> predict(b,newdata[-D])$class
错误于'[.data.frame'(newdata,as.character(object$formula[[2]])):
  undefined columns selected
```

所以必须保持观测值完整, 随便赋以什么值, 但还是有问题, 比如:

```
> newdata[,D]=factor(3)
> predict(b,newdata)$class
[1] "3"
```

这意味着赋以什么值就预测出什么值. 原因是这时只有一个水平:

```
> levels(newdata[,D])
[1] "3"
```

如果把水平补足, 问题就解决了:

```
> levels(newdata[,D])=levels(w[,D])
> levels(newdata[,D])
[1] "1" "2" "3"
> predict(b,newdata)$class
[1] "2"
```

总之, 对于一组有未知因变量的数据集, 比如这里用部分数据作为新的等待预测的数据new.data=w[1:20,], 对第D列随意赋值(在原有水平之内的值, 这里是20个"1"):new.data[,D]=factor(rep("1",20)), 再保持原有水平levels(new.data[,D])=levels(w[,D]), 就可以预测了.

```
new.data=w[1:20,]
new.data[,D]=factor(rep("1",20))#随意赋水平内的值
levels(new.data[,D])=levels(w[,D])
predict(b,new.data)$class
```

这个问题本来不应该费这么多的篇幅, 但由于有的学生在这上面挣扎了一两个晚上还不得其解, 为了避免同样问题, 特地在这里做出如此说明. 不单是函数boosting()的预测, 下面的bagging()在预测时也有同样的问题, 希望读者注意. 在交叉验证中, 在测试集的一些定性变量水平和原来水平不匹配时, 也可能会出现类似问题.

5.3.2 交叉验证

下面就前面利用函数Fold()所产生的10个随机数据集做10折交叉验证, 这里仅给出测试集的分类平均误判率. 相关的代码如下:

```
set.seed(1010)
E=rep(0,Z)
for(i in 1:Z){m=mm[[i]]
n1=length(m)
a=boosting(NSP~.,w[-m,])
E[i]=sum(as.character(w[m,D])!=predict(a,w[m,])$class)/n1}
mean(E)
```

计算结果表明, 10折交叉验证所得到的用adaboost对例5.1数据做分类的测试集的平均误判率为0.01223143. 很自然, 由于测试集数据没有参与建模, 这个结果不如用全部数据的结果好, 但这个结果更接近于这个模型的真实预测误差.

5.4 bagging分类

bagging(bootstrap aggregating的简写)可以译为"自助整合法", 但我们更愿意使用简单的bagging. 顾名思义, 它利用了自助法(bootstrap)放回抽样. 它对训练样本做许多次(比如k次)放回抽样, 每次抽取样本量相同的观测值, 由于是放回抽样, 因此就有了k个不同的样本. 然后, 对每个样本生成一棵决策树. 这样, 每棵树都对一个新的观测值产生一个预测. 如果目的是分类, 那么由这些树的分类结果"投票"来产生bagging分类. 和boosting()一样, 程序包adabag的函数bagging()默认的迭代次数为100次, 也就是产生100棵决策树.

5.4.1 拟合全部数据

先通过用bagging对全部数据进行拟合来介绍一下输出. 用下面的代码:

```
set.seed(1010);D=23
a=bagging(NSP~.,w)
wp=predict(a,w)$class
(z=table(w[,D],wp))
sum(w[,D]!=wp)/nrow(w)
barplot(a$importance, cex.names=0.8)#画出变量重要性图
```

这给出了下面的分类结果(见表5.4), 行是真实类, 列是预测类, 并得到误判率0.01599247.

表 5.4 bagging对胎心宫缩监护数据的分类结果

	NSP=1	NSP=2	NSP=3
NSP=1	1652	3	0
NSP=2	29	265	1
NSP=3	0	1	175

与adaboost一样, 如果使用a$trees命令, 会打印出boosting过程决策树产生的所有树(按照默认值会打印出100棵树), 相信你不想这样做. 如果要打印个别的树, 则可用诸如a$trees[[1]], a$trees[[2]]等代码, 也

可以用rpart.plot(a$trees[[100]],type=2,extra=4)画出某一棵(比如第100棵)树(事先需载入画图程序包rpart.plot). 输出中还有许多其他信息, 比如各个变量的相对重要性可通过a$importance得到, 它显示了每个变量被选为拆分变量的次数. 前面的代码产生了展示变量重要性的图5.3. 这里变量CLASS的相对重要性比adaboost中的更为显著, 这是因为bagging没有进行抽样权重调整, 作为优势变量的CLASS总是占优势. 从中也可以看出两种方法的不同.

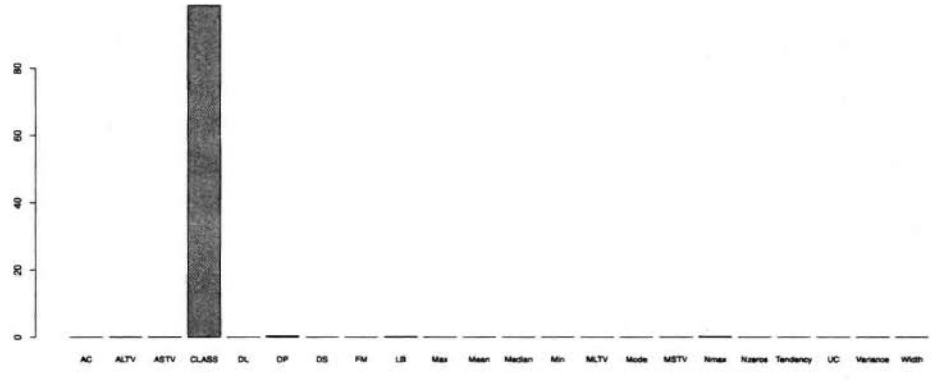


图 5.3 bagging拟合例5.1数据时的变量重要性图

5.4.2 交叉验证

下面再做10折交叉验证, 还是利用前面由函数Fold()得到的10个交叉验证数据集(回忆下面代码):

```
w=read.csv("CTG.NAOMIT.csv") #读入数据
F=21:23;for(i in F)w[,i]=factor(w[,i])
D=23;Z=10;n=nrow(w);mm=Fold(Z,w,D,8888)
```

对于bagging的交叉验证, 代码如下(这里仅给出测试集的分类平均误判率):

```
set.seed(1010)
E=rep(0,Z)
for(i in 1:Z){m=mm[[i]]
    n1=length(m)
```

```
    a=bagging(NSP~.,w[-m,])
    E[i]=sum(as.character(w[m,D])!=predict(a,w[m,])$class)/n1}
mean(E)
```

计算结果表明, 10折交叉验证所得到的用bagging对例5.1数据做分类的测试集的平均误判率为0.01601179.

5.5 随机森林分类

随机森林(random forests)和使用决策树作为基本分类器的bagging有些类似. 以决策树为基本模型的bagging在每次自助法(boostrap)放回抽样之后都产生一棵决策树, 抽多少样本就生成多少棵树, 在生成这些树的时候没有进行更多的干预. 而随机森林也是进行许多次自助法放回抽样; 所得到的样本数目及由此建立的决策树数量要远远多于bagging的样本数目. 随机森林与bagging的关键区别在于, 在生成每棵树的时候, 每个节点的变量都仅仅在随机选出的少数变量中产生. 因此, 不但样本是随机的, 就连每个节点变量的产生都有相当大的随机性.

随机森林让每棵树尽可能生长, 而且不进行修剪. 随机森林不惧怕很大的维数, 即使有数千个变量, 也不必删除. 它还会给出分类中各个变量的重要性. 在一个关于淋巴瘤的基因芯片数据中, 变量个数可以达到4682个, 而样本量仅有81个, 但随机森林可以很好地找到重要的基因(Breiman, 2001). 这种数据在经典统计中根本无法处理, Diaconis and Efron(1983) 曾经说过: "统计经验表明, 基于19个变量和仅仅155个数据点来拟合模型是不明智的." 这种说法显然不适用于算法建模.

程序包randomForest[1]包含了随机森林函数randomForest(), 其建立决策树棵数的默认值为500.

5.5.1 拟合全部数据

下面用随机森林对全部数据进行拟合来介绍一下输出. 用下面的代码:

```
library(randomForest)
set.seed(1010)
a=randomForest(NSP~.,w,importance=TRUE,proximity=TRUE)
wp=predict(a,w)
(z=table(w[,D],wp))
```

[1] A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. *R News* 2(3), 18–22.

```
(E0=(sum(z)-sum(diag(z)))/sum(z))
```

这给出了关于例5.1数据的分类结果(见表5.5), 行是真实类, 列是预测类, 并得到误判率0.0004703669, 仅仅分错一个观测值, 和adaboost类似.

表 5.5 随机森林对胎心宫缩监护数据的分类结果

	NSP=1	NSP=2	NSP=3
NSP=1	1655	0	0
NSP=2	1	294	0
NSP=3	0	0	176

各个变量的相对重要性可用a$importance打印出来, 前三列显示了各个变量对NSP三个水平的相对影响, 后两列分别展示了扰动这些变量对精确度及Gini指数的影响, 相对影响大的就被认为是重要的变量. 这5列指标展示在图5.4中, 图5.4(上)显示了各个变量对NSP三个水平的相对影响, 图5.4(中)和图5.4(下)分别展示了扰动这些变量对精确度及Gini指数的影响.

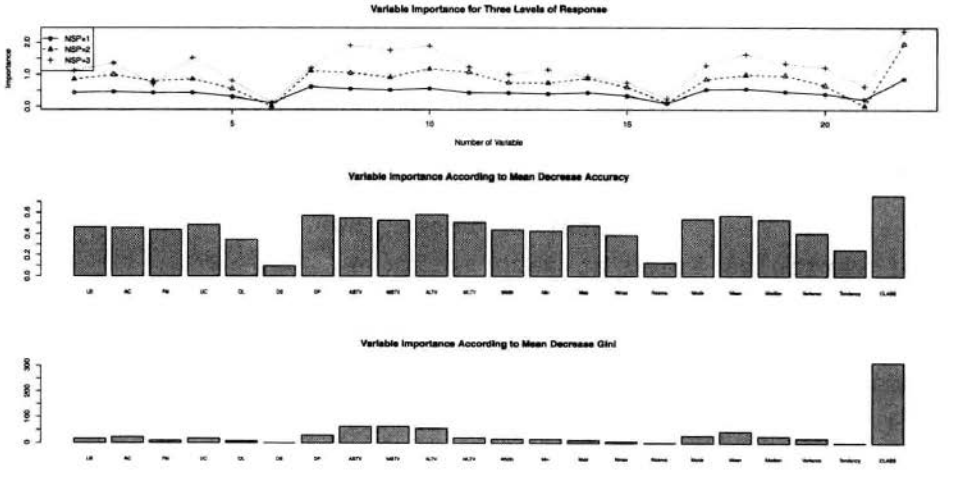


图 5.4 用随机森林拟合例5.1数据时变量的重要性图

绘制图5.4的代码如下:

```
par(mfrow=c(3,1))
matplot(importance(a)[,1:3],type="o",pch=1:3,lty=1:3,col=1,
    xlab="Variable Number", ylab="Importance")
```

```
title("Variable Importance for Three Levels of Response")
legend("topleft",legend=paste("NSP=", 1:3,sep=""),
    pch=1:3,lty=1:3,col=1)
barplot(importance(a)[,4],cex.name=0.6)
title("Variable Importance According to Mean Decrease
    Accuracy")
barplot(importance(a)[,5],cex.name=0.6)
title("Variable Importance According to Mean Decrease Gini")
par(mfrow=c(1,1))
```

基于OOB的结果 用print(a)(或简单地运行a)可以得到一个误差率的OOB估计及基于OOB的分类矩阵. 由于每棵树都由自助法抽样得来, 抽样是放回抽样, 这样每次都有30% ~ 40%的数据没有被抽到, 因此它们成为天然的测试集, 这些观测值数据被称为OOB数据, "OOB"是英文out of bag的缩写. 即漏掉的, 没有装入口袋的, 或在口袋外面的数据. 下面是打印出来的结果:

```
> a

Call:
randomForest(formula=NSP~.,data=w,importance=TRUE,
    proximity=TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 1.27%
Confusion matrix:
     1   2   3 class.error
1 1650   5   0 0.003021148
2   18 276   1 0.064406780
3    0   3 173 0.017045455
```

这个输出给出了OOB测试集得到的误判率, 这里是1.27%, 或0.0127, 称为误判率的OOB估计(OOB estimate of error rate). 这个由OOB测试集得到的误判率和后面做10折交叉验证测试集得到的误判率基本相符, 但和上面用既作为训练集也作为测试集的全部数据得到的结果不一样. 上面打印出来的随机森林关于例5.1的基于OOB数据的分类结果矩阵(混淆矩阵)也可

由a$confusion单独得到.

5.5.2 交叉验证

下面再做随机森林的10折交叉验证, 还是用原先产生的交叉验证数据集, 这里仅仅给出测试集的分类平均误判率. 相关的代码如下:

```
library(randomForest)
set.seed(1010)
E=rep(0,Z)
for(i in 1:Z){m=mm[[i]]
n1=length(m)
a=randomForest(NSP ~ ., data=w[-m,])
E[i]=sum(w[m,D]!=predict(a,w[m,]))/n1}
mean(E)
```

计算结果表明, 10折交叉验证所得到的用随机森林对例5.1数据做分类的测试集的平均误判率为0.01224472, 和做一次拟合的基于OOB的误判率差不了多少, 这表明基于OOB的误判率是非常聪明的误差验证指标.

5.6 支持向量机分类

前面已经简单介绍过支持向量机的原理, 这里不再赘述. 但这里是用程序包e1071[1]中的svm()函数和程序包kernlab[2]中的ksvm()函数来做SVM分类.

5.6.1 拟合全部数据

先用程序包e1071的svm()函数对例5.1的全部数据进行拟合, 代码如下:

```
library(e1071)
a=svm(NSP~ ., data = w,kernal="sigmoid")
wp=predict(a,w)
(z=table(w[,D], wp))
(E=(sum(z)-sum(diag(z)))/sum(z))
```

[1]Evgenia Dimitriadou, Kurt Hornik, Friedrich Leisch, David Meyer and Andreas Weingessel (2011). e1071: Misc Functions of the Department of Statistics (e1071), TU Wien. R package version 1.6. http://CRAN.R-project.org/package=e1071.

[2]Alexandros Karatzoglou, Alex Smola, Kurt Hornik, Achim Zeileis (2004). kernlab - An S4 Package for Kernel Methods in R. *Journal of Statistical Software* 11(9), 1-20. URL http://www.jstatsoft.org/v11/i09/.

这给出了表5.6展示的关于例5.1数据的分类结果, 并得到误判率0.01081844, 错分了23个观测值.

表 5.6 支持向量机(svm函数)对胎心宫缩监护数据的分类结果

	NSP=1	NSP=2	NSP=3
NSP=1	1652	3	0
NSP=2	15	278	2
NSP=3	2	1	173

再用程序包kernlab的svm()函数对例5.1的全部数据进行拟合, 代码如下:

```
library(kernlab)
a=ksvm(NSP~ ., data = w)
wp=predict(a,w)
(z=table(w[,D], wp))
(E=(sum(z)-sum(diag(z)))/sum(z))
```

这给出了表5.7展示的关于例5.1数据的分类结果, 并得到误判率0.008466604, 错分了18个观测值.

表 5.7 支持向量机(ksvm函数)对胎心宫缩监护数据的分类结果

	NSP=1	NSP=2	NSP=3
NSP=1	1652	3	0
NSP=2	12	283	0
NSP=3	2	1	173

5.6.2 交叉验证

下面对原先设定的10个交叉验证数据集再做支持向量机的10折交叉验证, 分别对两种方法做预测, 这里仅给出测试集的分类平均误判率. 相关的代码如下:

```
library(e1071)
E=rep(0,Z)
for(i in 1:Z){m=mm[[i]]
n1=length(m)
a=svm(NSP ~ ., data = w[-m,],kernal="sigmoid")
E[i]=sum(w[m,D]!=predict(a,w[m,]))/n1}
```

```
mean(E)

library(kernlab)
E=rep(0,Z)
for(i in 1:Z){m=mm[[i]]
n1=length(m)
a=ksvm(NSP ~ ., data = w[-m,])
E[i]=sum(w[m,D]!=predict(a,w[m,]))/n1}
mean(E)
```

计算结果表明, 利用10折交叉验证得到的用函数svm()对例5.1数据做分类的测试集的平均误判率为0.01695084. 而利用函数ksvm()所得到的测试集的平均误判率为0.01506175.

5.7 最近邻方法分类

最近邻方法(nearest neighbor algorithm)可能是所有算法建模中最简单的方法. 它基于训练集对测试集进行回归或分类. 每个回归或分类问题都有一些自变量, 它们组成一个多维空间. 首先在空间中假定一个距离. 比如, 在连续型自变量的情况下, 可用欧氏距离、平方欧氏距离、绝对距离等. 在分类问题中, 测试集的一个点应该被判为训练集中离该点最近的k个点中多数点所属的类型, 这种"多数投票方法"一般都加有权重, 离得越近的点, 投票权重越大, 在各种软件中都有各种权重函数以备选择. 在用最近邻方法做回归时, 一个测试集点的因变量的预测值应该等于离它最近的k个训练集点相应的因变量值的平均, 这些平均一般也都是加权平均. 至于k的选择, 一般都用测试集的交叉验证来进行. 这里所用的函数是程序包kknn[1]的函数kknn(). 先用最近邻方法对例5.1的全部数据进行拟合, 代码如下:

```
library(kknn)
a=kknn(NSP~., k=6,train= w,test=w)#k是交叉验证得到的
(z=table(w[,D], a$fit))
(E=(sum(z)-sum(diag(z)))/sum(z))
```

这给出了表5.8展示的关于例5.1数据的分类结果, 并得到误判率0.009407338, 错分了20个观测值. 这是对于$k = 6$得到的, 如果取$k = 2$, 则误判率为0.0004703669. 看上去少些, 但交叉验证时误差就大了. 注意, 这

[1]Klaus Schliep & Klaus Hechenbichler (2014). kknn: Weighted k-Nearest Neighbors. R package version 1.2-5. http://CRAN.R-project.org/package=kknn.

里的$k = 6$是经过多次交叉验证得到的最优近邻数目. 如果不用交叉验证, 则$k = 6$不是最好的选择. 因此, 不做交叉验证不可能得到k取多少最好的信息.

表 5.8 最近邻方法对胎心宫缩监护数据的分类结果

	NSP=1	NSP=2	NSP=3
NSP=1	1653	1	1
NSP=2	14	280	1
NSP=3	2	1	175

下面再做最近邻方法的10折交叉验证, 还是用$k = 6$, 这里仅仅给出测试集的分类平均误判率. 相关的代码如下:

```
library(kknn)
E=rep(0,Z)
for(i in 1:Z){m=mm[[i]]
n1=length(m)
a=kknn(NSP~., k=6,train= w[-m,],test=w[m,])#KKNN
E[i]=sum(w[m,D]!=a$fit)/n1}
mean(E)
```

计算结果表明, 10折交叉验证得到的用最近邻方法对例5.1数据做分类的测试集的平均误判率为0.02774942. 还容易得到, 如果$k = 2$(这是拟合全部数据最好的k), 交叉验证测试集的误差将是0.03335242, 比$k = 6$时要大.

5.8 神经网络分类

5.8.1 拟合

前面已经介绍过神经网络回归, 神经网络分类的原理和回归类似, 这里不再赘述. 这里用的是程序包nnet[1]中的函数nnet().

还是利用前面3.7.1节的方法来选择函数nnet()中的隐藏层节点数等参数, 然后用这些参数来拟合. 对于拟合全部数据, 代码如下:

```
library(nnet)
a=nnet(NSP~., data=w, subset=1:n,
    size = 2,rang = 0.01,decay=5e-4,maxit = 200)
wp=predict(a, w[m,], type = "class")
```

[1]Venables, W. N. & Ripley, B. D. (2002) *Modern Applied Statistics with S.* Fourth Edition. Springer, New York. ISBN 0-387-95457-0.

```
sum(w[m,D]!=wp)/length(m)
```

得到误判率为0.2180095. 如果用size=7, 则误判率为0.004739336. 看上去后者要好, 但在10折交叉验证时结果就不那么理想了. 下面用同样的参数做10折交叉验证:

```
library(nnet)
E=rep(0,Z)
for(i in 1:Z)
{m=mm[[i]];mc=setdiff(1:n,m)
a=nnet(NSP~., data = w, subset = mc, size = 2,
    rang = 0.1,decay = 0.01, maxit = 200)
E[i]=sum(w[m,D]!=predict(a, w[m,], type = "class"))/length(m)}
mean(E)
```

得到测试集误判率为0.01365323. 但如果用前面的size=7, 则误判率为0.01789647.

5.8.2 神经网络对于不同参数的拟合效果分析

我们可以考察一下隐藏层节点个数(number of units in the hidden layer)对于训练集和测试集的误判率的影响. 我们固定参数delay=0.01, 然后变化size(即隐藏层节点个数, 从1到20个), 最后得到图5.5中的训练集和测试集的误差变化图. 这里把数据随机分成两组, 代码为:

```
set.seed(1010);d1=sample(1:n,n/2);d2=setdiff(1:n,d1)
ww=matrix(0,20,2);for(s in 1:20)
{a=nnet(NSP~., data = w, subset = d1, size = s,
    rang = 0.1,decay = 0.01, maxit = 200)
ww[s,1]=sum(w[d1,D]!=predict(a,w[d1,],
    type="class"))/length(d1)
ww[s,2]=sum(w[d2,D]!=predict(a,w[d2,],
    type="class"))/length(d2)}
matplot(1:20,ww,type="o",lty=2:1,pch=c(16,18),
    ylab="Error rate",xlab="Size")
legend("topleft",c("Training set","Testing sets"),
    pch=c(16,18),lty=2:1)
title("Error rate for different size of hidden layer in nnet")
```

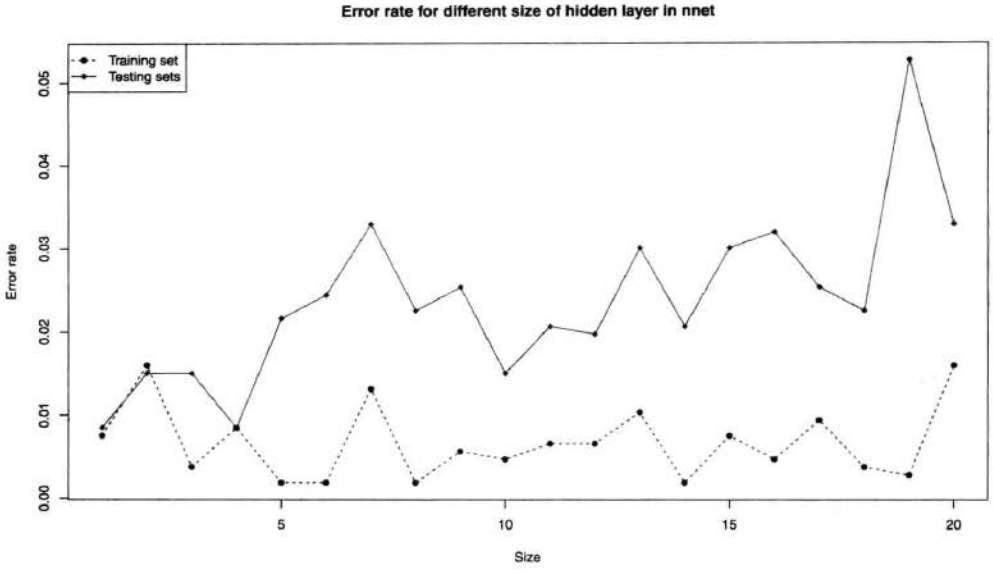


图 5.5 用不同隐藏层节点个数拟合例5.1数据时的训练集及测试集的误判率

从图5.5可以看出训练集和测试集的误差的差距. 实际上, 这些误判率随着训练集和测试集选取的不同而变化. 而且隐藏层节点个数仅仅是一个参数, 如果其他参数变化, 则情况更加复杂.

5.9 分类方法10折交叉验证结果汇总

前面利用4.2.2节的函数`Fold()`产生了例5.1数据的10个交叉验证子集, 并用本章的各种方法基于这些子集做了10折交叉验证. 交叉验证结果展示在图5.6中, 该图所依赖的数据在表5.9中. 图5.6中对应于每一种方法的条高是测试集误判率. 显然, 测试集误判率最低的是adaboost和随机森林, 最高的是最近邻方法, 这也可以从表5.9中看出. 利用测试集的误判率来比较模型的好坏是交叉验证的核心思想.

读者很可能会想, 可不可以对这个数据使用前面判别分析中提到的函数`lda()`和`mda()`即经典的判别分析方法来分类呢? 答案是这些方法不适用于这个数据, 原因是自变量中含有两个定性变量(Tendency和CLASS), 其水平分别为3个和10个, 这是经典判别分析所无法应对的. 这也说明了经典方法的局限性. 实际上, 如果自变量的定性变量再增加一些, 即使是一般的支持向量机算法也不能应对. 这里选择这个数据而不是更加复杂的数据, 就是因为要把各

种方法做一比较, 如果某些算法不能应对, 就无法参与比较了.

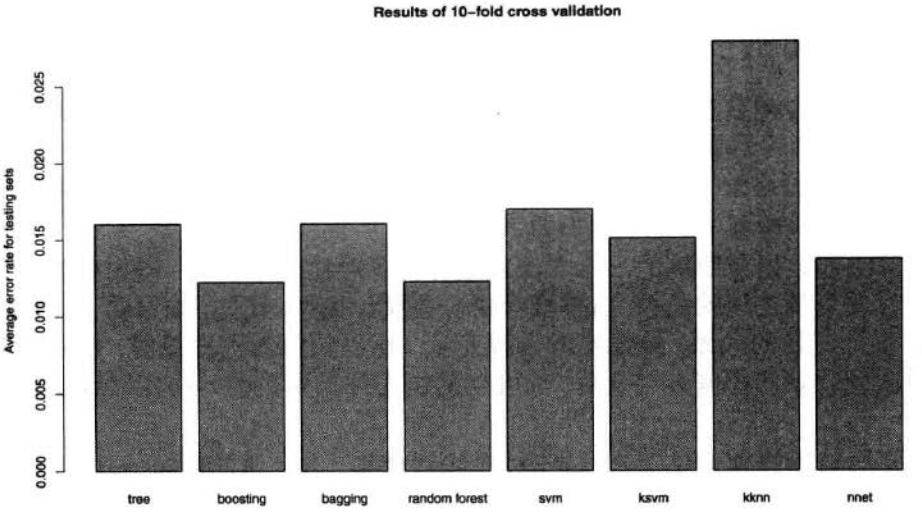


图 5.6 八种方法拟合例5.1数据时的10折交叉验证测试集的误判率

表 5.9 例5.1数据的八种方法的10个交叉验证测试集误判率

	测试集误差率
决策树	0.0160118
boosting	0.0122314
bagging	0.0160118
随机森林	0.0122447
支持向量机(svm)	0.0169508
支持向量机(ksvm)	0.0150617
kknn	0.0277494
nnet	0.0136532

第六章　横截面数据: 计数或有序因变量

6.1　概要和例子

还有一种与频数有关的数据, 比如来自抽样调查的数据, 那里变量大多是分类变量, 最终的结果可以用列联表形式展示, 列联表每个格子中的数字就是相应变量组合的频数(即计数). 因此, 频数虽然不是抽样调查时的变量, 但是在人们关心这些频数时, 就成了我们研究的对象.

应对频数数据最简单的汇总方法就是计算各种比例或频率. 但如果想研究频数和有关变量之间的关系, 就需要考虑一些统计模型, 比如, 如果要了解这个频数和各个变量的关系, 频数就成为因变量. 在以频数为因变量时, 自变量可以是组成列联表的那些分类变量, 也可以是作为背景资料的一些定性或定量变量.

涉及频数的经典方法是Poisson对数线性模型, 但经典的Poisson模型有很多弱点, 由此衍生了许多方法. 此外, 还有多项分布对数线性模型和多项logit模型. 当然, 机器学习方法也可以应用于频数问题.

例 6.1　血友病数据(hemophilia.csv)　这个数据[1]是由美国国家癌症研究所资助的多中心血友病队列研究(multicenter hemophilia cohort study, MHCS)获得的. 该项研究从1978年1月1日到1995年12月31日在16个治疗中心(12个在美国, 4个在西欧)跟踪了超过1600个血友病人, 该数据一共有2144个观测值及6个变量, 表6.1为变量情况.

表 6.1　例6.1血友病数据变量情况

变量名	描述	性质
hiv	患者的HIV状况(1=阴性, 2=阳性)	哑元型分类变量
factor	使用凝血因子制剂的5种剂量	哑元型分类变量
year	日历年	整数(也可作为分类变量)
age	年龄(按5岁递增的组)	整数/定序/分类
py	人年:该年该组参加该研究的时间总量	数量变量
deaths	该组死亡人数	整数

除了下面要做的以死亡数作为因变量的回归之外, 可以对这个数据做多方面的研究, 例如一些经典统计的检验, 请参看Nolan and Speed (2000).

[1]来自加州大学伯克利分校的STAT LABS网站: http://www.stat.berkeley.edu/users/statlabs/labs.html.

6.2 经典的Poisson 对数线性模型回顾

前面在讲logistic和probit回归时介绍了广义线性模型, 也提到了Poisson分布的典则连接函数为对数函数, 这时的线性模型称为Poisson对数线性模型, 其一般形式为:

$$\ln(\lambda) = \boldsymbol{X}^{\top}\boldsymbol{\beta},$$

式中, λ为Poisson分布的均值(也就是方差); $\boldsymbol{X}$为自变量向量; $\boldsymbol{\beta}$为系数向量. 一般定义中的Poisson分布的参数λ是个常数, 但在实践中, 这个参数很可能是一些变量的函数, 因此就产生了Poisson对数线性模型的概念, 这与logistic模型对二项分布参数p的建模思想是类似的. 正如前面提到的, Poisson对数线性模型与logistic回归及probit回归都属于广义线性模型的特例. 拟合数据时用的也都是`glm()`函数, 仅仅是选项不同而已.

一般来说, 因变量为计数的数据往往被联系到Poisson对数线性模型, 因为因变量取值为非负整数, 与Poisson分布一样. 如果因变量为一些相对稀有事件的数目, 那也和一般Poisson分布的例子有关. 由于Poisson对数线性模型比较容易理解, 结果也容易解释, 人们对其有比较充分的研究. 这里就用血友病的例子, 介绍如何用Poisson对数线性模型拟合数据. 当然, 这个模型不一定合适, 但的确是经典统计对于因变量为计数或频数时回归的选择之一.

先用下面语句输入例6.1的血友病数据, 并把用哑元代表的定性变量(hiv, factor)用函数`factor()`标识出来.

```
w=read.csv("hemophilia.csv")
for(i in 1:2)w[,i]=factor(w[,i])
```

这里准备用10折交叉验证来比较各种模型的测试集和训练集的NMSE. 希望在自变量中的10折数据分配尽可能均匀, 我们来考察从数据中得到的两个分类变量factor及hiv的频数表(见表6.2).

表 6.2 factor及hiv的频数

hiv	factor				
	1	2	3	4	5
1	213	208	227	248	249
2	208	199	195	218	179

从表6.2可以看出, 对于hiv来说factor的分布还是比较均匀的, 均匀本身可能是试验设计者的初衷. 这里把整个数据随机分成10份, 试图使得每一份对factor的每个水平来说都是较均匀的. 我们还是利用4.2.2节的函数`Fold()`来

随机选出10个交叉验证集, 虽然函数Fold()仅仅以一个分类变量(这里用变量factor)各个水平被平衡划分为基础, 但由于数据的平衡性, 另一个分类变量(hiv)也会被较平衡地划分:

```
#把第2个变量factor数据按照水平逐个随机选取10份
H=2;Z=10;n=nrow(w)
mm=Fold(Z,w,H,8888)
D=6
```

上面的H=2表示挑选序号为2的分类变量(factor)作为平均划分水平的基础, w为数据集, Z=10为折数, D=6表示因变量的位置, 由函数Fold()得到的10个交叉验证集的下标存在list格式的R对象mm之中, 即mm[[i]]代表第i个集合的下标集($i = 1, 2, ..., 10$). 下面说明如何通过R用Poisson对数线性模型来拟合数据. 使用广义线性模型函数glm()通过Poisson对数线性模型拟合例6.1的全部数据, 代码如下(注意这里的family="poisson"):

```
ap<-glm(deaths ~., family="poisson", data=w)
summary(ap)
AIC(ap)#1725.654
```

Poisson回归拟合例6.1数据后输出的系数及关于系数的近似正态性检验的p值列在表6.3中. 如果令α_i ($i = 1, 2$)代表hiv的两个水平, β_j ($j = 1, 2, ..., 5$)代表factor的5个水平, x_1代表year(θ_1代表year的系数), x_2代表age(θ_2代表age的系数), x_3代表py(θ_3代表py的系数), θ_0代表截距, 那么, 这个模型为:

$$\ln(\lambda) = \theta_0 + \alpha_i + \beta_j + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3.$$

根据计算机输出, 其中估计的参数值为:

$$\hat{\theta}_0 = -23.135,\ \hat{\alpha}_1 = 0,\ \hat{\alpha}_2 = 2.767,\ \hat{\beta}_1 = 0,\ \hat{\beta}_2 = -0.636,\ \hat{\beta}_3 = -0.403,$$
$$\hat{\beta}_4 = -0.708,\ \hat{\beta}_5 = -0.372,\ \hat{\theta}_1 = 0.211,\ \hat{\theta}_2 = 0.078,\ \hat{\theta}_3 = 0.033.$$

注意, 和线性模型一样, 定性自变量的各个水平的单独效应是不可估计的, 必须加上约束条件, 这里的约束条件是每个定性变量第一个水平为0. 对于我们的例子, 可以注意到, 那些与定性变量有关的效应α_1(hiv1)及β_1(factor1)按照R的默认约束条件都等于0.

类似于线性逐步回归, 还可以使用程序包MASS的函数stepAIC()来做逐步变量筛选. 对于这个数据, 做逐步回归不会产生不同的结果. 我们就不做了.

下面再用10折交叉验证来看一下对测试集的拟合结果, 代码如下:

表 6.3 例6.1数据死亡数目的分布

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-23.1353	1.3187	-17.54	0.0000
hiv2	2.7665	0.2033	13.61	0.0000
factor2	-0.6364	0.1519	-4.19	0.0000
factor3	-0.4034	0.1405	-2.87	0.0041
factor4	-0.7075	0.1427	-4.96	0.0000
factor5	-0.3716	0.1462	-2.54	0.0111
year	0.2110	0.0141	14.98	0.0000
age	0.0779	0.0155	5.03	0.0000
py	0.0330	0.0028	11.61	0.0000

```
NMSE=rep(0,Z)
for(i in 1:Z){m=mm[[i]]
a=glm(ff, family="poisson", data=w[-m,])
y1=predict(a,w[m,],type="response")
NMSE[i]=mean((w[m,D]-y1)^2)/mean((w[m,D]-mean(w[m,6]))^2)}
(MNMSE=mean(NMSE))
```

得到测试集的NMSE为0.7391105, 似乎还可以.

下面用"不那么适合的"简单线性回归来试试(在科学上, 好奇总是没有坏处的), 用最小二乘回归模型`lm(deaths~.,w)`来拟合数据, 并做交叉验证(这里不给出代码), 得到测试集的NMSE为0.807315. 因此不能用简单线性回归.

另外, 在Poisson回归时, 人们首先要考虑关于Poisson分布的基本条件是否满足. 也就是说, 其均值和方差是否相等. 在统计术语中, 方差不等于均值的现象称为散布(dispersion), 如果方差大于均值, 称为过散布(overdispersion), 否则称为欠散布(underdispersion). 此外, 还可能有其他一些原因造成Poisson对数线性模型的失败. 下一节将介绍一些可用于尝试解决这种问题的方法. 无论我们的数据是否会有这些问题, 都应该予以注意.

6.3 使用Poisson对数线性模型时的散布问题

使用Poisson对数线性模型时出现的散布问题的一种解决方法是双广义线性模型(double generalized linear model), 其方法体现在R的程序包`dglm`[1]之

[1]Peter K. Dunn and Gordon K. Smyth (2009). dglm: Double generalized linear models. R package version 1.6.1.

中. 其基本思想如下.

假定均值和方差有如下关系:

$$\mu_i = E(y_i),\ Var(y_i) = s_i V(\mu_i).$$

式中, s_i代表散布程度, 称为散布参数(dispersion parameter), 以$s_i = 1$为没有散布, 而$V(\mu_i)$为μ_i的一个函数(对没有散布时的Poisson分布, $V(\mu) = \mu$). 建模时对均值和散布参数分别建立广义线性模型, 连接函数分别是

$$g(\mu_i) = \boldsymbol{x}_i^\top \boldsymbol{\beta} \quad 与 \quad h(s_i) = \boldsymbol{z}_i^\top \boldsymbol{\alpha},$$

式中, 自变量$\boldsymbol{x}_i$及$\boldsymbol{z}_i$都选自原始的自变量. 对于Poisson分布(或者是下面要用的使用更加广泛的Tweedie分布)的普通广义线性模型仅有前面那个连接, 而这里对s_i建模时考虑的是Gamma模型, 即第二个连接. 关于双广义线性模型的细节, 请参看Smyth(1989).

下面介绍可代表很多广义线性模型分布的Tweedie分布. Tweedie分布是指数族分布的一个特例. 前面4.1.1节在介绍广义线性模型时已经介绍过指数族分布了, 那里均值μ是包含在θ之中的. 显性表示μ的指数族分布可以写成下面的形式:

$$f(y;\mu,\phi) = a(y,\phi)\exp\frac{\theta(\mu)y - \kappa(\theta(\mu))}{\phi}.$$

Tweedie分布则为上面指数族中有下面形式的$\theta(\mu)$和$\kappa(\theta(\mu))$的特殊形式:

$$\theta(\mu) = \begin{cases} \frac{\mu^{1-p}-1}{1-p}, & p \neq 1 \\ \ln\mu, & p = 1 \end{cases} \quad 以及\ \kappa(\theta(\mu)) = \begin{cases} \frac{\mu^{2-p}-1}{2-p}, & p \neq 2 \\ \ln\mu, & p = 2 \end{cases}.$$

其均值为μ, 方差为$\phi\mu^p$. 式中, $\phi > 0$, 为散布参数; p称为指标参数(index parameter). 指标参数唯一地确定Tweedie分布的具体成员, 比如$p = 0$代表正态分布; $p = 1, \phi = 1$代表Poisson分布; $p = 2$代表Gamma分布; $p = 3$代表逆高斯分布. R程序包`statmod`[1]中有`tweedie()`函数.

我们通过Tweedie分布用双广义线性模型拟合的代码如下:

```
library(dglm)
at=dglm(deaths ~hiv+factor+age +py,~hiv+factor+age +py,
    family=tweedie(var.power=1,link.power=0.2), data=w)
summary(at)
summary(at$dispersion.fit)
```

[1]Gordon Smyth with contributions from Yifang Hu, Peter Dunn and Belinda Phipson (2013). statmod: Statistical Modeling. R package version 1.4.17. http://CRAN.R-project.org/package=statmod.

可以看出, 这里的两个模型都用了同样的自变量. 所用的函数在程序包dglm中. 这里计算时根据Tweedie分布的广义线性模型假定

$$\mu_i^q = \boldsymbol{x}_i^\top \boldsymbol{\beta},\ Var(y_i) = \phi \mu_i^p,$$

式中, $\boldsymbol{x}_i^\top \boldsymbol{\beta}$为线性表示部分的协变量$\boldsymbol{x}_i$和系数$\boldsymbol{\beta}$; 而$q$为连接函数的幂(代码中link.power); p确定了指数族成员(代码中var.power). 没有AIC, 为了比较, 用10折交叉验证来看结果, 代码如下:

```
library(dglm)
NMSE=rep(0,Z)
for(i in 1:Z){
m=mm[[i]]
a=dglm(deaths ~hiv+factor+age+py,~hiv+factor+age+py,
    family=tweedie(var.power=1,link.power=0.2), data=w[-m,])
y1=predict(a,w[m,],type="response")
NMSE[i]=mean((w[m,D]-y1)^2)/mean((w[m,D]-mean(w[m,D]))^2)}
(MNMSE=mean(NMSE))
```

得到测试集的NMSE为0.9135642. 这个结果不如Poisson线性模型好.

当然还有其他对付散布的方法, Breslow (1984)建议用一种迭代方法拟合过散布的Poisson对数线性模型. 其模型和上面的双广义线性模型有所区别, 其中

$$E(y_i) = \mu_i \text{ 而且 } Var(y_i) = \mu_i + \mu_i^2 \phi = \mu_i(1 + \mu_i \phi),$$

式中, ϕ也称为散布参数, 如果$\phi > 0$, 就意味着有散布现象. 这里不对这个模型予以计算. 请读者自己尝试运行程序包dispmod[1]中的例子.

此外, 对付散布问题, 在程序包MASS中还有用负二项分布拟合这类数据的广义线性模型函数glm.nb(), 对我们的数据做交叉验证的代码如下:

```
library(MASS)
NMSE=rep(0,Z)
for(i in 1:Z){
m=mm[[i]]
a=glm.nb(deaths ~.,data=w[-m,])
y1=predict(a,w[m,],type="response")
NMSE[i]=mean((w[m,D]-y1)^2)/mean((w[m,D]-mean(w[m,D]))^2)}
(MNMSE=mean(NMSE))
```

[1]Luca Scrucca (2009). dispmod: Dispersion models. R package version 1.0.1.

得到10折交叉验证测试集的NMSE为0.7524876. 比Poisson对数线性模型稍差.

6.4 零膨胀时的Poisson回归

例6.1数据还有其他一些特征, 让我们看一下死亡数目的分布(用代码`table(w$deaths)`来展示), 该分布展示在表6.4及图6.1中(用代码`barplot(table(w$deaths))`来展示).

表 6.4 例6.1数据死亡数目的分布

死亡人数	0	1	2	3	4	5	6
频数	1833	212	62	28	6	2	1

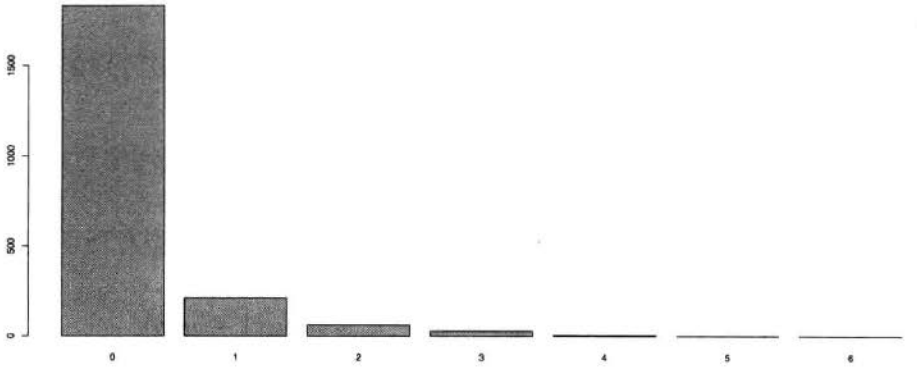


图 6.1 例6.1数据的死亡数目的条形图

由表6.4及图6.1可以看出, 死亡数等于零的组占了整个数据的绝大部分. 这种不平衡的计数数据称为零膨胀计数数据(zero-inflated count data). 这里介绍一种处理这个问题的方法.

零膨胀计数数据模型(Mullahy 1986; Lambert 1992)由两个部分组成, 一部分为集中在零点的点质量(如logistic或probit回归模型); 另一部分为某计数分布(比如Poisson或负二项回归模型). 如果用$f(y)$表示分布密度, $\pi(0)$表示在零点的点密度, $1-\pi(0)$表示在其他点的点密度, $f_c(y)$表示在其他点的计数分布, 则零膨胀密度为:

$$f(y) = \pi(0)I_{\{0\}}(y) + [1-\pi(0)]f_c(y),$$

这里$I_{\{0\}}(y)$是在零点的示性函数. 关于零点的点质量可以选与以$\pi \equiv \pi(0)$(或者$1-\pi(0)$)为概率的二项分布相关的logistic回归(也可以选其他的, 比

如probit回归):

$$\ln \frac{\pi}{1-\pi} = \boldsymbol{x}^\top \boldsymbol{\beta} \quad 或者 \quad \pi = \frac{\exp(\boldsymbol{x}^\top \boldsymbol{\beta})}{1+\exp(\boldsymbol{x}^\top \boldsymbol{\beta})}.$$

而在非零的地方则可选用Poisson对数线性模型(或其他模型, 如负二项模型等):

$$\ln(\lambda) = \boldsymbol{x}^\top \boldsymbol{\gamma} \quad 或者 \quad \lambda = \exp(\boldsymbol{x}^\top \boldsymbol{\gamma}).$$

而整个模型的均值μ应该是

$$\mu = \pi \cdot 0 + (1-\pi)\lambda.$$

用这个模型估计出来的参数也应该由两部分组成: logistic回归部分的$\hat{\boldsymbol{\beta}}$及Poisson模型的$\hat{\boldsymbol{\gamma}}$.

这里利用程序包pscl[1]中的zeroinfl()函数, 我们这里对于两部分的模型代码分别为deaths~hiv+factor+py+age及~hiv+py+age. 实际上, 每一部分用什么自变量完全通过试验来确定, 这与数据性质有关, 无法先验确定. 使用该模型拟合全部数据(整个数据都是训练集)的具体计算代码如下:

```
library(pscl)
az<-zeroinfl(deaths~hiv+factor+py+age|hiv+py+age,w)
summary(az)
```

可以得到两部分模型的拟合结果:

```
Call:
zeroinfl(formula = deaths ~ hiv + factor + py + age | hiv +
    py + age, data = w)

Pearson residuals:
     Min       1Q   Median       3Q      Max
-1.14319 -0.42659 -0.18263 -0.03452 40.55835

Count model coefficients (poisson with log link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.420939   0.457404  -5.293 1.20e-07 ***
hiv2         2.143870   0.308924   6.940 3.93e-12 ***
factor2     -0.639624   0.157413  -4.063 4.84e-05 ***
```

[1]Simon Jackman (2011). pscl: Classes and Methods for R Developed in the Political Science Computational Laboratory, Stanford University. Department of Political Science, Stanford University. Stanford, California. R package version 1.04.1. URL http://pscl.stanford.edu/.

```
factor3      -0.402610   0.146170  -2.754  0.00588 **
factor4      -0.577092   0.147251  -3.919 8.89e-05 ***
factor5      -0.414416   0.151140  -2.742  0.00611 **
py            0.028751   0.003557   8.083 6.32e-16 ***
age          -0.035519   0.024899  -1.427  0.15372

Zero-inflation model coefficients (binomial with logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  6.31058    1.03603   6.091 1.12e-09 ***
hiv2        -3.25043    0.83818  -3.878 0.000105 ***
py           0.02750    0.01138   2.416 0.015703 *
age         -0.86075    0.16164  -5.325 1.01e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of iterations in BFGS optimization: 40
Log-likelihood: -928.1 on 12 Df
```

可以通过AIC(az)得到该拟合的AIC值=1880.295. 按照输出, 前面模型中的logistic回归部分的拟合模型应该是

$$\ln \frac{\pi}{1-\pi} = 6.311 - 3.250\text{hiv2} + 0.028\text{py} - 0.861\text{age},$$

Poisson对数线性模型部分的拟合模型为:

$$\ln(\lambda) = -2.421 + \alpha_i + \beta_j + 0.029\text{py} - 0.036\text{age},$$

这里α_i代表hiv $= i$ $(i = 1, 2)$对截距的影响, 估计结果为: $\hat{\alpha}_1 = 0$(软件默认值), $\hat{\alpha}_2 = 2.144$; β_j代表factor $= j$ $(j = 1, 2, ..., 5)$对截距的影响, 估计结果为: $\hat{\beta}_1 = 0$(软件默认值), $\hat{\beta}_2 = -0.640$, $\hat{\beta}_3 = -0.403$, $\hat{\beta}_4 = -0.577$, $\hat{\beta}_5 = -0.414$. 这些结果和前面各种模型的拟合结果很像, 相信读者可以轻松地理解. 此外, 再对其进行10折交叉验证, 这里对于logistic模型的变量做了选取, 以使得结果更好, 代码如下:

```
NMSE=rep(0,Z)
for(i in 1:Z){m=mm[[i]]
a=zeroinfl(deaths~.|factor+age+py,data=w[-m,],dist = "negbin")
y1=predict(a,w[m,])
NMSE[i]=mean((w[m,D]-y1)^2)/mean((w[m,D]-mean(w[m,D]))^2)}
(MNMSE=mean(NMSE))
```

得到测试集的NMSE为0.6680084. 这个结果比前面的各个结果都好, 但是在十次迭代的拟合一步中有四次计算机给出了警告, 希望这些警告对预测结果不会造成太大影响.

上述拟合都是以经典统计模型为基础的探讨, 下面尝试用算法模型来拟合这个数据, 欲采用的方法包括随机森林、决策树、支持向量机.

6.5 用机器学习的算法模型拟合计数因变量数据

把用于回归的算法模型用来拟合计数数据显然不太符合经典的数学模型驱动的思维方式. 当然, 在Poisson回归之前, 都是用线性模型来做的. 从实际工作预测的角度来看, 只要预测得好, 容易解释, 就是好模型. 实际数据都是复杂的, 一些简单的线性或非线性数学公式不可能涵盖所有可能的问题. 这里的代码完全类似于前面已经使用过的随机森林、决策树和支持向量机进行10折交叉验证的代码, 因此, 就不做详细说明了.

6.5.1 随机森林拟合例6.1数据的10折交叉验证

```
library(randomForest)
set.seed(1010)
NMSE=rep(0,Z)
for(i in 1:Z){
m=mm[[i]]
A=randomForest(deaths ~ ., data=w[-m,],
    importance=TRUE,proximity=TRUE)
y1=predict(A,w[m,])
NMSE[i]=mean((w[m,D]-y1)^2)/mean((w[m,D]-mean(w[m,D]))^2)}
(MNMSE=mean(NMSE)) # 0.651716
```

6.5.2 决策树(回归树)拟合例6.1数据的10折交叉验证

```
library(rpart)
NMSE=rep(0,Z)
for(i in 1:Z){
m=mm[[i]]
a=rpart(deaths ~ ., data=w[-m,])
y1=predict(a,w[m,])
NMSE[i]=mean((w[m,D]-y1)^2)/mean((w[m,D]-mean(w[m,D]))^2)}
(MNMSE=mean(NMSE))#0.730649
```

6.5.3 支持向量机拟合例6.1数据的10折交叉验证

```
library(rminer)
set.seed(1010)
NMSE=rep(0,Z)
for(i in 1:Z){
m=mm[[i]]
M=fit(deaths~.,w[-m,],model="svm")
y1=predict(M,w[m,])
NMSE[i]=mean((w[m,6]-y1)^2)/mean((w[m,6]-mean(w[m,6]))^2)}
(MNMSE=mean(NMSE))#0.6826057
```

6.5.4 各种方法拟合例6.1数据的10折交叉验证结果比较和一些讨论

1. 各种方法拟合例6.1数据的10折交叉验证结果比较

这些算法模型以及前面提到过的经典模型的10折交叉验证结果列在表6.5中. 图6.2是表6.5的直观显示.

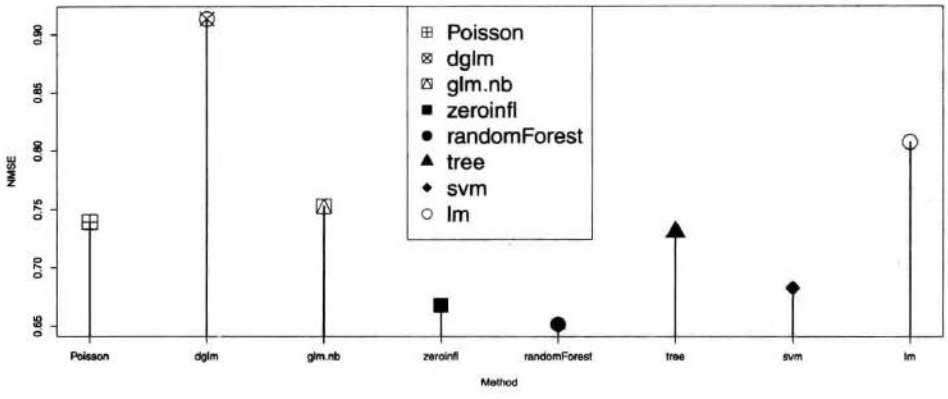


图 6.2 例6.1数据的各种方法标准化均方误差(NMSE)的比较

我们在这里所关注的是测试集的标准化均方误差(NMSE). 显然, 对于这个数据, 经典的计数模型中最好的为零膨胀计数回归; 在算法模型中, 最差的是决策树, 最好的是随机森林, 之间的是支持向量机. 但总体来说, 按照NMSE从优到劣排序为: 随机森林、零膨胀计数回归、支持向量机、决策树、Poisson对数线性模型、负二项线性模型、简单线性回归、双广义线性模型.

表 6.5 各种方法关于测试集10折交叉验证的NMSE

回归方法	测试集NMSE
Poisson对数线性模型	0.739111
双广义线性模型	0.913564
负二项线性模型	0.752488
零膨胀计数回归	0.668008
随机森林	0.651716
决策树(回归树)	0.730649
支持向量机	0.682606
简单线性回归	0.807315

2. 关于模型驱动还是数据驱动的简单讨论

从对例6.1数据的10折交叉验证结果来看, 专门为计数设计的若干经典模型竟然不如广谱的算法模型. 如何看待这个问题呢? 经典模型通常需要对数据做许多假定, 如果数据满足这些假定, 则经典模型很完美, 数学上是被精确描述的, 而且很容易根据模型的分析表达式对数据及结果作出解释.

另外, 在经典模型中, 人们追求的是漂亮模型的精确解, 所有结论都是在"模型是正确的"这一前提下得到的, 比如在对数据及模型的严格假定下的相合性和无偏性. 但所有的模型都是对真实世界的猜想, 因此, 误差就由两部分组成: 一部分是模型与产生实际数据的规律的差距; 另一部分是根据模型所作预测或估计的误差, 体现在各种检验以及无偏性与相合性等概念上. 经典统计研究的相当大部分的注意力都集中在后者, 所有的检验所反映的显著性以及无偏性和相合性都必须在严格的假定之下才有意义. 然而, 对于模型与实际规律的差距很难作出分析上的量化和评估, 这也是人们有意无意回避这个问题的原因.

机器学习的算法模型由于没有对数据作出任何假定, 因此不会得到类似p值那样的显著性度量, 更不用诸如相合性及无偏性等概念来评价模型, 算法模型在评估模型时用的是容易被外行接受的交叉验证方法. 这些方法的优点是预测精度高, 缺点是无法用精确的数学公式来描述.

经典模型就像一个假想的为所有动物服务的服装店, 它只有有限的几个精确尺寸, 却要满足地球上大约150万种动物的要求, 其难度可想而知. 算法模型则试图为每组数据量体裁衣, 虽然这些模型也有局限性, 但和经典分析模型相比要灵活得多.

统计学家John Tukey说过: "一个**正确**问题的近似答案远远好于一个错误问题的**精确**答案, 尽管对正确问题的叙述往往有些含糊, 而对错误问题的描述

却总是精确的."[1] (Tukey,1962) 这准确地表明了他对问题驱动研究及模型驱动研究的态度.

6.6 多项logit模型及多项分布对数线性模型回顾

为了方便读者, 笔者特地在本节准备了下面多项分布数据的两个回归模型的简单回顾. 其中, 多项logit模型是二项分布的logistic回归向多项分布的推广; 多项分布对数线性模型是基于高维列联表数据分析而发展起来的.

6.6.1 多项logit模型回顾

前面介绍过在因变量为二水平分类变量时使用logistic回归模型, 那是与二项分布有关的. 当分类变量有两个以上的水平且这些水平为仅有的可能时, 可以考虑多项logit模型(multinomial-logit). 注意, 这里与一般分类方法的区别为因变量没有所列水平之外的可能. 例如, 当因变量为旅行方式时, 如果水平为汽车、火车、轮船、飞机, 那么就是一般的分类问题, 不能用多项logit回归, 因为还有其他选择, 比如步行、骑自行车、驾驶帆船、骑马、滑雪等. 但是如果因变量水平为汽车、火车、轮船、飞机、其他, 就可以用多项logit回归了.

假定对于第i个观测, 因变量Y_i有M个取值: $1, 2, ..., M$, 自变量为X_i, 则多项logit回归的模型为:

$$P(Y_i = k) = \frac{\exp(X_i\beta_k)}{1 + \sum_{j=2}^{M} \exp(X_i\beta_j)}, \quad k = 2, 3, ..., M.$$

k为什么从2开始呢? 因为

$$P(Y_i = 1) = \frac{1}{1 + \sum_{j=2}^{M} \exp(X_i\beta_j)},$$

可以从约束条件$P(Y_i = 1) = 1 - \sum_{j=2}^{M} P(Y_i = j)$得到. 当然也可以考虑等价模型:

$$P(Y_i = k) = \frac{\exp(X_i\beta_k)}{\sum_{j=1}^{M} \exp(X_i\beta_j)}, \quad k = 1, 2, ..., M.$$

多项logit模型显然是logistic模型的推广, 或者说logistic回归是多项logit模型的特例. 下面用一个选择旅行方式的例子来介绍计算过程.

[1] 原文为: Far better an approximate answer to the *right* question, which is often vague, than an *exact* answer to the wrong question, which can always be made precise.

例 6.2 旅行模式数据(TravelMode0.csv, TravelMode.csv) 这个数据[1]来自Greene, W.H. (2003), TravelMode0.csv是程序包**Ecdat**[2]带有的数据; TravelMode.csv是程序包**AER**[3]中带有的数据. 该数据是关于在澳大利亚悉尼和墨尔本之间旅行模式的选择, 一共有840个观测值. 这两个数据是一样的, 但后面一个是前面一个的加工结果.[4] 变量情况见表6.6

表 6.6 例6.2旅行模式数据变量情况(带*号的是第二个数据增加的)

变量名	描述	性质	变量名	描述	性质
individual*	家庭编号	分类变量	travel	汽车内耗时	数量变量
mode	旅行模式(2/4*种)	分类变量	gcost	广义花费度量	数量变量
choice*	选择(no,yes)	分类变量	income	家庭收入	数量变量
wait	等待时间	数量变量	size	旅行人数	整数
vcost	车辆花费	数量变量			

这两个数据每四行代表一个家庭, TravelMode0.csv没有家庭的编号, 实际上家庭编号是由下面代码计算出来的:

```
individual=rep(1:210,rep(4,210))
```

对于每个家庭的四行, 变量size和income是一样的(同一个家庭应该一样). 而TravelMode0.csv的变量mode为0和1组成的哑元(0相应于FALSE, 1相应于TRUE), 相应于该数据没有指出的4个旅行模式(air, train, bus, car, 这些为TravelMode.csv数据的变量mode所具有). 变量wait对于乘车来说为0. 本小节使用程序包**mlogit**.[5] 为了对比这两个数据, 可以运行下面代码:

```
library(mlogit)
v=read.csv("TravelMode.csv")
w=read.csv("TravelMode0.csv")
w1=mlogit.data(w, choice="mode",
    shape="long", alt.levels=c("air","train","bus","car"))
head(w1);head(v)
```

[1]来自网站http://pages.stern.nyu.edu/ wgreene/Text/tables/tablelist5.htm.

[2]Yves Croissant (2014). Ecdat: Data sets for econometrics. R package version 0.2-7. http://CRAN.R-project.org/package=Ecdat.

[3]Christian Kleiber and Achim Zeileis (2008). *Applied Econometrics with R*. New York: Springer-Verlag. ISBN 978-0-387-77316-2. URL http://CRAN.R-project.org/package=AER.

[4]本书第二版用的是后者, 现使用前者(但把变量名称改成与后者匹配), 因为前者更加原始, 使得大家更明白数据的构成.

[5]Yves Croissant (2011). mlogit: multinomial logit model. R package version 0.2-2. http://CRAN. R-project.org/package=mlogit.

可以看出, w1和v是完全对应的. 两个数据集的关系是:

- w1的行名字(row.names(w1))就是v的individual和mode的组合;
- 而w1的choice则隐含在mlogit格式的w1当中.

用多项logit模型拟合时用程序包mlogit的函数mlogit(). 我们这里对全体数据建模, 代码(与R中的例子相同)为:

```
a=mlogit(mode~gcost+wait|income, reflevel="car", data=w1)
summary(a)
```

输出为:

```
Call:
mlogit(formula = mode ~ gcost + wait | income, data = w1,
    reflevel = "car", method = "nr", print.level = 0)

Frequencies of alternatives:
    car     air   train     bus
0.28095 0.27619 0.30000 0.14286

nr method
5 iterations, 0h:0m:0s
g'(-H)^-1g = 0.000614
successive function values within tolerance limits

Coefficients :
                    Estimate Std. Error t-value  Pr(>|t|)
air:(intercept)    5.8747921  0.8020903  7.3244 2.400e-13
train:(intercept)  5.5498345  0.6404244  8.6659 < 2.2e-16
bus:(intercept)    4.1302566  0.6763628  6.1066 1.018e-09
gcost             -0.0109273  0.0045878 -2.3818   0.01723
wait              -0.0954602  0.0104732 -9.1147 < 2.2e-16
air:income        -0.0053735  0.0115294 -0.4661   0.64116
train:income      -0.0565616  0.0139733 -4.0478 5.170e-05
bus:income        -0.0285836  0.0154442 -1.8508   0.06420

Log-Likelihood: -189.53
McFadden R^2:  0.33209
Likelihood ratio test : chisq = 188.47 (p.value = < 2.22e-16)
```

从输出结果可以看出, 对于不同的模式, 自变量的系数不同. 令α_i $(i = 1, 2, 3, 4)$分别代表使用4种交通工具(car, air, train, bus)对截距的影响, β代表变量gcost的斜率, δ代表wait的斜率, $\alpha\gamma_i$代表4种交通工具对income的斜率(交叉效应). 拟合的模型为:

$$P(Y=i)=\frac{\exp(\hat{\alpha}_i+\hat{\beta}\text{gcost}+\hat{\delta}\text{wait}+\widehat{\alpha\gamma}_i\text{income})}{1+\sum_{i=1}^{4}[\exp(\hat{\alpha}_i+\hat{\beta}\text{gcost}+\hat{\delta}\text{wait}+\widehat{\alpha\gamma}_i\text{income})]},\quad i=1,2,3,4,$$

而$P(Y=1)=1-\sum_{i=2}^{4}P(Y=i)$, 这里参数的估计值为$\hat{\alpha}_1=0, \hat{\alpha}_2=5.875$, $\hat{\alpha}_3=5.550$, $\hat{\alpha}_4=4.130$, $\hat{\beta}=-0.011$, $\hat{\delta}=-0.095$, $\widehat{\alpha\gamma}_1=0, \widehat{\alpha\gamma}_2=-0.005$, $\widehat{\alpha\gamma}_3=-0.057$, $\widehat{\alpha\gamma}_4=-0.029$.

说明: 如果没有公式里面的"I"以及后面的变量`income`, 输出中就没有交叉效应. 选项`reflevel = "car"`意味着变量`car`是个参照, 因此默认的效应为0($\hat{\alpha}_1=0$), 它和`income`的默认交互效应也是0($\widehat{\alpha\gamma}_1=0$). 如果不写选项`reflevel = "car"`, 那么默认的参照变量就是`air`(因为它是第一个).

6.6.2 多项分布对数线性模型回顾

对于计数因变量, 前面引进了logistic回归、Poisson对数线性模型、多项logit回归等方法. 在列联表数据中, 每个格子中都是各种变量组合的计数. 假定有n个格子, 如果观测点落入每个格子都有一个概率p_i $(i=1,2,...,n)$, 那么可以用多项分布来描述这个问题. 注意, 这里有约束条件$\sum_i p_i=1$. 如果每个格子计数的均值都随着一些自变量的变化而变化, 则可以考虑多项分布对数线性模型. 一般来说, 假定对于第i个格子, μ_i为计数的均值, 自变量为X_i, 多项分布对数线性模型为把$\ln(\mu_i)$表达为自变量X_i的线性组合形式.

下面用两个例子来说明这种回归方法.

例 6.3 Titanic**数据**(Titanic.csv) 这是著名的泰坦尼克号的相关数据, 以4维数组(array)的形式在R中提供. 这是一个典型的4维列联表数据, 一共有2201个观测值(总计数), 4个变量, 具体情况见表6.7.

表 6.7 例6.3 Titanic数据变量情况

No	变量	水平
1	Class	1st, 2nd, 3rd, Crew
2	Sex	Male, Female
3	Age	Child, Adult
4	Survived	No, Yes

显然, 这个列联表为$4 \times 2 \times 2 \times 2$维的, 一共有32个格子. 由于都是分类变量, 因此相应的多项分布对数线性模型为:

$$\ln(\mu_{ij}) = \beta_0 + \sum_{i=1}^{4} \beta_{ij}, \quad i = 1,2,3,4, \ j = n_i \ (n_1 = 4, \ n_2 = n_3 = n_4 = 2).$$

也就是说, 对于每一种变量组合, $\ln(\mu_{ij})$都是以一个数目(截距)的方式表达的, 算上公共截距β_0, 一共有11个参数. 注意, 这些对应于分类变量各个水平的参数只在其所属变量各水平之间才有相对意义, 也就是说, 它们的估计是在一些约束条件之下做出的.

计算时, 用程序包MASS中的loglm()函数, 具体代码如下:

```
library(MASS)
a=loglm(~Class+Sex+Survived+Age,Titanic)
pchisq(a$pea,a$df,low=F) #得到Pearson检验p值
pchisq(a$lrt,a$df,low=F) #得到似然比检验p值
a$para #可得各个参数
```

其中被估计的参数, 即上面提到的11个参数β_0及β_{ij}估计值的输出如下:

```
$`(Intercept)`
[1] 3.015185

$Class
       1st        2nd        3rd       Crew
-0.4115541 -0.5428901  0.3642359  0.5902083

$Sex
      Male     Female
 0.6518609 -0.6518609

$Age
    Child     Adult
-1.477264  1.477264

$Survived
       No        Yes
 0.3699295 -0.3699295
```

读者很容易看出, 上面的估计值的约束为$\sum_{j=1}^{n_i} \hat{\beta}_{ij} = 0$.

下面的例子不是单纯的列联表数据, 有很多数量变量, 用不同程序包的函数来处理.

例 6.4 钓鱼数据(Fishing.csv) 这个数据来自Herriges and Kling (1999), 也是程序包mlogit中带有的数据. 该数据有1182个观测, 是关于美国钓鱼模式的, 有10个变量. 变量情况列在表6.8中, 这里模式(mode)为仅有的分类变量, 其4个水平为: beach(沙滩), pier(栈桥), boat(开船), charter(租船).

表 6.8 例6.4 钓鱼数据变量情况

变量名	描述	性质	变量名	描述	性质
mode	4种钓鱼方式	分类变量	catch.beach	沙滩捕捉率	数量变量
price.beach	沙滩花费	数量变量	catch.pier	栈桥捕捉率	数量变量
price.pier	栈桥花费	数量变量	catch.boat	开船捕捉率	数量变量
price.boat	开船花费	数量变量	catch.charter	租船捕捉率	数量变量
price.charter	租船花费	数量变量	income	月收入	数量变量

这里将捕鱼的4种模式看作捕鱼的所有可能选择. 如果把这9个自变量分别用$x_1, x_2, ..., x_9$代表, 用β_{ij}, $j = 0$及$1,2,3,4,5,6,7,8,9$分别表示相应于第i种钓鱼方式的截距及9个自变量的斜率($i = 2,3,4$), 则可以考虑多项分布对数线性模型(总计数假定为N)

$$\ln(\mu_i) = \beta_{i0} + \sum_{j=1}^{9} \beta_{ij} x_j, \quad i = 2,3,4, \quad \mu_1 = N - \sum_{i=2}^{4} \mu_i.$$

这里用程序包nnet的multinom()函数来拟合数据. 代码为:

```
data(Fishing,package="mlogit");library(nnet)
a= multinom(mode ~ ., data = Fishing)
summary(a)
```

输出为:

```
Call:
multinom(formula = mode ~ ., data = Fishing)

Coefficients:
        (Intercept)  price.beach   price.pier price.boat
pier      -6.252015 -0.002469672 -0.002469672 -0.2934729
boat     -13.127751  0.015698054  0.015698054 -0.5650396
charter  -15.581916  0.013459039  0.013459039 -0.6534132
```

```
        price.charter catch.beach catch.pier catch.boat
pier        0.2936538    34.71251  -48.15681 -1.4650480
boat        0.5422299    57.12954  -84.59076 -0.4383507
charter     0.6358946    58.83028  -84.82774  0.5562081
        catch.charter        income
pier       0.05204001 -8.987710e-05
boat       0.30151638  1.064670e-05
charter    0.09880167 -6.056782e-05

Std. Errors:
        (Intercept) price.beach  price.pier  price.boat
pier    0.000504553 0.001672146 0.001672146 0.008273766
boat    0.002612635 0.001693029 0.001693029 0.009458203
charter 0.002665084 0.001674973 0.001674973 0.009069802
        price.charter  catch.beach   catch.pier   catch.boat
pier      0.007445115 0.0001769799 0.0001480912 0.0002019135
boat      0.008891930 0.0024047937 0.0016138224 0.0116042051
charter   0.008525760 0.0024759877 0.0016492126 0.0118472123
        catch.charter       income
pier     0.0007322931 5.019554e-05
boat     0.0395411226 5.711061e-05
charter  0.0404433209 5.666854e-05

Residual Deviance: 2128.394
AIC: 2182.394
```

则上面输出的3×10系数矩阵为$\{\hat{\beta}_{ij}\}$ $(i = 2, 3, 4,\ j = 0, 1, ..., 9)$. 由于变量mode的4个水平之间不独立, 有约束条件, 因此上面拟合结果只涉及3个水平.

6.7 有序变量的比例优势模型

有很多变量的取值是有序的, 比如疾病的严重程度、某项产品的评级、对某项事务的评价, 等等. 这就导致了本节**比例优势模型**(proportional odds model)的产生. 下面引进一个例子.

例 6.5 世界价值数据(WVS.csv) 该数据为1995—1997年在澳大利亚、挪威、瑞典、美国做的世界价值调查数据(Fox and Andersen, 2006), 一

共有5381个观测值和6个变量, 变量情况见表6.9.

表 6.9 例6.5 世界价值数据变量情况

变量名	描述	性质
poverty	政府在贫困上的作为	有序变量: 3水平(Too Little, About Right, Too Much)
religion	是否信教	分类变量: 2水平(no, yes)
degree	有没有大学学位	分类变量: 2水平(no, yes)
country	国家	分类变量: 4水平(Australia, Norway, Sweden, USA)
age	年龄	数量变量
gender	性别	分类变量: 2水平(male, female)

程序包`effects`[1]以名称`WVS`提供这个数据. 该数据中的变量poverty就是一个有序变量, 我们把它作为因变量, 希望用$\boldsymbol{x}$的一个线性表示$\boldsymbol{\beta}^\top\boldsymbol{x}$来描述poverty. 有些类似于广义线性模型, 我们把与因变量有关的概率函数和$\boldsymbol{\beta}^\top\boldsymbol{x}$联系起来. 作为分类变量的poverty没有精确数量值, 但它是有序的, 有三个水平: Too Little, About Right, Too Much. 形式上, 我们可以记(这里借用符号"$\prec$"表示程度的大小, 不是严格数学意义上的不等号)

$$\text{Too Little} \prec \text{About Right} \prec \text{Too Much}.$$

如何对这种有序分类变量建模就是本节要讨论的问题.

假定有一个连续变量Z, 其精确取值不可观测, 只知道它可能取下面某两个值之间的值: $-\infty = \zeta_0 < \zeta_1 < \cdots < \zeta_k = \infty$ $(k \geqslant 2)$. 作为Z的实际记录, 以下面方式定义响应量Y:

$$\zeta_{j-1} < Z \leqslant \zeta_j \quad \Leftrightarrow \quad Y = j.$$

这样的Y代表了记录下来的一类, 说明不可精确观测的变量Z取值于$(\zeta_{j-1},\ \zeta_j]$. 在给定自变量$\boldsymbol{x}$时, 记$\gamma_j = P(Y \leqslant j|\boldsymbol{x})$. 一般形式的logistic比例优势模型为

$$\begin{aligned}\operatorname{logit}(\gamma_j) &= \ln\left(\frac{\gamma_j}{1-\gamma_j}\right) = \ln\left(\frac{P(Y\leqslant j|\boldsymbol{x})}{1-P(Y\leqslant j|\boldsymbol{x})}\right) = \ln\left(\frac{P(Y\leqslant j|\boldsymbol{x})}{P(Y>j|\boldsymbol{x})}\right)\\ &= \zeta_j - \boldsymbol{\beta}^\top\boldsymbol{x}.\end{aligned}$$

公式左边是logit函数, 也就是小于或等于第j类的优势的对数. 因为对于不同的j, 左边优势的对数表达式只有一个常数的差别, 因此优势本身是成比例的.[2]

[1] John Fox (2003). Effect Displays in R for Generalised Linear Models. *Journal of Statistical Software*, 8(15), 1-27. URL http://www.jstatsoft.org/v08/i15/.

[2] 如果某事件发生的概率为p, 则$p/(1-p)$称为优势(odds); 优势的对数则是p的logit函数$\ln[p/(1-p)]$. 在很多场合, 人们喜欢用优势而不是概率. 比如, 某人说其猜中某件事的优势是3:5, 实际上是说猜中的概率为3/8, 因为3/5 = (3/8)/(5/8). 优势之比称为优比(odds ratio).

对于例6.5的因变量poverty, 三个水平Too Little, About Right, Too Much分别相应于区间$(-\infty,\zeta_1]$, $(\zeta_1,\zeta_2]$, (ζ_2,∞).

还有更一般的情况. 考虑线性模型$Z=\boldsymbol{\beta}^\top\boldsymbol{x}+\epsilon$. 这里假定$\epsilon$的累积分布函数为$F(\cdot)$. 那么, $P(Z\leqslant z)=F(z-\boldsymbol{\beta}^\top\boldsymbol{x})$.

$$\gamma_j=P(Y\leqslant j)=P(Z\leqslant\zeta_j)=F(\zeta_j-\boldsymbol{\beta}^\top\boldsymbol{x}) \quad \text{或者} \quad F^{-1}(\gamma_j)=\zeta_j-\boldsymbol{\beta}^\top\boldsymbol{x}.$$

当这里的函数$F(z)=\mathrm{e}^z/(1+\mathrm{e}^z)$时, 相应的模型就是这里的logistic比例优势模型, 那时ϵ有logistic密度. 如果F是正态分布累积分布函数, 则有累积probit模型. 如果ϵ有极值分布(即$\exp(\epsilon)$有指数或Weibull分布), 则相应于成组的比例危险模型或互补的log-log模型.

首先输入例6.5数据, 并把变量poverty的水平按照次序排序(这一步很重要, 否则会得到错误的输出), 这样做的代码如下:

```
w=read.csv("WVS.csv")
levels(w[,1])#检查发现其水平次序是按照字母顺序排列的
#下面按升序排列水平:
w[,1]=ordered(w[,1],c("Too Little","About Right","Too Much"))
```

如果上面代码`levels(w[,1])`所显示的水平已经按升序排列了, 后面接着的命令可以简化成`w[,1]=ordered(w[,1])`.

用logistic比例优势模型对例6.5数据进行拟合, 使用的是程序包`MASS`中的`polr()`函数[1], 下面是拟合数据的代码:

```
library(MASS);library(car)
a0 <- polr(poverty ~ ., w)
Anova(a0)
```

得到下面输出:

```
Analysis of Deviance Table (Type II tests)

Response: poverty
         LR Chisq Df Pr(>Chisq)
religion    5.434  1   0.019753 *
degree      4.518  1   0.033542 *
country   250.881  3  < 2.2e-16 ***
age        51.120  1   8.69e-13 ***
gender     11.096  1   0.000865 ***
```

[1] 该函数的名字为proportional odds logistic regression的缩写.

看来各个变量都比较显著. 如果做Pearson χ^2检验, 变量country是最显著的. 对于Pearson χ^2检验, 可以用下面代码对因变量和各个定性自变量做检验:

```
for(i in c(2:4,6))print(chisq.test(w[,1],w[,i])$p.value)
```

得到poverty和各个变量的Pearson χ^2检验的p值:

```
     religion        degree       country        gender
 9.593472e-01  1.221051e-02 7.611715e-126  2.077244e-03
```

可以看出, poverty只和religion没有太大关系, 与其他变量都有不同程度的相关. 但是要注意, Pearson χ^2检验并没有考虑poverty各个水平的顺序.

上面的模型没有考虑交互作用, 我们通过下面代码来考察各种两两交互作用:

```
a1=polr(poverty~
    (religion+degree+country+age+gender)^2,w)
Anova(a1)
```

得到不同显著性的交互作用结果(这里不展示). 根据这个结果, 我们选择拟合下面模型:

```
a2=polr(poverty~religion*(degree+country)+
    country*age+gender,w)
Anova(a2)
```

得到

```
Analysis of Deviance Table (Type II tests)

Response: poverty
                LR Chisq Df Pr(>Chisq)
religion           4.530  1  0.0333142 *
degree             4.232  1  0.0396800 *
country          254.216  3  < 2.2e-16 ***
age               52.485  1  4.336e-13 ***
gender            10.814  1  0.0010074 **
religion:degree    7.241  1  0.0071259 **
religion:country  18.886  3  0.0002886 ***
country:age       16.891  3  0.0007441 ***
```

这个模型中的交互作用还算显著.

上面的拟合没有考虑唯一的定量变量age的高阶影响, 下面把关于age的多项式样条的B样条基矩阵也放入自变量, 这里使用了程序包splines[1]的bs()函数, 一般的具有交互作用的拟合代码如下:

```
library(splines)
a3=polr(poverty~(religion+degree+country+bs(age,4)+gender)^2,w)
Anova(a3)
```

从得到的结果(这里没有显示)中发现, gender和任何其他变量的交互作用都不显著. 因此我们拟合下面模型:

```
a4=polr(poverty~(religion+degree+country+bs(age,4))^2+gender,w)
Anova(a4)
```

得到

```
Analysis of Deviance Table (Type II tests)

Response: poverty
                    LR Chisq Df Pr(>Chisq)
religion               4.189  1  0.0406781 *
degree                 4.552  1  0.0328868 *
country              249.846  3  < 2.2e-16 ***
bs(age, 4)            64.109  4  3.964e-13 ***
gender                11.218  1  0.0008103 ***
religion:degree        4.072  1  0.0436018 *
religion:country      17.405  3  0.0005832 ***
religion:bs(age, 4)   10.056  4  0.0394905 *
degree:country        10.072  3  0.0179614 *
degree:bs(age, 4)     10.103  4  0.0387358 *
country:bs(age, 4)    41.343 12  4.294e-05 ***
```

这个模型的各个变量看上去都算是显著的.

为了得到各种估计的参数, 可以使用代码summary(a4), 输出为:

```
Re-fitting to get Hessian

Call:
polr(formula=poverty~(religion+degree+country+bs(age,
```

[1]R Core Team (2014). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. URL http://www.R-project.org/.

```
    4))^2 + gender, data = w)

Coefficients:
                                  Value Std. Error t value
religionyes                     -0.43811     0.3470 -1.2627
degreeyes                       -0.27955     0.3965 -0.7051
countryNorway                    0.86547     0.3920  2.2079
countrySweden                    1.17969     0.6162  1.9144
countryUSA                       0.15750     0.3997  0.3941
bs(age, 4)1                      0.96799     0.6098  1.5875
bs(age, 4)2                      0.47544     0.6273  0.7579
bs(age, 4)3                      0.80909     0.9587  0.8440
bs(age, 4)4                      2.10000     1.0447  2.0102
gendermale                       0.17907     0.0535  3.3473
religionyes:degreeyes            0.40449     0.2013  2.0094
religionyes:countryNorway       -0.31957     0.2213 -1.4442
religionyes:countrySweden       -1.03572     0.5107 -2.0279
religionyes:countryUSA           0.45474     0.1782  2.5521
religionyes:bs(age, 4)1          0.06682     0.6114  0.1093
religionyes:bs(age, 4)2          1.20017     0.6399  1.8755
religionyes:bs(age, 4)3          0.57554     0.9580  0.6008
religionyes:bs(age, 4)4         -0.39871     1.0776 -0.3700
degreeyes:countryNorway         -0.05401     0.2142 -0.2522
degreeyes:countrySweden          0.49464     0.2265  2.1841
degreeyes:countryUSA             0.25318     0.2128  1.1898
degreeyes:bs(age, 4)1            0.18454     0.6063  0.3044
degreeyes:bs(age, 4)2           -1.41669     0.5882 -2.4084
degreeyes:bs(age, 4)3            2.19897     0.9988  2.2017
degreeyes:bs(age, 4)4           -3.84495     1.3306 -2.8897
countryNorway:bs(age, 4)1       -0.48173     0.6316 -0.7628
countrySweden:bs(age, 4)1       -0.50033     0.6868 -0.7285
countryUSA:bs(age, 4)1           0.71088     0.6491  1.0951
countryNorway:bs(age, 4)2       -1.26592     0.6834 -1.8523
countrySweden:bs(age, 4)2       -1.67004     0.7900 -2.1141
countryUSA:bs(age, 4)2          -1.64182     0.6002 -2.7354
countryNorway:bs(age, 4)3       -1.77748     1.1218 -1.5845
```

```
countrySweden:bs(age, 4)3 -0.99290     1.3162 -0.7544
countryUSA:bs(age, 4)3     1.65883     0.8674  1.9124
countryNorway:bs(age, 4)4  0.70399     1.6493  0.4268
countrySweden:bs(age, 4)4  0.29334     2.3227  0.1263
countryUSA:bs(age, 4)4    -0.79930     0.8515 -0.9386

Intercepts:
                        Value   Std. Error t value
Too Little|About Right  0.8239  0.3401     2.4225
About Right|Too Much    2.6571  0.3419     7.7721

Residual Deviance: 10290.80
AIC: 10368.80
```

和以前一样, 没有出现的水平或截距、与水平交互作用的斜率, 一定是参照效应, 而且, 根据软件默认为0, 比如, **religionno**为0, **degreeno**为0, **countryAustralia**为0, **genderfemale**为0, 等等.

同样的模型可以用不同的程序包拟合, 下面代码用程序包**VGAM**[1]的函数**vglm()**[2]:

```
a41=vglm(poverty~(religion+degree+country+bs(age,4))^2+gender,
    family=cumulative(link="logit",parallel=TRUE),data=w)
summary(a41)
```

得到

```
Call:
vglm(formula=poverty~(religion + degree + country + bs(age,
    4))^2+gender,family=cumulative(link="logit",parallel=TRUE),
    data = w)

Pearson residuals:
                  Min      1Q Median     3Q   Max
logit(P[Y<=1]) -2.096 -1.0639 0.5438 0.8996 2.581
logit(P[Y<=2]) -4.658  0.1949 0.2538 0.5572 1.233
```

[1]Thomas W. Yee and C. J. Wild (1996). Vector Generalized Additive Models. *Journal of Royal Statistical Society*, Series B, 58(3), 481-493.

[2]该函数名称是vector generalized linear models的缩写.

```
Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept):1                    0.82380    0.34427   2.393 0.016718
(Intercept):2                    2.65704    0.34603   7.679 1.61e-14
religionyes                      0.43828    0.35050   1.250 0.211140
degreeyes                        0.27935    0.40754   0.685 0.493057
countryNorway                   -0.86570    0.40630  -2.131 0.033114
countrySweden                   -1.17949    0.62036  -1.901 0.057263
countryUSA                      -0.15739    0.38725  -0.406 0.684420
bs(age, 4)1                     -0.96828    0.61351  -1.578 0.114507
bs(age, 4)2                     -0.47391    0.62128  -0.763 0.445589
bs(age, 4)3                     -0.81001    0.94544  -0.857 0.391581
bs(age, 4)4                     -2.09990    0.99266  -2.115 0.034393
gendermale                      -0.17908    0.05335  -3.356 0.000789
religionyes:degreeyes           -0.40447    0.20043  -2.018 0.043588
religionyes:countryNorway        0.31968    0.23157   1.381 0.167432
religionyes:countrySweden        1.03574    0.50602   2.047 0.040673
religionyes:countryUSA          -0.45468    0.17358  -2.619 0.008810
religionyes:bs(age, 4)1         -0.06648    0.61370  -0.108 0.913736
religionyes:bs(age, 4)2         -1.20228    0.62967  -1.909 0.056214
religionyes:bs(age, 4)3         -0.57387    0.93562  -0.613 0.539645
religionyes:bs(age, 4)4          0.39735    1.01462   0.392 0.695335
degreeyes:countryNorway          0.05413    0.21855   0.248 0.804377
degreeyes:countrySweden         -0.49459    0.23047  -2.146 0.031870
degreeyes:countryUSA            -0.25307    0.21147  -1.197 0.231431
degreeyes:bs(age, 4)1           -0.18375    0.62145  -0.296 0.767480
degreeyes:bs(age, 4)2            1.41461    0.59107   2.393 0.016696
degreeyes:bs(age, 4)3           -2.19571    1.00423  -2.186 0.028783
degreeyes:bs(age, 4)4            3.84026    1.30788   2.936 0.003322
countryNorway:bs(age, 4)1        0.48127    0.64869   0.742 0.458138
countrySweden:bs(age, 4)1        0.49857    0.69832   0.714 0.475248
countryUSA:bs(age, 4)1          -0.71148    0.62694  -1.135 0.256436
countryNorway:bs(age, 4)2        1.26812    0.69992   1.812 0.070015
countrySweden:bs(age, 4)2        1.67473    0.80525   2.080 0.037548
countryUSA:bs(age, 4)2           1.64268    0.57840   2.840 0.004511
countryNorway:bs(age, 4)3        1.77392    1.14633   1.547 0.121750
```

```
countrySweden:bs(age, 4)3  0.98472    1.34719   0.731 0.464813
countryUSA:bs(age, 4)3    -1.66053    0.83467  -1.989 0.046651
countryNorway:bs(age, 4)4 -0.69724    1.67900  -0.415 0.677942
countrySweden:bs(age, 4)4 -0.27721    2.40550  -0.115 0.908256
countryUSA:bs(age, 4)4     0.80150    0.81636   0.982 0.326203
---
Number of linear predictors:  2

Names of linear predictors: logit(P[Y<=1]), logit(P[Y<=2])

Dispersion Parameter for cumulative family:   1

Residual deviance: 10290.8 on 10723 degrees of freedom

Log-likelihood: -5145.4 on 10723 degrees of freedom

Number of iterations: 5
```

除了一个负号之外, 结果当然和前面使用函数`polr()`是一样的. 这里的负号显然是因为它用的模型是

$$\text{logit}(\gamma_j) = \zeta_j + \boldsymbol{\beta}^\top \boldsymbol{x}$$

而不是我们前面的$\text{logit}(\gamma_j) = \zeta_j - \boldsymbol{\beta}^\top \boldsymbol{x}$.

第七章　纵向数据(多水平模型、面板数据)

纵向数据与横截面数据不同, 每个对象有不止一个观测值, 这些不同的观测是在不同时间记录的, 各个对象的观测次数不一定一样, 观测间隔不一定一样, 观测次数也不一定很多, 因此, 这与经典时间序列模型所要求的大量相等时间间隔的观测大不相同.

这种数据既不同于横截面数据, 也不能用时间序列的方法来处理. 这就产生了分析这一类数据的一套方法. 在这类数据中, 一些是描述随时间而变化的度量, 因此也叫纵向数据(longitudinal data). 由于数据背景不同, 着手的出发点不同, 处理这类数据的方法有很多不同的名称. 这些名称包括随机效应模型(random-effects models)、方差分量模型(variance component models)、多层模型或多水平模型(multilevel models)、两步模型(two-stage models)、随机效应混合模型(random-effects mixed models)、混合模型(mixed models)、经验贝叶斯模型(empirical Bayes models)、随机回归模型(random regression models)、分层线性模型(hierarchical linear models). 此外, 计量经济学中的面板数据(panel data)是纵向数据的特例. 这些模型的一个基本特点就是把不同个体的差异作为随机效应加入回归模型之中以反映个体对其重复测量的影响. 这些随机效应描述了每个对象的度量随时间的变化, 并解释了纵向数据的相关结构. 由于不要求每个对象都有相同的观测数目, 观测的时间间隔也不那么严格, 因此这类模型有一定的优势.

下面例7.1的数据就是一个典型的纵向数据.

例 7.1　Riesby数据(riesby.csv, riesby4.csv)　这两个数据[1]来源于Riesby et al. (1977), 获取该数据的目的是研究丙咪嗪(imipramine, IMI)和脱甲丙咪嗪(desipramine, DMI)血浆水平之间的关系以及66个患抑郁症的住院病人的反应. 丙咪嗪是抗抑郁的基本药物, 它可以生物转化成活性代谢物脱甲丙咪嗪. 抑郁症被划分为两种主要类型, 一种是非内生的(nonendogenous)或反映型忧郁, 与某些悲剧式生活经验有关; 另一种是内生的(endogenous), 与各种外部事件无关. 药物一般对后者更有效. 在本例中有29个患者被划分为非内生的, 而余下的37个患者被划分为内生的. 该项研究的设计为: 在服用一周的安慰剂之后, 患者连续4周每天摄入225mg的丙咪嗪. 在该项研究中, 在服用安慰剂的一周的头尾以及4个用药的周末各测量一次Hamilton抑郁得分(Hamilton depression (HD) rating scale). 在每个用药的周末都测量血浆的IMI及代谢DMI 的水平, 此外, 还记录患者

[1]这两个数据可分别从下面的网址下载: http://tigger.uic.edu/~hedeker/RIESBY.DAT.txt, http://tigger.uic.edu/~hedeker/RIESBYT4.DAT.txt.

的性别和年龄及其内生或非内生的诊断结果. 虽然患者总数为66, 但在研究过程中人数有波动, 在第0周(开始用安慰剂)有61个患者, 第1周(结束安慰剂周)有63人, 第2周(结束第1周用药)有65人, 第3周(结束第2周用药)有65人, 第4周(结束第3周用药)有63人, 第5周(结束第4周用药)有58人. 所有人中, 只有46人完成了整个过程. 如果按照重复测量的多元方差分析, 则必须舍弃将近1/3的数据, 然而, 我们将利用所有66个对象. 数据riesby4.csv的变量情况见表7.1. riesby.csv数据在时间上多一些, 变量仅有id, hamd(即hamdelt),week, endog有意义, 因此不另外介绍. 在这两个数据中都有变量intcpt, 这个变量全取1, 看来是为了产生一个截距项而加, 在用R软件时是不必要的.

表 7.1　例7.1 riesby4.csv数据变量情况

变量名	描述	性质	变量名	描述	性质
id	患者代码	分类	sex	性别(0/1)	分类
hamdelt	HD得分	数量	endog	内生	分类
intcpt	附加的1	不用	lnimi	IMI水平	数量
week	周数(0-3)	分类	lndmi	DMI 水平	数量

在这个数据中, 每个患者都重复观测若干次, 因此各个观测不是独立的, 至少每一个患者本身的各个HD度量是相关的. 他们的HD度量对时间的散点图展示在图7.1中(根据riesby.csv数据). 显然, 这些患者的观测次数并不多, 而且各个患者的观测次数也不尽相同. 绘制该图的代码为:

```
w=read.csv("riesby.csv",na.strings="-9")#数据缺失值用-9代表
interaction.plot(w$week,w$id,w$hamd,legend=F,type="o",pch=1,
    xlab="week",ylab="Hamilton depression rating scale")
```

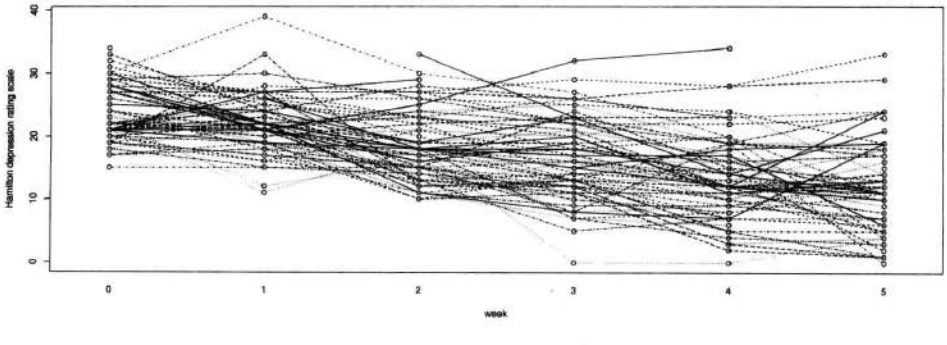


图 7.1　例7.1的66个患者的HD度量对时间的散点图

后面将要对这个数据做各种分析.

7.1 线性随机效应混合模型

图7.1表明, 例7.1riesby.csv数据的每个患者的HD度量(变量名hamd, 用y表示)随时间(变量名week, 用x表示)变化的曲线近似于直线, 但斜率和截距均有不同, 因此不能简单用诸如

$$y_{ij} = \pi_{0i} + \pi_{1i}x_{ij} + \epsilon_{ij}, \quad \epsilon_{ij} \sim N(0, \sigma_\epsilon^2).$$

(这里的下标ij对应于第i个对象的第j次观测)那样处理横截面数据的简单回归来拟合数据, 但是, 这里的系数即截距π_{0i}和斜率π_{1i}, 很可能受某种随机因素影响, 于是可以尝试建立第二层模型:

$$\begin{aligned}\pi_{0i} &= \beta_0 + \zeta_{0i},\\ \pi_{1i} &= \beta_1 + \zeta_{1i},\end{aligned}$$

式中, β_0, β_1是固定的数, 而假定

$$\boldsymbol{\zeta}_i \equiv \begin{pmatrix}\zeta_{0i}\\ \zeta_{1i}\end{pmatrix} \sim N(\mathbf{0}, \boldsymbol{\Sigma}) \equiv N\left(\mathbf{0}, \begin{pmatrix}\sigma_0^2 & \sigma_{01}\\ \sigma_{10} & \sigma_1^2\end{pmatrix}\right),$$

同时假定所有的$\boldsymbol{\zeta}_i$与所有的ϵ_{ij}独立. 把这第二层模型的π_{0i}, π_{1i}代入前面的(第一层)模型, 有

$$\begin{aligned}y_{ij} &= (\beta_0 + \zeta_{0i}) + (\beta_1 + \zeta_{1i})x_{ij} + \epsilon_{ij}\\ &= (\beta_0 + \beta_1 x_{ij}) + (\zeta_{0i} + \zeta_{1i}x_{ij} + \epsilon_{ij}),\end{aligned}$$

式中, 第一个括号内为该模型的非随机部分, 第二个括号内为其随机部分. 这里的正态性假定有很强的主观性, 也不一定合理, 这个假定是为使用最大似然法来拟合而做的. 由正态性假定, 因变量就是正态的, 因而可以根据上面的模型很容易地写出因变量的似然函数, 似然函数中的$\beta_0, \beta_1, \sigma_\epsilon, \sigma_0, \sigma_1, \sigma_{01}(=\sigma_{10})$(下面介绍的R程序包计算的是相关系数$\gamma_{01} = Cor(\zeta_{0i}, \zeta_{1i})$的估计, 而不是协方差$\sigma_{01}$的估计) 等6个参数需要通过最大似然法(maximum likelihood, ML)或约束的最大似然法(restricted maximum likelihood, REML) 来估计.

下面用这个模型来拟合例7.1的riesby.csv数据. 这里用程序包`lme4`[1]中的函数`lmer()`, 代码(包括读入数据)为:

```
library(lme4)
w=read.csv("riesby.csv",na.strings="-9")#数据缺失值用-9代表
(b22=lmer(hamd~week+(week|id),w))
```

[1] Douglas Bates, Martin Maechler and Ben Bolker (2011). lme4: Linear mixed-effects models using S4 classes. R package version 0.999375-42. http://CRAN.R-project.org/package=lme4.

注意这里代码的形式与公式中的很相似, 由于模型的非随机部分及随机部分都有时间week, 自变量的非随机部分用week表示, 而随机部分用(week|id)表示, 以指明个体id的作用. 函数lmer()默认把缺失值删除. 输出结果显示出那6个待估计参数的估计值(注意: 给出的是ζ_{0i}及ζ_{1i}的相关系数的估计$\hat{\gamma}_{01}$, 而不是它们协方差的估计):

$$\hat{\sigma}_0 = 3.598, \hat{\sigma}_1 = 1.458,\ \hat{\gamma}_{01} = -0.28,$$
$$\hat{\sigma}_\epsilon = 3.495,\ \hat{\beta}_0 = 23.577,\ \hat{\beta}_1 = -2.377.$$

这些输出结果摘自下面的计算机打印结果(读者很快就会熟练解读这类输出):

```
Linear mixed model fit by REML ['lmerMod']
Formula: hamd ~ week + (week | id)
   Data: w
REML criterion at convergence: 2219.918
Random effects:
 Groups   Name        Std.Dev. Corr
 id       (Intercept) 3.598
          week        1.458    -0.28
 Residual             3.495
Number of obs: 375, groups:  id, 66
Fixed Effects:
(Intercept)         week
     23.577       -2.377
```

下面我们再拟合有更多变量的riesby4.csv数据. 在拟合之前, 先看看变量sex对HD得分是否有影响, 分别对全体患者、非内生患者、内生患者点出三个盒形图(图7.2的上、中、下三图), 从盒形图看似乎有些关系. 该图是用下面代码作出的:

```
u=read.csv("riesby4.csv",na.strings="-9")
par(mfrow=c(3,1))
boxplot(u[,2]~u[,5],horizontal =T, col="grey")
title("All patients")
boxplot(u[u[,6]==0,2]~u[u[,6]==0,5],horizontal=T,col="grey")
title("endog=0")
boxplot(u[u[,6]==1,2]~u[u[,6]==1,5],horizontal=T,col="grey")
title("endog=1")
```

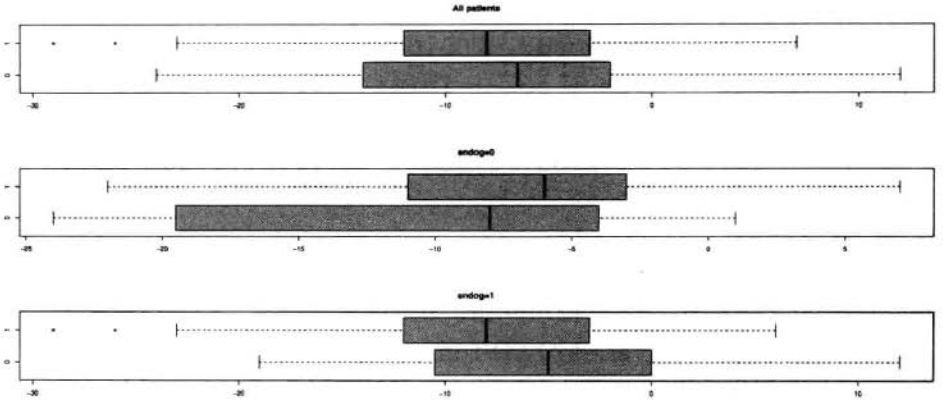


图 7.2 例7.1数据的66个患者的HD度量对性别的盒形图

用y_{ij}代表因变量hamdelt, $x_{1ij}, x_{2ij}, x_{3ij}, x_{4ij}, x_{5ij}$分别代表自变量week, sex, lnimi, lndmi, endog, 这里下标ij对应于个体i的第j个观测. 我们考虑下面的随机效应混合模型:

$$y_{ij} = (\beta_0 + \sum_{k=1}^{5} \beta_k x_{kij}) + (\zeta_{0i} + \zeta_{1i} x_{1ij} + \zeta_{2i} x_{2ij} + \epsilon_{ij}),$$

这里假定

$$\boldsymbol{\zeta}_i = \begin{pmatrix} \zeta_{0i} \\ \zeta_{1i} \\ \zeta_{2i} \end{pmatrix} \sim N(\mathbf{0}, \boldsymbol{\Sigma}) \equiv N\left(\mathbf{0}, \begin{pmatrix} \sigma_0^2 & \sigma_{01} & \sigma_{02} \\ \sigma_{10} & \sigma_1^2 & \sigma_{12} \\ \sigma_{20} & \sigma_{21} & \sigma_2^2 \end{pmatrix}\right), \ \epsilon_{ij} \sim N(0, \sigma_\epsilon^2),$$

而且假定所有的$\boldsymbol{\zeta}_i$与所有的ϵ_{ij}独立. 由于`lme4`程序包对这个模型的计算不稳定, 我们采用程序包`nmle`[1]中的`lme()`函数拟合例7.1的riesby4.csv 数据:

```
library(nlme)
u=read.csv("riesby4.csv",na.strings="-9")
u=na.omit(u)#lme()不自动删除缺失值
(b=lme(hamdelt~week+sex+lnimi+lndmi+endog,
    random=~week+sex|id,u))
```

结果输出为:

```
Linear mixed-effects model fit by REML
```

[1]Pinheiro J, Bates D, DebRoy S, Sarkar D and R Core Team (2015). _nlme: Linear and Nonlinear Mixed Effects Models_. R package version 3.1-120, <URL: http://CRAN.R-project.org/package=nlme>.

```
  Data: u
  Log-restricted-likelihood: -705.0036
  Fixed: hamdelt ~ week + sex + lnimi + lndmi + endog
(Intercept)    week      sex    lnimi    lndmi  endog
  0.8105    -1.9773 -1.2188  0.6697  -1.7111 0.7737

Random effects:
 Formula: ~week + sex | id
 Structure: General positive-definite,
     Log-Cholesky parametrization
            StdDev   Corr
(Intercept) 5.136642 (Intr) week
week        1.738191  0.680
sex         5.423979 -0.605 -0.644
Residual    3.073970

Number of Observations: 238
Number of Groups: 66
```

根据输出, 可得各个参数的估计. 由于模型稍微多用了些变量, 这个拟合riesby4.csv数据的输出比前面拟合riesby.csv数据的输出多一些内容, 但形式类似, 相信读者很容易理解其意义.

现在可以引进线性随机效应混合模型的一般形式

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta}_i + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \quad i = 1, 2, ..., N,$$

式中, $\boldsymbol{y}_i$为$n_i \times r$维的; $\boldsymbol{X}_i$为$n_i \times p$维的; $\boldsymbol{\beta}_i$为$p \times r$维的; $\boldsymbol{Z}_i$为$n_i \times q$维的; $\boldsymbol{b}_i$为$q \times r$维的, $\boldsymbol{\epsilon}_i$为$n_i \times r$维的.

$$\begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_N \end{pmatrix} \sim N(\boldsymbol{0}, \boldsymbol{\Psi});$$

对所有的i, $\boldsymbol{\epsilon}_i \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$, 且独立于$\boldsymbol{b}_i$. 通常$\boldsymbol{X}_i$及$\boldsymbol{Z}_i$的第一列为常数, $\boldsymbol{Z}_i$包含的是$\boldsymbol{X}_i$的子集, 要估计的是$\boldsymbol{\beta}_i, \boldsymbol{\Sigma}, \boldsymbol{\Psi}$. 公式中的$\boldsymbol{X}_i\boldsymbol{\beta}_i$为固定效应部分,

而$\boldsymbol{Z}_i\boldsymbol{b}_i$为随机效应部分. 如果把数据写成下面形式

$$\boldsymbol{y}=\begin{pmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\\ \vdots\\ \boldsymbol{y}_N\end{pmatrix};\boldsymbol{X}=\begin{pmatrix}\boldsymbol{X}_1 & 0 & \cdots & 0\\ 0 & \boldsymbol{X}_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \boldsymbol{X}_N\end{pmatrix};\boldsymbol{Z}=\begin{pmatrix}\boldsymbol{Z}_1 & 0 & \cdots & 0\\ 0 & \boldsymbol{Z}_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \boldsymbol{Z}_N\end{pmatrix}$$

和

$$\boldsymbol{\beta}=\begin{pmatrix}\boldsymbol{\beta}_1 & 0 & \cdots & 0\\ 0 & \boldsymbol{\beta}_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \boldsymbol{\beta}_N\end{pmatrix};\boldsymbol{b}=\begin{pmatrix}\boldsymbol{b}_1 & 0 & \cdots & 0\\ 0 & \boldsymbol{b}_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \boldsymbol{b}_N\end{pmatrix};\boldsymbol{\epsilon}=\begin{pmatrix}\boldsymbol{\epsilon}_1\\ \boldsymbol{\epsilon}_2\\ \vdots\\ \boldsymbol{\epsilon}_N\end{pmatrix}.$$

模型$\boldsymbol{y}_i=\boldsymbol{X}_i\boldsymbol{\beta}_i+\boldsymbol{Z}_i\boldsymbol{b}_i+\boldsymbol{\epsilon}_i$ $(i=1,2,...,N)$也可以写成

$$\boldsymbol{y}=\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{Z}\boldsymbol{b}+\boldsymbol{\epsilon}.$$

式中, $\boldsymbol{y}_i$为$n_i\times r$维的, 在单因变量时, $r=1$. 对于每个$i\in\{1,2,...,N\}$, 有n_i个观测值.

说明: 这里的正态性假定很强, 它使得人们可以使用最大似然法来估计参数. 虽然有的方法不强调正态性而使用最小二乘法(7.5节), 但基本上和最大似然法等价, 因为在正态条件下, 最小二乘法等价于最大似然法. 由于真实数据的正态性及模型表示的准确性很难验证, 因此, 对于一个实际数据, 一般很难确定一个最优的模型及方法.

7.2 广义线性随机效应混合模型

前面介绍的是用与线性模型结合的随机效应混合模型来拟合纵向数据. 对于因变量为二分变量、计数变量等形式的纵向数据则可能需要用下面介绍的广义线性随机效应混合模型. 其原理与前面介绍的广义线性模型类似, 仅仅增加了随机部分而已. 本节仅介绍广义logistic模型, 相信读者能举一反三, 很快掌握其他广义模型. 先来看一个例子.

例 7.2 精神分裂症数据(schizx1.csv) 这个数据[1]由Hedeker and Gibbons (1994)研究过, 是美国国家精神健康研究所(National Institute of Mental Health, NIMH)所做的关于精神分裂症的处理及其严重程度改变的合作研究. 其中原始的因变量为imps79(imps为住院病人多维精神得分(inpatient multidimensional psychiatric scale)的英文缩写). 该得分的意义为: 1=正常无病(normal, not at all ill); 2=精神病边缘(borderline mentally ill); 3=轻度

[1]该数据可从下面网址下载: http://tigger.uic.edu/~hedeker/SCHIZX1.DAT.txt.

患病(mildly ill); 4=中等程度患病(moderately ill); 5=显著患病(markedly ill); 6=严重患病(severely ill); 7=最严重症状(among the most extremely ill). 该因变量在数据中有几种形式: 连续型的(imps79)、顺序的(imps79o)、两分的(imps79b). 其他都是自变量, 包括id, int(只有一个值1), tx, week, sweek(week的平方根), txswk(tx与sweek的乘积). 每个患者(id)有7个观测值.

下面就以例7.2数据为例来介绍广义线性随机效应混合模型中的一个特例. 我们用广义logistic回归模型来拟合数据. 选择imps79b(两个水平: 0和1代表轻重程度) 作为因变量, 而作为自变量的tx有两个水平(0=用安慰剂,1=用药), 自变量week或sweek为连续变量. 其他变量这里不用. 令x_1代表tx, x_2代表sweek, y代表imps79b. 首先拟合没有随机效应的logistic回归以作比较, 模型为:

$$\ln \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2.$$

计算通过下面的代码实现(包括读入数据):

```
w=read.csv("schizx1.csv",na.strings="-9")#数据中"-9"代表缺失值
a=glm(imps79b~tx*sweek,data=w,family=binomial)
summary(a)
```

估计的系数为$\hat{\beta}_0 = 3.7027$, $\hat{\beta}_1 = -0.4054$, $\hat{\beta}_2 = -1.1126$, $\hat{\beta}_{12} = -0.4180$, 而AIC=1370.

下面再考虑一个随机效应广义logistic模型(令下标i代表第i个个体的观测).

$$\ln \frac{p_i}{1-p_i} = \beta_{0i} + \beta_{1i} x_1 + \beta_{2i} x_2 + \beta_{i12} x_1 x_2,$$

这里第二层模型为:

$$\begin{aligned} \beta_{0i} &= \beta_0 + \zeta_{0i}, \\ \beta_{1i} &= \beta_1 + \zeta_{1i}, \\ \beta_{2i} &= \beta_2 + \zeta_{2i}. \end{aligned}$$

而$(\zeta_{0i}, \zeta_{1i}, \zeta_{2i})^\top$假定有联合正态分布, 而且$\gamma_{jk} = Cor(\zeta_{ji}, \zeta_{ki}) = Cor(\zeta_{ki}, \zeta_{ji})$, $\sigma_k^2 = Var(\zeta_{ki})$ $(k, j = 0, 1, 2)$. 于是上面的模型可写为:

$$\ln \frac{p_i}{1-p_i} = (\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{i12} x_1 x_2) + (\zeta_{0i} + \zeta_{1i} x_1 + \zeta_{2i} x_2),$$

固定部分和随机部分分别置于上式右边的两个括号中. 下面的代码用该模型拟合例7.2数据:

```
library(lme4)
a4=lmer(imps79b~tx+sweek+tx:sweek+(tx+sweek|id),
```

```
    family=binomial,data=w)
summary(a4)
```

输出的结果为:

```
Generalized linear mixed model fit by the Laplace approximation
Formula: imps79b ~ tx + sweek + tx:sweek + (tx + sweek | id)
   Data: w
  AIC  BIC logLik deviance
 1196 1250 -587.9     1176
Random effects:
 Groups Name        Variance Std.Dev. Corr
 id     (Intercept) 162.22   12.736
        tx          153.13   12.375    0.958
        sweek       118.65   10.893   -0.700 -0.845
Number of obs: 1603, groups: id, 437

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   14.944      3.264   4.578 4.69e-06 ***
tx            15.056      4.442   3.390   0.0007 ***
sweek         -2.613      2.278  -1.147   0.2513
tx:sweek     -12.227      2.715  -4.504 6.65e-06 ***
```

从中可以读出: AIC=1196, $\hat{\sigma}_0^2 = 162.22$, $\hat{\sigma}_1^2 = 153.13$, $\hat{\sigma}_2^2 = 118.65$, $\hat{\gamma}_{01} = 0.958$, $\hat{\gamma}_{02} = -0.700$, $\hat{\gamma}_{12} = -0.845$, $\hat{\beta}_0 = 14.944$, $\hat{\beta}_1 = 15.056$, $\hat{\beta}_2 = -2.613$, $\hat{\beta}_{12} = -12.227$. 这里的系数大小和正负号与单纯的logistic回归大不相同. 结合背景问题, 对该输出的具体解释就留给读者了.

7.3 决策树及随机效应模型

第7.1节末尾给出了线性随机效应混合模型的一般形式

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta}_i + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \quad i = 1, 2, ..., N.$$

这里, 固定效应和随机效应部分都是线性的. 但是, 如果固定效应不一定是线性的, 而又无法写出非线性关系的分析表达式, 则可以用决策树来取代固定的部分. 该方法是Sela et al. (2011)给出的. 其公式为:

$$\boldsymbol{y}_i = f(\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_p) + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\epsilon}_i, \quad i = 1, 2, ..., N,$$

式中, $\boldsymbol{y}_i$为$n_i \times r$维的; $\boldsymbol{X}_i$为$n_i \times p$维的; $\boldsymbol{Z}_i$为$n_i \times q$维的; $\boldsymbol{b}_i$为$q \times r$维的; $\boldsymbol{\epsilon}_i$为$n_i \times r$维的.

$$\begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_q \end{pmatrix} \sim N(\mathbf{0}, \boldsymbol{\Psi});$$

对所有的i, $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma})$而且独立于$\boldsymbol{b}_i$. 这里公式中的$f(\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_p)$为用决策树来解释的固定效应部分, 而$\boldsymbol{Z}_i\boldsymbol{b}_i$为随机效应部分. 这种建模使得模型的适应性及灵活性大大提高.

下面用该模型来拟合例7.1的数据并输出图7.3.

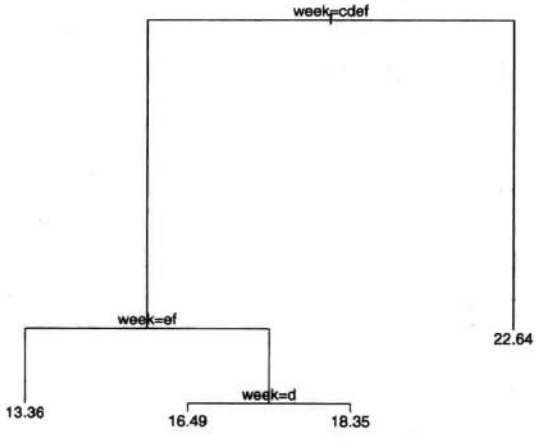


图 7.3 随机效应模型拟合中固定效应部分的决策树

所用的程序包为REEMtree.[1] 代码如下:

```
w=read.csv("riesby.csv",na.strings="-9")
w$week=factor(w$week);w=na.omit(w)
library(REEMtree)
a<-REEMtree(hamd~week, data=w, random=~week|id)
print(a);plot(a)
```

[1]Rebecca J. Sela and Jeffrey S. Simonoff (2011). REEMtree: Regression Trees with Random Effects. R package version 0.90.3.

下面是输出结果, 图7.3为固定效应部分的决策树, 图中week可能取的值(不一定都取到)a, b, c, d, e, f分别代表week的实际(哑元)值0, 1, 2, 3, 4, 5. 而在图中的节点处, 标出"week=cdef"意味着如果week取2, 3, 4, 5, 那么走向左叉, 否则到右叉, 参见下面打印输出. 下面的输出除了有决策树的细节之外, 还有估计的随机部分的有关协方差和方差. 这个程序的运行比较慢, 需要耐心等待. 对结果的解释留给读者来讨论.

```
[1]  "RE-EM Tree "
n= 375

node), split, n, deviance, yval
      * denotes terminal node

 1) root 375 5817.55100 17.83467
   2) week=2,3,4,5 251 1410.00500 15.45928
     4) week=4,5 121   94.57532 13.35514 *
     5) week=2,3 130  281.08770 17.41775
      10) week=3 65   86.56147 16.48644 *
      11) week=2 65   81.77366 18.34905 *
   3) week=0,1 124  124.48030 22.64293 *
[1] "Estimated covariance matrix of random effects:"
            (Intercept)     week1     week2     week3
(Intercept)   17.147331 -6.130499 -7.474685 -7.784836
week1         -6.130499 14.812644 10.079534  9.799587
week2         -7.474685 10.079534 23.816731 23.544710
week3         -7.784836  9.799587 23.544710 35.469570
week4         -9.551307  8.186031 24.358772 35.557939
week5        -11.962865  9.145116 23.085852 32.689039
               week4      week5
(Intercept) -9.551307 -11.962865
week1        8.186031   9.145116
week2       24.358772  23.085852
week3       35.557939  32.689039
week4       45.522887  39.768561
week5       39.768561  61.669898
[1] "Estimated variance of errors: 3.90687414871061"
[1] "Log likelihood:  -1097.46252573416"
```

7.4 纵向生存数据分析

例 7.3 原发性胆汁性肝硬化数据(pbc2.csv) 这个关于原发性胆汁性肝硬化(primary biliary cirrhosis, pbc)的数据[1]是最初的原发性胆汁性肝硬化(pbc)研究[2]的后续. 原先的研究可见于Fleming and Harrington(1991), 目前的数据由Murtaugh et al. (1994)研究过. 该数据有312个患者的观测, 由于每个患者不止观测一次, 该数据集有1945个观测值及20个变量. 另外, 在患者死亡之前或进行换肝手术之前的最后一次观测(行)往往比其他行有更多的缺失值. 变量情况见表7.2. 这里变量years与year都是从天数转换成年的; 用药的选择为D-penicillamine(用D-青霉胺)及placebo(用安慰剂); 变量edema(水肿)的水平的意义为No edema(无水肿), edema no diuretics(未用利尿剂水肿), edema despite diuretics(用利尿剂仍水肿). 一些缩写的原文如下: serBilir: serum bilirubin; serChol: serum cholesterol; alkaline: alkaline phosphatase; SGOT: serum glutamic-oxaloacetic transaminase. 对这个数据的研究还可参看Therneau and Grambsch (2000).

下面介绍处理这个数据的两种方法. 这两种方法的视角有很大不同, 很难比较, 希望读者能够对此进行讨论.

7.4.1 Cox 随机效应混合模型

与前面的随机效应混合模型的一般形式类似, 这里的模型为:

$$h(t) = h_0(t)\exp\{\boldsymbol{X}^\top\boldsymbol{\beta} + \boldsymbol{Z}^\top\boldsymbol{b}\},\ \boldsymbol{b} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}(\theta)),$$

式中, $h_0(t)$为不确定的基础危险函数; $\boldsymbol{X}$和$\boldsymbol{Z}$分别为固定效应及随机效应的设计矩阵; $\boldsymbol{\beta}$为固定效应的系数; $\boldsymbol{b}$为服从联合正态分布的随机效应的系数.

为了得到估计, 使用Cox部分(对数)似然函数(partial likelihood, PL)

$$\ln[PL(\boldsymbol{\beta}, \boldsymbol{b})] = \sum_{i=1}^{n}\int_0^\infty \left\{ Y_i(t)\eta_i(t) - \ln\left[\sum_j Y_j(t)\exp(\eta_j(t))\right]\right\} \mathrm{d}t,$$

式中, $\eta_i(t) = (\boldsymbol{X}_i(t))^\top\boldsymbol{\beta} + (\boldsymbol{Z}_i(t))^\top\boldsymbol{b}$, 而

$$Y_i(t) = \begin{cases} 1, & \text{如果对象}i\text{仍然在观察中;} \\ 0, & \text{如果对象}i\text{已不在观察中.} \end{cases}$$

然后, 通过对此似然函数做关于$\boldsymbol{\beta}$及$\boldsymbol{b}$的积分可得到需要的估计. 细节参阅Ripatti and Palmgren(2000) 和Therneau, Grambsch and Pankratz(2003).

[1]来源于网站http://lib.stat.cmu.edu/datasets/pbcseq, 也以pbc2的名字包含在程序包JM中.

[2]数据在http://lib.stat.cmu.edu/datasets/pbc.

表 7.2 例7.3 原发性胆汁性肝硬化数据变量情况

变量名	描述	性质
id	个人代码	分类
years	登记到换肝或死亡年数	数量
status	状态(alive, transplanted,dead)	分类
drug	用药(D-penicil,placebo)	分类
age	年龄	数量
sex	性别(male,female)	分类
year	登记日到访问年数	数量
ascites	腹水(NO,Yes)	分类
hepatomegaly	肝肿大(No,Yes)	分类
spiders	蜘蛛纹(No,Yes)	分类
edema	水肿(No edema, edema no diuretics, edema despite diuretics)	分类
serBilir	血清胆红素(mg/dl)	数量
serChol	血清胆固醇(mg/dl)	数量
albumin	白蛋白(mg/dl)	数量
alkaline	碱性磷酸酶(U/liter)	数量
SGOT	血清谷氨酸谷草转氨酶(U/liter)	数量
platelets	血小板(每升千个)	数量
prothrombin	凝血酶原(秒)	数量
histologic	疾病组织学分期(1,2,3,4期)	分类
status2	状态2(1=换肝或死亡, 0=活着)	分类

这里使用的程序包为coxme.[1] 把所有自变量都放入模型, 并把变量year作为随机效应的协变量, 代码如下:

```
data(pbc2, package="JM")#利用JM程序包的pbc2数据
library(coxme)
fit=coxme(Surv(years, status2)~serBilir+
    serChol+drug+year+sex+age+
    SGOT+platelets+ascites +hepatomegaly+spiders+edema+
    albumin+alkaline+prothrombin+factor(histologic)+
    (year|id),pbc2)
```

[1]Terry Therneau (2012). coxme: Mixed Effects Cox Models. R package version 2.2-2. http://CRAN.R-project.org/package=coxme.

```
fit
```

输出为:

```
Cox mixed-effects model fit by maximum likelihood
 Data: pbc2
 events,n=322,1113
    (832 observations deleted due to missingness)
 Iterations= 7 49
                NULL Integrated Fitted
Log-likelihood -2089      -1713  -1599

                  Chisq  df p AIC BIC
Integrated loglik   752  20 0 712 637
 Penalized loglik   980 100 0 779 401

Model: Surv(years,status2)~serBilir+serChol+drug+year+sex+
age+SGOT+platelets+ascites+hepatomegaly+spiders+edema+
albumin+alkaline+prothrombin+factor(histologic)+(year|id)
Fixed coefficients
                          coef   exp(coef) se(coef) z     p
serBilir                  1.35e-01  1.144  0.016  8.62 0.0e+00
serChol                   9.25e-04  1.001  0.000  2.99 2.8e-03
drugD-penicil             1.85e-02  1.019  0.148  0.12 9.0e-01
year                     -4.42e-01  0.643  0.047 -9.37 0.0e+00
sexfemale                -4.35e-01  0.647  0.201 -2.16 3.1e-02
age                       4.38e-02  1.045  0.008  5.36 8.2e-08
SGOT                      6.15e-04  1.001  0.001  0.70 4.8e-01
platelets                -5.87e-04  0.999  0.001 -0.76 4.5e-01
ascitesYes                5.55e-02  1.057  0.205  0.27 7.9e-01
hepatomegalyYes           2.17e-01  1.242  0.164  1.32 1.9e-01
spidersYes                2.27e-01  1.254  0.153  1.48 1.4e-01
edemaedema no diuretics   6.86e-02  1.071  0.173  0.40 6.9e-01
edemaedema despite        9.24e-01  2.520  0.246  3.76 1.7e-04
diuretics
albumin                  -8.23e-01  0.439  0.186 -4.41 1.0e-05
alkaline                  2.19e-05  1.000  0.000  0.64 5.2e-01
prothrombin               6.04e-02  1.062  0.051  1.19 2.3e-01
```

```
factor(histologic)2     -1.73e-02  0.983  0.537 -0.03 9.7e-01
factor(histologic)3      5.43e-01  1.721  0.514  1.06 2.9e-01
factor(histologic)4      1.02e+00  2.762  0.525  1.94 5.3e-02

Random effects
 Group Variable Std Dev Variance
 id    year     0.349   0.122
```

从上面的计算输出可以看出, 变量drug根本不显著. 比较显著的变量有serBilir (血清胆红素), year, age. 另外edema(水肿), albumin(白蛋白)和histologic(疾病组织学分期) 也比较显著. 但是必须注意, 后面的一些指标实际上反映了身体的实际状况, 不做统计分析也可以由这些指标看出患者的前景. 把这些指标及不显著的指标去掉, 再重新做拟合. 代码如下:

```
fit2=coxme(Surv(years, status2)~serBilir+year+sex+age+
    (year|id),data=pbc2)
fit2
```

结果如下:

```
Cox mixed-effects model fit by maximum likelihood
  Data: pbc2
  events, n = 725, 1945
  Iterations= 18 138
                    NULL Integrated    Fitted
Log-likelihood -5101.349  -4399.475 -4057.875

                    Chisq     df p     AIC     BIC
Integrated loglik 1403.75   5.00 0 1393.75 1370.82
 Penalized loglik 2086.95 175.14 0 1736.66  933.41

Model:Surv(years, status2)~serBilir+year+sex+age+(year|id)
Fixed coefficients
                 coef exp(coef)    se(coef)     z       p
serBilir   0.15588036 1.1686864 0.010595050 14.71 0.0e+00
year      -0.64735997 0.5234258 0.122937626 -5.27 1.4e-07
sexfemale -0.38200365 0.6824926 0.137833260 -2.77 5.6e-03
age        0.04159196 1.0424690 0.005526548  7.53 5.2e-14
```

```
Random effects
 Group Variable Std Dev  Variance
 id    year     1.526068 2.328883
```

从输出的结果可以看出, 剩下的这些变量都比较显著.

7.4.2 分步联合建模

本小节要介绍的方法源自R程序包JM.[1] 那里首先分别建立两个模型, 即纵向模型和生存模型, 然后联合两部分, 因此称为联合模型(JM: Joint Models for Longitudinal and Survival Data). 该程序包还包含了另一个数据pbc2.id, 它是pbc2数据的一部分, 仅仅包含每个患者的第一次度量, 因此只有312个观测值. 由于在建立生存分析数据模型时所需的诸如存活时间、状态、用药等内容对每个患者的各次测量都一样, 因此不需要纵向数据pbc2, 只需要pbc2.id就行了. 但在对纵向数据建模时就必须用pbc2数据了.

令$y_i(t)$为在时间t对患者i观测得到的纵向数值, 如果每个对象有n_i个观测, 令$y_{ij} = \{y_i(t_{ij}), j = 1, 2, ..., n_i\}$. 令$m_i(t)$为$y_i(t)$试图近似的(不可观测的)真实度量值, T_i为观测到的第i个对象的死亡(失败)时间, T_i^*为不可观测的真实死亡时间, C_i为观测到的删失时间, 那么$T_i = \min(T_i^*, C_i)$. 删失指数为$\delta_i = I(T_i^* \leqslant C_i)$. 再令$M_i(t) = \{m_i(u), 0 \leqslant u < t\}$表示$m_i$在时间$t$之前的(不可观测的)历史, $h_0(\cdot)$ 表示基础危险函数. Therneau and Grambsch (2000)的相对危险模型为:

$$h_i(t|M_i(t), \boldsymbol{w}_i) = h_0(t) \exp\{\boldsymbol{\gamma}^\top \boldsymbol{w}_i + \alpha m_i(t)\} \tag{7.1}$$

式中, $\boldsymbol{w}_i$为基本自变量, 如治疗办法指数、疾病历史、性别等, 相应的回归系数向量为$\boldsymbol{\gamma}$. α描述潜在的输出m_i对危险h_i的影响, 而Cox and Oakes (1984)的模型为:

$$h_i(t|M_i(t), \boldsymbol{w}_i) = h_0(V_i(t)) \exp\{\boldsymbol{\gamma}^\top \boldsymbol{w}_i + \alpha m_i(t)\} \tag{7.2}$$

式中,

$$V_i(t) = \int_0^t \exp\{\boldsymbol{\gamma}^\top \boldsymbol{w}_t + \alpha m_i(s)\} \mathrm{d}s.$$

这个模型假定个体的年龄服从某种加速机制$V_i(t)$, 而不仅仅是t的函数. 我们只能利用已有的知识$\{y_i(t)\}$来估计$m_i(t)$及$M_i(t)$. 为此假定正态模型

$$y_i(t) = m_i(t) + \epsilon_i(t) = (\boldsymbol{x}_i(t))^\top \boldsymbol{\beta} + (\boldsymbol{z}_i(t))^\top \boldsymbol{b}_i + \epsilon_i(t),\ \epsilon_i(t) \sim N(0, \sigma^2), \tag{7.3}$$

[1] Dimitris Rizopoulos (2010). JM: An R Package for the Joint Modelling of Longitudinal and Time-to-Event Data. *Journal of Statistical Software*, 35(9), 1-33. URL http://www.jstatsoft.org/v35/i09/.

这里假定ϵ_i独立于随机效应$\boldsymbol{b}_i$, 而$\boldsymbol{b}_i \sim N(\mathbf{0}, \boldsymbol{D})$. 无论按照哪个生存模型, 在条件独立的假定下, h_i对第i个对象的对数似然函数的贡献为:

$$\ln p(T_i, \delta_i, y_i; \theta) = \ln \int p(T_i, \delta_i | \boldsymbol{b}_i; \theta_t, \beta) \left[\prod_j p\{y_i(t_{ij}) | \boldsymbol{b}_i; \theta_y\} \right] p(\boldsymbol{b}_i; \theta_b) \mathrm{d}\boldsymbol{b}_i, \tag{7.4}$$

式中,

$$p(T_i, \delta_i | \boldsymbol{b}_i; \theta_t, \beta) = \{h_i(T_i | M_i(T_i); \theta_t, \beta)\}^{\delta_i} S_i(T_i | M_i(T_i); \theta_t, \beta)$$

其中的生存函数为:

$$S_i(t | M_i(t); \theta_t, \beta) = P(T_i^* > t | M_i(t); \theta_t, \beta) = \exp\left\{ - \int_0^t h_i(s | M_i(s); \theta_t, \beta) \mathrm{d}s \right\}$$

这里生存分析模型(7.1)或者生存分析模型(7.2)与纵向数据模型(7.3)通过$m_i(t)$(或y_i)结合起来, 形成了对数似然函数(7.4). 因此, 就可以利用最大似然法把这两个模型的参数求出来.

对于我们的数据, 首先用横截面数据拟合一遍, 看哪个变量可以作为上面生存模型的$m_i(t)$, 然后用它来建立纵向模型, 再联系两者, 以得出从横截面数据得不到的时间与生存函数之间的关系.

1. **探索性分析**

(1) 从横截面数据层面看看哪个自变量对生存影响大. 先把所有的变量都加入, 进行生存模型的拟合, 看哪个显著影响生存函数(或危险函数). 发现真正突出的是血清胆红素(serBilir)的对数. 由于变量太多, 输出庞大, 我们仅给出只有变量drug及log(serBilir)作为自变量的拟合及输出, 这里用的是Therneau and Grambsch (2000) 的相对危险模型, 使用的函数为程序包survival中的survreg(). 代码和结果为:

```
library(JM)
a=survreg(Surv(years,status2)~drug+log(serBilir),
    pbc2.id,x=TRUE);summary(a)
```

```
Call:
survreg(formula = Surv(years, status2) ~ drug + log(serBilir),
    data = pbc2.id, x = TRUE)
                Value Std. Error       z         p
(Intercept)    2.9703     0.1230  24.149 7.67e-129
drugD-penicil -0.0693     0.1220  -0.568  5.70e-01
log(serBilir) -0.7456     0.0644 -11.586  4.83e-31
```

```
Log(scale)     -0.3334      0.0717   -4.649   3.33e-06

Scale= 0.717

Weibull distribution
Loglik(model)= -446   Loglik(intercept only)= -511.8
        Chisq= 131.68 on 2 degrees of freedom, p= 0
```

可以看出血清胆红素log(serBilir)要比drug显著得多, 后者根本不显著.

(2) 从图形看用药与血清胆红素, 以及用药与生存函数是否有关系. 利用下面的代码(数据pbc2和pbc2.id已经在R中) 点出随着时间变化两用药组的血清胆红素对数含量变化图(见图7.4) 以及不同用药组的Kaplan-Meier生存函数图(见图7.5).

```
library("lattice")
xyplot(log(serBilir)~year|drug,group=id,data=pbc2,xlab="year",
    ylab="Serum bilirubin",col=1,type="l")
plot(survfit(Surv(years, status2) ~ drug, data = pbc2.id),
    conf.int=FALSE, mark.time=TRUE, col=c("black", "red"),
    lty = 1:2,ylab = "Survival", xlab = "Years")
legend("topright",c("D-penicillamine","placebo"),lty = 1:2,
    col=c("black","red"),bty="n")
```

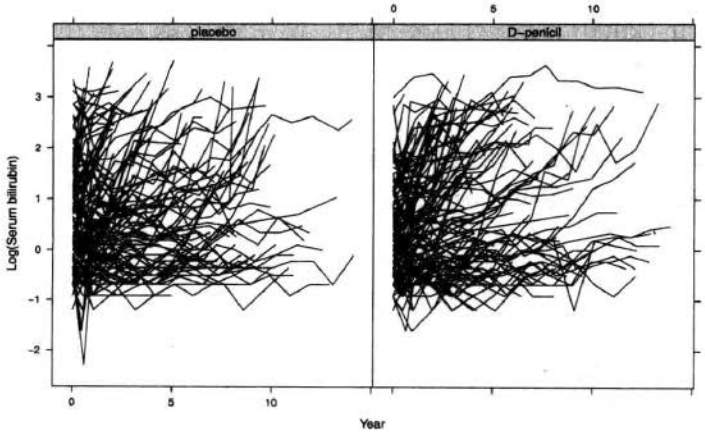


图 7.4 不同用药组的血清胆红素对数含量变化图

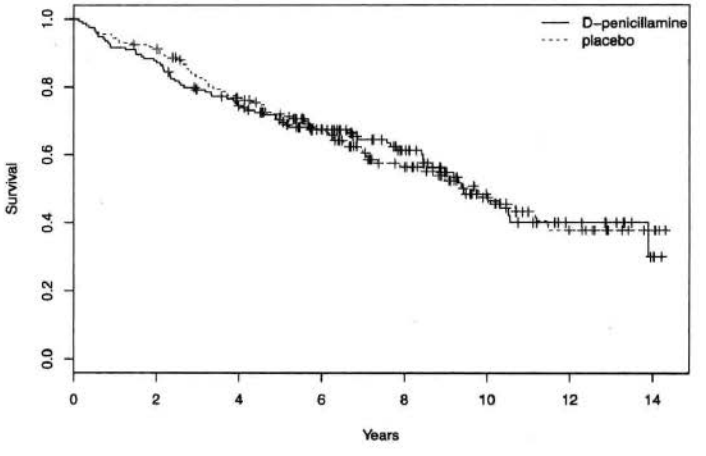


图 7.5 不同用药组的Kaplan-Meier生存函数

从图7.4看不出不同用药对血清胆红素对数含量的影响; 从图7.5也看不出用药对生存函数有什么影响.

2. **联合模型第一步: 生存函数的拟合**

作为联合模型的第一步, 用pbc2.id数据建立单独以drug(用药)为自变量的Therneau and Grambsch (2000) 相对危险模型. 代码及结果如下:

```
fitSURV=survreg(Surv(years,status2)~drug,data=pbc2.id,x=TRUE)
summary(fitSURV)
```

```
Call:
survreg(formula=Surv(years, status2)~drug,data=pbc2.id,
    x=TRUE)
                 Value Std. Error        z         p
(Intercept)    2.614872     0.1194 21.89178 3.11e-106
drugD-penicil  0.000421     0.1570  0.00268  9.98e-01
Log(scale)    -0.074076     0.0752 -0.98490  3.25e-01

Scale= 0.929

Weibull distribution
Loglik(model)= -511.8   Loglik(intercept only)= -511.8
```

```
Chisq= 0 on 1 degrees of freedom, p= 1
```

这里估计的是模型(7.1)中指数部分$[\boldsymbol{\gamma}^\top \boldsymbol{w}_i + \alpha m_i(t)]$中的$\boldsymbol{\gamma}$, 这时还没有纵向部分变量$m_i(t)$的加入(我们将选变量serBilir 的对数作为$m_i(t)$). 自变量$\boldsymbol{w}_i$取的是drug(用药). 得到的估计值为: 截距为2.614872, 不用D-penicil时对截距的贡献是0(默认), 用D-penicil时对截距的贡献为0.000421. 这里的生存函数用的是Weibull分布族.

3. 联合模型第二步: 纵向数据的拟合

从上面的结果来看, drug(用药)对生存函数没有多大影响, 而通过前面一开始以drug和血清胆红素(serBilir)的对数建立的生存模型可以看出, 血清胆红素(serBilir)含量对生存函数的影响比较显著. 因此纵向部分就以serBilir的对数作为纵向数据模型(7.3)左边的因变量. 这里模型自变量的组合, 无论是固定的还是随机的, 都经过筛选, 为节省篇幅, 这里只介绍一种选择, 即

$$\ln(\text{serBilir}) = [\mu + \text{drugD-penicil} + (\beta + \beta_1)\text{year}] + [\zeta + \zeta_1 \text{year} + \epsilon],$$

式中, 右侧两个方括号中的项分别代表固定部分和随机部分. 所要估计的包括: 截距μ, drugD-penicil(用药对截距的贡献, 不用drugD-penicil时默认为0), β(year的斜率), β_1 (drugD-penicil对year斜率的贡献, 不用drugD-penicil时, 其对year斜率的贡献默认为0), ζ与ζ_1各自的标准差及它们的相关系数, ϵ的标准差. 代码如下:

```
fitLME=lme(log(serBilir)~drug*year,random=~year|id,data=pbc2)
summary(fitLME)
```

输出结果为:

```
Linear mixed-effects model fit by REML
 Data: pbc2
       AIC      BIC    logLik
  3085.474 3130.042 -1534.737

Random effects:
 Formula: ~year | id
 Structure: General positive-definite,
    Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 0.9985870 (Intr)
year        0.1722100 0.417
```

```
Residual     0.3489329

Fixed effects: log(serBilir) ~ drug * year
                          Value  Std.Error   DF  t-value p-value
(Intercept)           0.5630313 0.08255260 1631  6.820274 0.0000
drugD-penicil        -0.1332474 0.11610813  310 -1.147614 0.2520
year                  0.1797540 0.01782101 1631 10.086631 0.0000
drugD-penicil:year   -0.0044099 0.02490439 1631 -0.177075 0.8595
 Correlation:
                     (Intr) drgD-p year
drugD-penicil        -0.711
year                  0.248 -0.176
drugD-penicil:year   -0.177  0.251 -0.716
```

结果为: $\hat{\mu} = 0.563$, 估计的drugD-penicil $= -0.133$, $\hat{\beta} = 0.180$, $\hat{\beta}_1 = -0.004$, $SD(\zeta) = 1.000$, $SD(\zeta_1) = 0.172$, $Cor(\zeta, \zeta_1) = 0.417$, $SD(\epsilon) = 0.349$. 这个结果表明, 变量year对血清胆红素的贡献极大.

4. **联合模型第三步: 联合两个模型**

下面把这两个结果即fitSURV和fitLME用函数jointModel()结合起来. 程序包JM的jointModel()函数结合了程序包nlme中的lme()函数的线性混合模型的结果及程序包survival中的coxph()或survreg()函数的结果. jointModel()中的"method = "有几种选择, 主要是危险函数的选择和近似上面的对数似然函数式(7.4)的方法的选择. 下面用的是选项method的默认值(weibull-PH-aGH), 使得在Therneau and Grambsch (2000)模型下的h_0为Weibull基本危险函数. 为了采用(拟)适应性Gauss-Hermite积分法来近似式(7.4), 使用如下代码:

```
fitJOINT <- jointModel(fitLME,fitSURV,timeVar="year")#联合模型
summary(fitJOINT)
```

得到下面的输出结果:

```
Call:
jointModel(lmeObject=fitLME2,survObject=fitSURV,timeVar="year")
Joint Model Summary:
Longitudinal Process: Linear mixed-effects model
Event Process: Weibull relative risk model
Parameterization: Time-dependent
```

```
   log.Lik       AIC       BIC
 -1918.529 3861.058 3905.974

Variance Components:
             StdDev    Corr
(Intercept)  1.0003  (Intr)
year         0.1807  0.4244
Residual     0.3471

Coefficients:
Longitudinal Process
                     Value Std.Err z-value p-value
(Intercept)         0.5594  0.0827  6.7653 <0.0001
drugD-penicil      -0.1334  0.1162 -1.1483  0.2509
year                0.1865  0.0189  9.8835 <0.0001
drugD-penicil:year -0.0038  0.0255 -0.1506  0.8803
```

这时随机部分估计的三个标准差和一个相关系数与单独的纵向模型差不多, 固定部分的估计也差不多. 可以看出, 最终对生存影响最大的变量是year而不是用药. 如果只用生存分析模型, 则无法得到这个结论, 因为单纯生存分析只能用横截面数据(此例是pbc2.id, 它的变量year只有一个值). 注意: 变量year和years是两个变量.

5. **不同模型的比较**

再用不同模型, 比如拟合纵向部分时不考虑随机斜率, 与生存模型结合:

```
fitLME0=lme(log(serBilir)~drug*year,random=~1|id,data=pbc2)
fitJOINT0=jointModel(fitLME0,fitSURV,timeVar="year")#联合
```

这里不打印最终结果, 但可以将这个模型(fitJOINT0)和前面那个模型(fitJOINT)进行比较, 代码如下:

```
anova(fitJOINT0,fitJOINT) #比较上面两个模型
```

得到下面输出结果:

```
             AIC     BIC  log.Lik    LRT df p.value
fitJOINT0 4632.89 4670.32 -2306.44
fitJOINT  3861.06 3905.97 -1918.53 775.83  2 <0.0001
```

这表明加有随机斜率的模型优于只有随机截距的模型.

7.5 计量经济学家的视角: 面板数据

在计量经济学中, 把纵向数据的一些特殊形式称为面板数据(panel data). 而且按照计量经济学的习惯观点来分类, 这些分类法和术语是一般讨论纵向数据(或多层模型, 或随机效应混合模型)的统计教科书所没有的, 因此笔者认为有必要专门介绍. 此外, 这里要使用的程序包`plm`[1]也很有特色, 和一般的关于面板数据或纵向数据的程序包不同, 它主要使用最小二乘法来拟合面板数据. 按照`plm`作者的说法, 这个程序包所能处理的模型并不是简单地被`lme4`或`nlme`所能处理的模型包含.

计量经济学用来拟合面板数据的一般模型为:

$$y_{it} = \alpha_{it} + \boldsymbol{\beta}_{it}^{\top}\boldsymbol{x}_{it} + u_{it}. \tag{7.5}$$

如果假定了参数的齐性, 即对于所有的i和t, $\alpha_{it} = \alpha$及$\boldsymbol{\beta}_{it} = \boldsymbol{\beta}$, 并且令$u_{it} = \mu_i + \epsilon_{it}$, 则有

$$y_{it} = \alpha + \boldsymbol{\beta}^{\top}\boldsymbol{x}_{it} + \mu_i + \epsilon_{it},\ i = 1, 2, ..., n,\ t = 1, 2, ..., T, \tag{7.6}$$

和前面一样, i为代表个体的下标, t为代表时间的下标. 通常假定ϵ_{it}为独立同分布的零均值、方差固定的白噪声(不一定正态), 而且独立于自变量$\boldsymbol{x}_{it}$和个体误差μ_i.

与一般的纵向数据随机效应混合模型相比, 这里的模型的确比较简单. 但正因为其简单, 所以在计量经济学文献中把该模型的各种情况讨论得非常详尽. 这里不可能讨论所有的概念和方法, 主要围绕程序包`plm`所基于的模型(7.6)的概念和术语来讨论, 当然, 这并不代表其他研究面板数据模型的文献也是如此.

如果个体误差μ_i和自变量相关, 则$\boldsymbol{\beta}$的普通最小二乘估计(OLS)是不相合的, 因此习惯上把μ_i看作n个参数来估计, 称之为固定效应(fixed effects). 如果μ_i和自变量不相关, 则称之为随机效应(random effect). 因此两项总和$u_{it} = \mu_i + \epsilon_{it}$也是随机效应, 这时通常用某些广义最小二乘估计(generalized least squares, GLS)方法来估计.

如果没有个体项μ_i, 则对$\boldsymbol{\beta}$最有效的估计为OLS, 这时模型称为汇集(pooling)模型.

如果ϵ_{it}不是白噪声, 即可能有异方差性和序列相关性, 则用更一般的GLS方法, 称为无约束的(unrestricted)或一般(general)GLS. 其特例为μ_i和自变量相关的固定效应GLS.

---

[1] Yves Croissant, Giovanni Millo (2008). Panel Data Econometrics in R: The plm Package. *Journal of Statistical Software* 27(2). URL http://www.jstatsoft.org/v27/i02/.

一种消除不随时间而变的部分的方法是在时间序列里常用的差分. 这时

$$\Delta y_{it} = \boldsymbol{\beta}^\top \Delta \boldsymbol{x}_{it} + \Delta u_{it},$$

式中, $\Delta u_{it} = \Delta(\mu_i + \epsilon_{it}) = \Delta\epsilon_{it}$. 该模型可以用OLS来估计. 这称为一阶差分(first difference, FD)估计.

此外, 还有其间(between)模型, 它基于时间或组的平均数计算.

变量系数模型(variable coefficients model)是把较强的约束$\boldsymbol{\beta}_{it} = \boldsymbol{\beta}, \forall i, t$去掉, 使得系数可以沿着一个方向变化, 比如假定$\boldsymbol{\beta}_{it} = \boldsymbol{\beta}_i$, 就是固定系数模型. 如果假定$\boldsymbol{\beta}_{it} = \boldsymbol{\beta} + \boldsymbol{\eta}_i, \forall t$, 这里$\boldsymbol{\eta}_i$为零均值的, 就是随机效应模型.

正如计量经济学中对于其他统计模型的应用(如后面要介绍的单位根检验和协整检验)一样, 这里对于确定应该使用何种模型有一系列格式化的(检验)步骤. 比如, 用函数`pooltest()`做检验以确定是否用汇集(pooling)模型, 用函数`phtest()`做Hausman类的检验以确定是用固定效应模型还是用随机效应模型等. 在实际操作中, 有的人经常以不能拒绝零假设作为"接受零假设"的理由, 这在统计中是不正确的. 不能拒绝零假设只意味着能把零假设作为一个没有被否定的假定来对待, 而绝对不是一个确定的结论.

计量经济学中的格式化步骤可能使得搞经济的人能够很方便地、不用左顾右盼地得到该领域认可的结论, 而不必涉及太多的诸如"描述经济规律是否还有其他更好的模型"之类的不易回答的问题. 这有点像沿着事先设立的路标行走. 统计专业的人可能对此种思维不那么熟悉. 实际上, 应用统计方法的许多其他领域的实际工作者都是这样做的. 这没有什么好奇怪的, 不然要学统计的干什么? 顺着路标走路不是统计学家的任务, 探索及发现可行的道路并设立路标才是统计学家的使命.

我们通过一个例子来说明如何用`plm`程序包来拟合面板数据.

例 7.4　工资数据(Wages.csv)　这是美国的595名个体从1976到1982年的数据, 共有4165个观测值, 来自收入动态(Income Dynamics, PSID)的面板研究. 来源于Baltagi(2003)和Cornwell and Rupert (1988). 在`plm`程序包中包含这个数据, 其12个变量的情况见表7.3.

对这个例子, 首先选用模型(7.6)中μ_i和自变量不相关的情况, 也就是随机效应模型, 这时, 回归估计出来的参数除了系数之外, 还有ϵ_{ij}及μ_i的方差.

先以lwage为因变量, 以married, exp及其平方, bluecol, union, sex, black, ed为自变量做回归, 由于该数据没有给出时间和个体的代码, 因此只要在函数中输入`index=595`就能自动按照顺序把每7个(7=4165/595)对象算成一个个体, 而每7个观测(按顺序)对应于从1976到1982年的观测. 相关代码如下:

```
library(plm)
```

表 7.3 例7.4 工资数据变量情况

exp	全时工作年限	整数
wks	年工作周数	整数
bluecol	是否蓝领	分类
ind	制造业工作	分类
south	在南方吗	分类
smsa	是否居住在标准都市	分类
married	结婚状况	分类
sex	性别	分类
union	工资是工会合同定的吗	分类
ed	受教育年限	整数
black	是非裔吗	分类
lwage	工资的对数	数量

```
data("Wages",package="plm")
rd=plm(lwage~married+exp+I(exp^2)+bluecol+union+sex+black+ed,
    data=Wages,model="random",index=595)
summary(rd)
```

得到下面的输出结果:

```
Oneway (individual) effect Random Effect Model
   (Swamy-Arora's transformation)

Call:
plm(formula=lwage~married+exp+I(exp^2)+bluecol+union+
    sex+black+ed,data=Wages,model="random",index=595)

Balanced Panel: n=595, T=7, N=4165

Effects:
                  var std.dev share
idiosyncratic 0.02313 0.15209  0.23
individual    0.07755 0.27847  0.77
theta:  0.7978

Residuals :
```

```
   Min. 1st Qu.  Median 3rd Qu.    Max.
-2.0600 -0.1110  0.0076  0.1190  2.0900

Coefficients :
               Estimate  Std. Error  t-value  Pr(>|t|)
(Intercept)  3.8931e+00  9.7377e-02  39.9792 < 2.2e-16 ***
marriedyes  -7.4031e-02  2.2802e-02  -3.2467 0.0011768 **
exp          8.4119e-02  2.8310e-03  29.7140 < 2.2e-16 ***
I(exp^2)    -8.0216e-04  6.2445e-05 -12.8459 < 2.2e-16 ***
bluecolyes  -4.6025e-02  1.6483e-02  -2.7922 0.0052586 **
unionyes     6.1649e-02  1.6945e-02   3.6381 0.0002780 ***
sexmale      3.3501e-01  5.2763e-02   6.3493 2.395e-10 ***
blackyes    -2.2062e-01  5.9925e-02  -3.6816 0.0002347 ***
ed           1.0166e-01  5.8281e-03  17.4433 < 2.2e-16 ***

Signif. codes:  0 '***' 0.001 '**' 0.01 '*'

Total Sum of Squares:    267.06
Residual Sum of Squares: 160
R-Squared      :  0.40089
Adj. R-Squared :  0.40002
F-statistic: 347.619 on 8 and 4156 DF, p-value: < 2.22e-16
```

其中, 与随机部分有关的估计为$\hat{\sigma}_\mu^2 = 0.02313$和$\hat{\sigma}_\epsilon^2 = 0.07755$. 固定效应部分的输出格式与通常回归类似.

为了探究该程序包的部分强大功能, 下面再做带有工具变量的回归. 在社会学和计量经济学领域中, 工具变量是在没有可控制的实验的情况下用来探讨回归中的因果关系的. 当某自变量与误差项相关时, 需要用工具变量来解决估计的无偏性和相合性问题. 工具变量应该和与因变量可能有因果关系的内生变量在给定其他自变量的条件下相关, 而且不能和回归的误差项相关. 这里不打算更多地讨论这个问题的细节. 统计学家通常认为因果关系只有在可干预情形下通过实验数据才有可能导出, 单纯从观测数据出发很难做任何因果推断.

这里以lwage为因变量, 以wks, south, smsa, married, exp及其平方, bluecol, ind, union, sex, black, ed为自变量, 以sex, black, bluecol, south, smsa, ind为工具变量(instrumental variables). 代码如下(在"|"后面的为工具变量, 这和其他某些程序包有所区别):

```
ht <- plm(lwage~wks+south+smsa+married+exp+I(exp^2)+
          bluecol+ind+union+sex+black+ed |
          sex+black+bluecol+south+smsa+ind,
          data=Wages,model="ht",index=595)
summary(ht)
```

输出结果为:

```
Oneway (individual) effect Hausman-Taylor Model
Call:
pht(formula=lwage~wks+south+smsa+ married + exp + I(exp^2) +
    bluecol + ind + union + sex + black + ed | sex + black +
    bluecol + south + smsa + ind, data = Wages, index = 595)

T.V. exo   : bluecol,south,smsa,ind
T.V. endo  : wks,married,exp,I(exp^2),union
T.I. exo   : sex,black
T.I. endo  : ed

Balanced Panel: n=595, T=7, N=4165

Effects:
                  var std.dev share
idiosyncratic 0.02304 0.15180 0.025
individual    0.88699 0.94180 0.975
theta:  0.9392

Residuals :
    Min.  1st Qu.   Median  3rd Qu.     Max.
-1.92000 -0.07070  0.00657  0.07970  2.03000

Coefficients :
               Estimate  Std. Error t-value  Pr(>|t|)
(Intercept)  2.7818e+00  3.0765e-01  9.0422 < 2.2e-16 ***
wks          8.3740e-04  5.9973e-04  1.3963   0.16263
southyes     7.4398e-03  3.1955e-02  0.2328   0.81590
smsayes     -4.1833e-02  1.8958e-02 -2.2066   0.02734 *
marriedyes  -2.9851e-02  1.8980e-02 -1.5728   0.11578
```

```
exp           1.1313e-01  2.4710e-03 45.7851 < 2.2e-16 ***
I(exp^2)     -4.1886e-04  5.4598e-05 -7.6718 1.696e-14 ***
bluecolyes   -2.0705e-02  1.3781e-02 -1.5024   0.13299
ind           1.3604e-02  1.5237e-02  0.8928   0.37196
unionyes      3.2771e-02  1.4908e-02  2.1982   0.02794 *
sexmale       1.3092e-01  1.2666e-01  1.0337   0.30129
blackyes     -2.8575e-01  1.5570e-01 -1.8352   0.06647 .
ed            1.3794e-01  2.1248e-02  6.4919 8.474e-11 ***

Signif. codes:0 '***' 0.001 '**' 0.01 '*'  0.05 '.'  0.1 ' ' 1

Total Sum of Squares:    886.9
Residual Sum of Squares: 95.947
F-statistic: 2852.33 on 12 and 4152 DF, p-value: < 2.22e-16
```

上面结果还指出了哪些是外生变量(exogenous variable, 这里为工具变量), 哪些是内生变量(endogenous variable, 这里为没有列在工具变量名单中的变量), 以及是不是随时间变化(time variable 或time invariable), 它们的4个组合分别用缩写T.V. exo, T.I. exo, T.V. endo, T.I. endo表示.

这里只为读者介绍了plm程序包的很少一部分功能. 希望能够引起有关领域读者的兴趣, 对其做更深入的了解, 从而让该程序包更充分地发挥作用.

第八章　多元分析

本书的所有内容都是关于多元数据的, 这里的标题主要是为了将就经典"多元分析"的名称和内容. 前面分析数据的问题都包含因变量和自变量, 并根据数据训练出预测模型, 这里的分析问题不区分因变量和自变量.

经典的多元分析以数据的正态性假定为基础, 内容包括聚类分析、主成分分析及因子分析、典型相关分析、判别分析. 有的教科书也包括不那么经典的对应分析. 前面章节已经讨论过判别分析及机器学习分类方法, 因此本章就不涉及分类(判别)问题了. 本章的图形很多, 图中有些文字太小, 不易分辨, 但由于都给出了R的制图代码, 因此图形小些不是问题, 读者可以自己绘制所有图形.

这里首先回顾一下经典多元分析方法, 然后介绍功能强大而且适应性及可视性都很强的新多元分析方法.

8.1　实数轴上的数据: 经典多元分析内容回顾

经典多元分析要求变量都是实数轴上的数量变量, 而且大多数方法都要求这些变量有联合正态分布, 以做各种检验和推断. 当然, 变量中不能有分类变量.

8.1.1　主成分分析及因子分析

主成分分析及因子分析的目的都是降维, 就是把相关的变量数目减少, 用较少的变量来取代原始变量, 而这些新变量为原始变量的线性组合. 如果原始变量互相正交, 那么这两种方法都不可用. 这两种方法的区别不在于数据或研究目的, 而在于数学模型.

一般来说, 因子分析的效果有可能优于主成分分析. 主成分分析的优点是原理简单, 计算容易. 因子分析实际上是后面要引进的图模型或隐变量模型的特例. 由于有很强的关于正态性及协方差矩阵的假定, 因子分析甚至可以在只有样本协方差矩阵而没有原始数据的条件下进行, 这与用协方差解图模型的方法一致.

当然, 较强的假定往往使得模型与现实距离更远, 也更有理由对结果提出质疑. 在做主成分分析及因子分析时, 最主观、对结果影响最大的就是选择哪些变量参与建模, 这方面怎么强调都不过分.

本小节用一个洛杉矶街区的例子来说明这两种方法的应用.

例 8.1 洛杉矶街区数据(LA.Neighborhoods.csv) 这是美国普查局2000年的数据.[1] 一共有110个街区, 15个变量. 变量情况见表8.1. 表中的API为涉及学生成绩的Academic Performance Index的缩写.

表 8.1 例8.1 洛杉矶街区数据变量情况

变量名	描述	性质	变量名	描述	性质
LA.Nbhd	街区名字	分类	Black	非裔比例	数量
Income	收入中位数	数量	Latino	拉美裔比例	数量
Schools	公立学校API中位数	数量	White	欧裔比例	数量
Diversity	种族多样性(0-1打分)	分类/定序	Population	人口	数量
Age	年龄中位数	数量	Area	面积	数量
Homes	有房家庭比例	数量	Longitude	经度	数量
Vets	复员军人比例	数量	Latitude	纬度	数量
Asian	亚裔人口比例	数量			

后面将要对这个数据进行各种分析, 并介绍相应的方法.

1. 主成分分析

假定原始数据有p个感兴趣的变量(用向量$\boldsymbol{x}$表示), 主成分分析的原理是在原始变量组成的p维空间中找到方差最大的方向, 也就是代表性最强的方向, 这等价于寻找线性组合$\boldsymbol{a}^\top\boldsymbol{x}$的系数$\boldsymbol{a}$, 使得$Var(\boldsymbol{a}^\top\boldsymbol{x}) = \boldsymbol{a}^\top\boldsymbol{R}\boldsymbol{a}$最大. 式中, $\boldsymbol{R}$为协方差矩阵. 在线性代数中, 这等价于求$\boldsymbol{R}$ 的最大特征值所对应的特征向量. 称这个向量为第一主成分, 记该方向的单位向量为$\boldsymbol{a}_1$; 类似地, 在与$\boldsymbol{a}_1$正交的空间, 再找使得方差最大的单位向量$\boldsymbol{a}_2$, 如此下去, 可以找到p个主成分以及对应于$\boldsymbol{R}$的p个特征值$\lambda_1, \lambda_2, ..., \lambda_p$ $(\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_p)$的$p$个特征向量$\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_p$. 为叙述方便, 不妨假定这些$\boldsymbol{a}_i$为单位向量.

第i个主成分的贡献率为$\lambda_i/\sum_{j=1}^{p}\lambda_j$, 前$k$个主成分的累积贡献率为$\sum_{j=1}^{k}\lambda_i/\sum_{j=1}^{n}\lambda_j$. 由于协方差矩阵$\boldsymbol{R}$是未知的, 因此用样本协方差矩阵或样本相关阵来代替. 第i个主成分与第j个原始变量之间的相关系数为$\sqrt{\lambda_i}a_{ij}$, 这里a_{ij}是$\boldsymbol{a}_i$的第j个分量, 当然, 向量$\sqrt{\lambda_i}\boldsymbol{a}_i$ 和$\boldsymbol{a}_i$是同方向的特征向量, 只不过$\boldsymbol{a}_i$是单位向量而已. 在一些软件的输出中(比如SPSS)称$\sqrt{\lambda_i}\boldsymbol{a}_i$为载荷(loading), 但在另一些软件(比如R的`princomp()`函数及SAS)的输出中, 称$\boldsymbol{a}_i$为载荷, 这没有多大关系.

[1]数据网站: http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_LA_Neighborhoods_Data.

如果用$y_1, y_2, ..., y_p$表示p个主成分, 用$x_1, x_2, ..., x_p$表示原始变量, 那么它们之间的关系为:

$$\begin{aligned} y_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1p}x_p \\ y_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2p}x_p \\ &\vdots \\ y_p &= a_{p1}x_1 + a_{p2}x_2 + \cdots + a_{pp}x_p. \end{aligned}$$

下面用根据原始数学公式求特征值问题的R代码来计算主成分及诸如贡献率、累积贡献率等各种指标(图8.1和相应的表8.2).

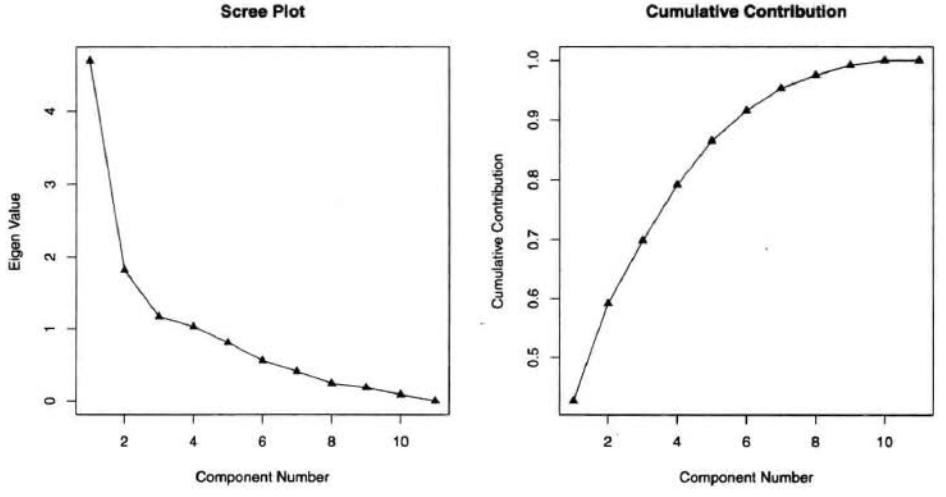


图 8.1 例8.1的主成分分析崖底碎石图(贡献率图, 左)和累积贡献率图(右)

```
w=read.csv("LA.Neighborhoods.csv") #读入数据
w$density=w$Population/w$Area #增加人口密度变量
u=w[,-c(12:15)]#去掉人口、面积、经纬度变量
a=eigen(cor(scale(u[-1]))) #解标准化后的数据的特征值问题
(cca=(a$va)/sum(a$va))#贡献率
(ca=cumsum(a$va)/sum(a$va))#累积贡献率
par(mfrow=c(1,2))#画贡献率图(崖底碎石图)及累积贡献率图
plot(1:11,a$va,type="o", pch=17,col=4, main="Scree Plot",
    xlab="Component Number",ylab="Eigen Value")
```

```
plot(1:11,ca,type="o",pch=17,col=4,
    main="Cumulative Contribution",
    xlab="Component Number",ylab="Cumulative Contribution")
par(mfrow=c(1,1))
```

表 8.2 例8.1的主成分分析的特征值、贡献率、累积贡献率

成分	1	2	3	4	5	6	7	8	9	10	11
特征值	4.69	1.82	1.17	1.03	0.81	0.56	0.41	0.24	0.19	0.09	0.00
贡献率	0.43	0.17	0.11	0.09	0.07	0.05	0.04	0.02	0.02	0.01	0.00
累积贡献率	0.43	0.59	0.70	0.79	0.86	0.92	0.95	0.97	0.99	1.00	1.00

图8.1是根据表8.2中的数据绘出的, 可以看出, 第一主成分的贡献率较大, 而其他主成分的贡献率都不那么大, 一直到第4主成分时, 累积贡献率才超过75%. 贡献率图之所以叫崖底碎石图(scree plot), 是因为希望图形一开始很陡, 如悬崖一般, 而剩下的数值都很小, 如悬崖下的碎石一样. 但这个图中悬崖只有部分是陡的, 说明用主成分分析方法降维的效果并不是特别突出.

所有的方法都有其局限性, 可能对某些数据有效, 而对另一些数据不那么有效. 对于基本独立的变量就无法做主成分分析. 笔者曾经在一次研讨会上看到有人极其荒谬地对通过主成分分析所选出来的成分再做主成分分析, 而且声称结果很好, 并作为经验介绍. 笔者当时没有机会来提问, 但愿以后再也不会听到类似的"经验"了.

下面的代码计算(特征值都大于1的)头四个主成分和原始变量的相关系数(载荷), 其中第一行就是计算相关系数的语句, 所用的公式就是前面提到的$\sqrt{\lambda_i}a_{ij}$, 即对每个特征向量(a$vec)乘以相应的特征值的平方根(sqrt(a$value)); 后面语句作出四个主成分与原始变量的相关系数表(表8.3), 绘出载荷图(图8.2).

```
(loadings=sweep(a$vec,2,sqrt(a$value),"*"))->b #计算载荷
par(mfrow=c(1,2))
plot(b[,1:2], type="n", xlab="Component 1 (42%)",
    ylab="Component 2 (17%)",
    xlim=c(-1.2,1.2),ylim=c(-1.2,1.2),main="Loadings")
text(b[,1],b[,2],names(u[,-1]));abline(h=0);abline(v=0)
plot(b[,3:4], type="n", xlab="Component 3 (11%)",
    ylab="Component 4 (9%)", xlim=c(-1.2,1.2),
    ylim=c(-1.2,1.2),main="Loadings")
text(b[,3],b[,4],names(u[,-1]))
```

```
abline(h=0);abline(v=0)
par(mfrow=c(1,1))
```

表 8.3 例8.1的四个主成分与原始变量的相关系数

	主成分			
原始变量	1	2	3	4
Income	−0.8852	0.0657	0.2212	−0.0301
Schools	−0.0662	−0.3460	0.1051	0.7980
Diversity	0.0993	−0.6771	−0.6012	−0.0060
Age	−0.9247	−0.0489	−0.1288	−0.1779
Homes	−0.6854	−0.0650	−0.1261	0.3256
Vets	−0.7957	0.2810	−0.3606	0.1356
Asian	−0.0526	-0.8584	−0.1722	−0.1945
Black	0.3156	0.6302	−0.6498	0.0871
Latino	0.8320	−0.0535	0.2478	0.2262
White	−0.8752	−0.0901	0.2674	−0.1865
density	0.6136	−0.0686	−0.0101	−0.3191

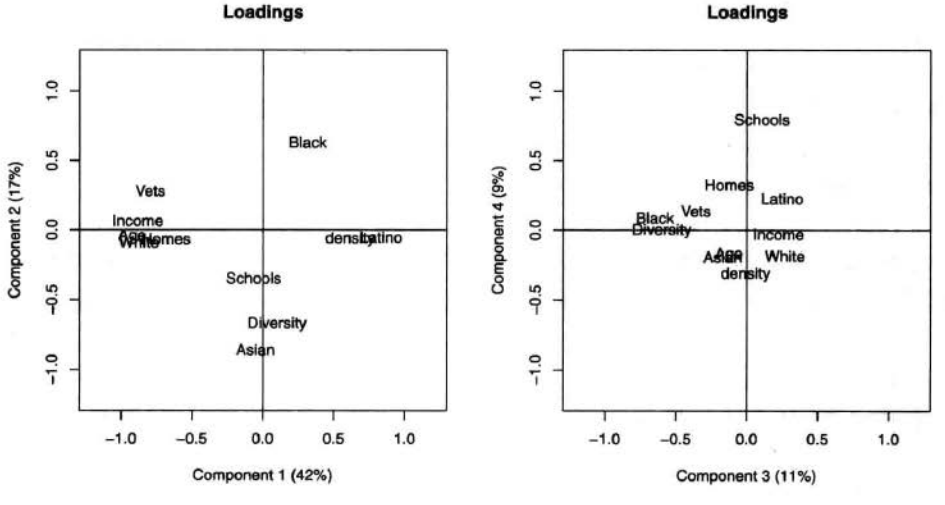


图 8.2 例8.1的主成分分析载荷图

载荷图图8.2的左图对应于例8.1的四个主成分与原始变量的相关系数表(表8.3)的第一、二列, 可以看出, 第一主成分小(负值)的对应于收入高、年

龄大、拥有住房的家庭多、复员军人多及欧裔多的社区, 第一主成分大的则对应于拉美裔人口多、人口密度高的社区. 第二主成分大的对应于非裔较多的社区, 第二主成分小(负值)的则对应于亚裔人口较多、种族多样性较高的社区. 其他变量和这两个主成分相关性不大. 图8.2的右图对应于主成分与原始变量相关系数表中的第三、四列, 可以看出, 这里的点相对左图更接近中间(相关系数接近0). 第三主成分小(负值)的对应于非裔较多、种族多样性较高的社区, 第四主成分大的对应于学校成绩较好的社区.

对于每个观测值都可以得到它们关于每个主成分的得分. 下面的语句计算了头四个主成分的得分, 即用标准化的原始数据矩阵乘特征向量矩阵, 并把这些得分展示在图8.3中. 图中各个点是用街区名字代表的, 虽然有些拥挤(特别是右图), 但还是可以看出一些特别突出的街区名字.

```
par(mfrow=c(1,2))
sc=as.matrix(scale(u[,-1]))%*%a$ve[,1:4] #计算得分
plot(sc[,1],sc[,2],type="n",ylim=c(-6,6),xlim=c(-7,6),
    main="Sample Principal Components", xlab="Component 1",
    ylab="Component 2")
text(sc[,1],sc[,2],u[,1],cex=.4)
abline(v=0,col=2);abline(h=0,col=2)
plot(sc[,3],sc[,4],type="n",ylim=c(-6,6),xlim=c(-7,6),
    main="Sample Principal Components", xlab="Component 3",
    ylab="Component 4")
text(sc[,3],sc[,4],u[,1],cex=.4)
abline(v=0,col=2);abline(h=0,col=2)
par(mfrow=c(1,1))
```

相信读者只要把图8.2和图8.3对照起来, 就可以看出各个街区的特点了, 比如哪些街区属于较富裕的, 哪些街区种族多样性高, 等等. 对于一些数据, 有可能可以根据这些特点给各个主成分起名字, 但有时候很难说清楚这些主成分的意义, 更谈不上取名字了.

有些人把主成分分析和因子分析中的成分或因子作为综合评价的基础, 比如, 某个成分为"教育成分", 某个成分为"经济发展成分", 等等, 然后用这些成分或者它们的加权平均作为"政绩"的综合评价标准, 并用来排序. 但是, 如果分析时不仅仅局限于经济变量, 还包括廉政、环境、公平、效率等其他变量, 结果就不那么简单了. 这时, "经济发展成分"和腐败及污染很可能都十分相关, 这时如何称呼, 如何进行综合评价都会成为问题. 比如, 公平和效率也是负相关的, 反映效率的成分难道可以称为"不公平成分"? 因此, 进行主成分分析或因

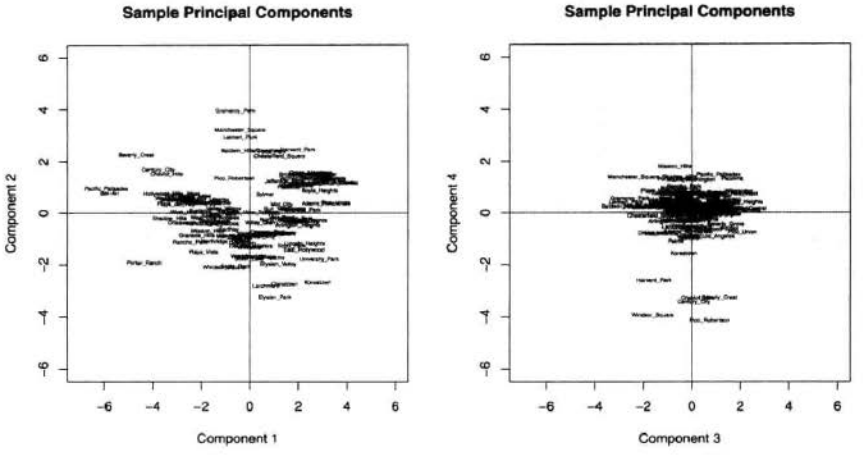


图 8.3 例8.1的样本主成分得分

子分析最重要的是客观地、毫无偏见地选择变量.

2. 因子分析

因子分析也是寻找原始变量的一些不相关的线性组合, 称为因子, 因子数量m比原始变量的数量p小, 即$m < p$. 假定$f_1, f_2, ..., f_m$为需要找到的因子, 那么它们和原始变量$x_1, x_2, ..., x_p$之间的关系为:

$$\begin{aligned} x_1 - \mu_1 &= a_{11}f_1 + a_{12}f_2 + \cdots + a_{1m}f_m + \epsilon_1 \\ x_2 - \mu_2 &= a_{21}f_1 + a_{22}f_2 + \cdots + a_{2m}f_m + \epsilon_2 \\ &\vdots \\ x_p - \mu_p &= a_{p1}f_1 + a_{p2}f_2 + \cdots + a_{pm}f_m + \epsilon_p, \end{aligned}$$

或者以矩阵形式表示为:

$$\boldsymbol{x} - \boldsymbol{\mu} = \boldsymbol{A}\boldsymbol{f} + \boldsymbol{\epsilon},$$

式中, μ_i为变量x_i的均值; ϵ_i称为第i个特殊因子, 只与x_i有关; 而f_i称为第i个公共因子, 与所有变量有关; a_{ij}为第i个变量在第j个因子上的载荷, 也就是第i个变量与第j个公共因子的相关系数. 不能观测的值假定满足下列条件: $\boldsymbol{f}$和$\boldsymbol{\epsilon}$独立, $E(\boldsymbol{f}) = \boldsymbol{0}$, $Cov(\boldsymbol{f}) = \boldsymbol{I}$, $E(\boldsymbol{\epsilon}) = \boldsymbol{0}$, $Cov(\boldsymbol{\epsilon}) = \boldsymbol{\Psi}$, $\boldsymbol{\Psi}$ 是对角矩阵. $\sum a_{ij}^2$称为共性方差(也称公共方差(common variance)或变量共同度(communalities)), 而$\boldsymbol{\Psi}^2$称为特殊方差. 变量共同度刻画全部公共因子对变量x_i的总方差所做的贡献. 根据这些假定、代数的知识及计算方法, 可以得到

对a_{ij}的矩阵$\boldsymbol{A}$的解, 但这个解不唯一, 解之间相差一个正交变换或刚体旋转. 因此方差分析还需要在这些解中寻找一个最能够解释数据的解. 这也是因子分析多了旋转这个步骤的原因.

下面的代码利用了R固有的函数`factanal()`来做例8.1数据的因子分析(取$m = 2$), 并计算出载荷及样本因子得分. 同时画出前两个因子的载荷图(图8.4左) 和样本因子得分图(图8.4右). 而这里的因子得分不像主成分分析那样自然, 它必须用诸如回归等方法反解出来, 这需要在函数`factanal()`的选项中指明, 下面的程序表明要用`scores="regression"`来得到得分. 得分的表达式为:

$$\begin{aligned} f_1 &= \beta_{11}x_1 + \beta_{12}x_2 + \cdots + \beta_{1p}x_p \\ f_2 &= \beta_{21}x_1 + \beta_{22}x_2 + \cdots + \beta_{2p}x_p \\ &\vdots \\ f_m &= \beta_{m1}x_1 + \beta_{m2}x_2 + \cdots + \beta_{mp}x_p, \end{aligned}$$

式中, $m < p$. 这里用回归等方法算出的β_{ij}不再是相关系数了. 在读任何软件输出时都要注意这一点, 以免与主成分分析的输出格式混淆.

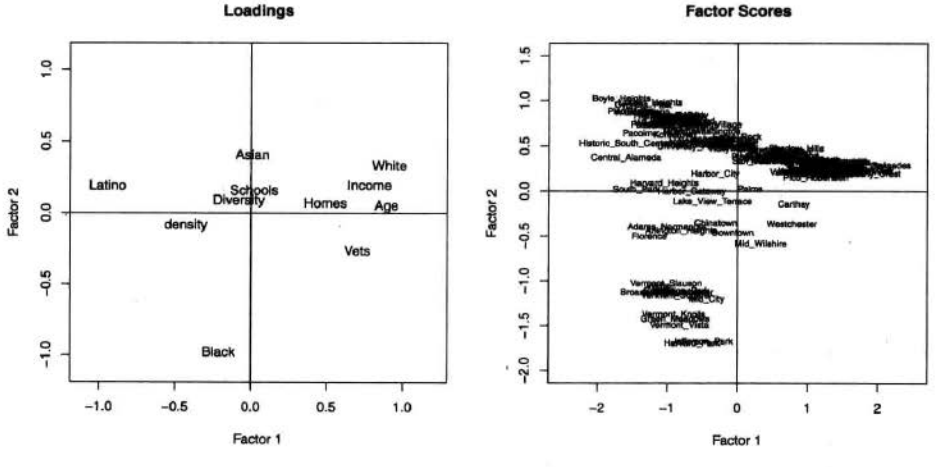


图 8.4　例8.1的两个因子的载荷(左)与样本因子得分(右)

```
#下面进行因子分析时默认旋转rotation ="varimax":
w=read.csv("LA.Neighborhoods.csv") #读入数据
w$density=w$Population/w$Area #增加人口密度变量
```

```
u=w[,-c(12:15)]#去掉人口、面积、经纬度变量
(a=factanal(factors=2,scale(u[,-1]),scores="regression"))
(a$loadings);(a$scores)#打印载荷和得分
par(mfrow=c(1,2))#作因子载荷与样本因子得分图
plot(a$loadings[,1:2],type="n",ylim=c(-1.1,1.1),
    xlim=c(-1.1,1.2),
    xlab="Factor 1", ylab="Factor 2", main="Loadings")
abline(h=0);abline(v=0)
text(a$loadings[,1],a$loadings[,2],
labels=row.names(a$loadings),cex=1)
plot(a$scores[,1:2],type="n",ylim=c(-2,1.5),xlim=c(-2.5,2.5),
    xlab="Factor 1", ylab="Factor 2", main="Factor Scores")
abline(h=0);abline(v=0)
text(a$scores[,1],a$scores[,2],labels=u[,1],cex=.7)
```

在做主成分分析时, 可以事先不确定主成分的个数, 只需要在计算之后, 根据特征值的大小及贡献率来决定选择几个主成分. 但在因子分析中, 事先选择不同的因子个数, 可能会造成很不一样的结果. 就例8.1来说, 如果把上面程序中的因子个数改成4(factors=4), 所得到的头两个因子的载荷图及因子得分图与只选两个因子的情况就很不一样. 希望读者自己来体验.

下面为上面程序中命令(a$loadings)(载荷)的打印输出(其中数值小于0.1的就自动用空白代替), 它是例8.1的第一、二个因子与原始变量的相关系数(载荷), 分别对应于图8.4(左)的横坐标和纵坐标.

```
Loadings:
          Factor1 Factor2
Income     0.781   0.197
Schools            0.163
Diversity
Age        0.893
Homes      0.491
Vets       0.702  -0.263
Asian              0.408
Black     -0.212  -0.975
Latino    -0.944   0.195
White      0.913   0.334
density   -0.428
```

综合这个输出和图8.4(左)可以看出, 第一因子大(正方向)的对应收入高、年龄大、拥有房子的家庭多、复员军人多及欧裔多的社区, 相反方向则对应拉美裔人口多、人口密度高的社区. 第二因子主要描述了Black变量(在第二因子的负方向).

在因子得分图(图8.4右)中各个点是用街区名字代表的, 从图中可以看出各种街区所在的位置, 也可以识别出一些特别突出的街区. 因子得分的打印结果太长, 还是图8.4(右)更清楚.

8.1.2 分层聚类及k均值聚类

数据有几个变量, 就形成几维空间, 每个观测值都是该空间的一个点. 聚类分析就是根据点之间的距离, 把离得近的一些点划归为一类, 原则是使得类间距离尽可能大, 而类内点间的距离尽可能小. 现在一般教科书上都介绍的聚类方法是分层聚类和k均值聚类. 下面就用例8.1洛杉矶街区数据来回顾这两种方法. 在此之前先介绍两种确定聚类个数的程序.

1. 聚类数目的选择

后面要介绍的k均值聚类必须事先确定聚多少类才能计算, 而分层聚类虽然不要求事先确定分多少类, 但必须要自己在各种可能的结果中确定分多少类. 下面介绍两种确定聚类个数的方法, 它们是基于k均值聚类的.

第一种方法是利用程序包`cluster`[1]的函数`clusGap()`计算聚类度量的优度, 即Gap统计量. 对每个类数目k, 它比较$\ln(W(k))$和$E^*[\ln(W(k))]$, 后者由自助法定义, 即根据参考分布来做模拟.

寻找最大值之所在要利用"1-SE准则". 默认的方法为"firstSEmax", 寻求使得局部最大值在一个标准误差之内的最小的k. 这和"Tibs2001SEmax"方法类似但又有所不同. Tibshirani et al. 推荐由Gap统计量和它们的标准差来确定类的数目.

我们先用R软件自带的数据faithful来说明. 数据faithful是黄石公园著名的"老忠实"(Old Faithful)间歇式温泉喷发持续时间(eruptions)和喷发之间的等待时间(waiting)的记录.

使用第一种方法的代码如下:

```
library(cluster)
(z=clusGap(faithful, FUN=kmeans, 10))
```

得到输出:

[1] Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.(2015). cluster: Cluster Analysis Basics and Extensions. R package version 2.0.1.

```
Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 100)  [one "." per sample]:
.................................................. 50
.................................................. 100
Clustering Gap statistic ["clusGap"].
B=100 simulated reference sets, k = 1..10
--> Number of clusters (method 'firstSEmax', SE.factor=1):2
          logW   E.logW       gap     SE.sim
 [1,] 6.953597 7.089672 0.1360751 0.02828301
 [2,] 6.096229 6.407273 0.3110443 0.02365501
 [3,] 5.808396 6.004685 0.1962899 0.02648897
 [4,] 5.530438 5.727977 0.1975388 0.02460236
 [5,] 5.342815 5.512082 0.1692671 0.02939966
 [6,] 5.201743 5.351821 0.1500774 0.03193237
 [7,] 5.099751 5.221105 0.1213541 0.03602503
 [8,] 4.926106 5.112029 0.1859229 0.04370349
 [9,] 4.876992 5.022500 0.1455079 0.04603757
[10,] 4.747445 4.946601 0.1991561 0.05661983
```

其中, Number of clusters (method 'firstSEmax',SE.factor=1): 2说明利用firstSEmax方法选择2个类.

可以对Gap统计量画出图来, 并利用后面要介绍的k均值聚类来点出对于faithful数据分类的效果图. 这里所用的k均值聚类方法为程序包cluster中的函数pam(), 它比通常的k均值聚类方法更加稳健. 根据刚才算出的结果绘制Gap图及k均值聚类的具体代码如下:

```
a=pam(faithful,2)#more robust version of K-means.
par(mfrow=c(1,2))
plot(z, main = "Gap statistic for the Faithful data")
plot(faithful, col=4, type="n")
points(faithful[a$clust==1, ], col=2, pch=16)
points(faithful[a$clust==2, ], col=4, pch=15)
legend("topleft",c("Cluster 1","Cluster 2"),col=c(2,4),
    pch=16:15)
```

图8.5的左图显示在位置2时(类数为2), Gap统计量达到极大值, 而且在极大值的一个标准差的距离之内除了2之外没有别的点, 这就是firstSEmax方法的直观效果. 图8.5的右图为对faithful数据聚2类的效果.

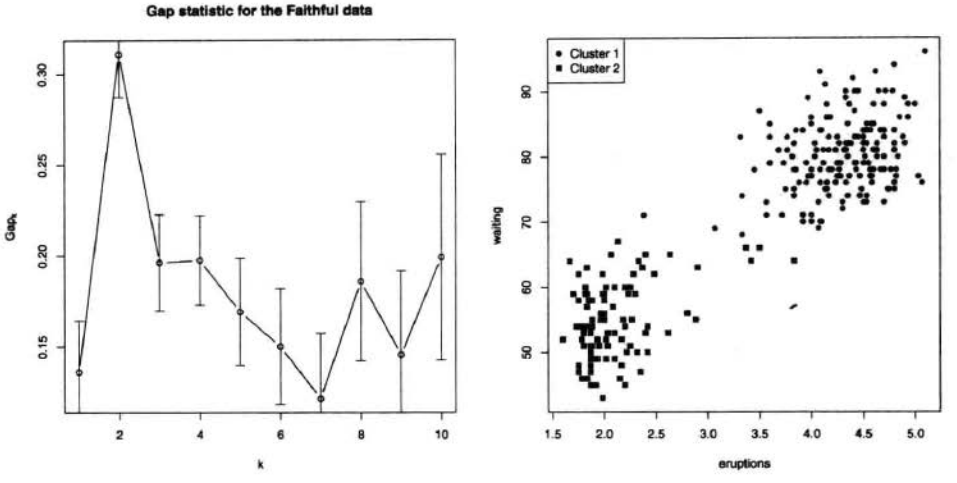


图 8.5 faithful数据的Gap统计量(左)和k均值聚类效果(右)

第二种方法是利用所谓的INCA指数, 具体方法参见Arenas and Cuadras (2002)和Irigoien and Arenas (2008). 我们所用的程序包为ICGE.[1]

对于数据faithful, 使用下面代码计算并点出图(图8.6)来:

```
library(ICGE)
d=dist(faithful)
T <- rep(NA, 5)
for (i in 2:10)
{
part <- pam(d, i)$clustering
T[i] <- INCAindex(d, part)$Total
}
plot(T,  type="b", xlab="Number of clusters",
    ylab="INCA",  xlim=c(1.5,  10.5))
title("Percentage of objects well classified
    in the partition")
```

图8.6也显示, 最大值点在类数取2处, 也就是说, 应该分成两类, 和前面方法的结果一样.

[1]Itziar Irigoien and Concepcion Arenas (2014). ICGE: Estimation of number of clusters and identification of atypical units. R package version 0.3.http://CRAN. R-project.org/package=ICGE.

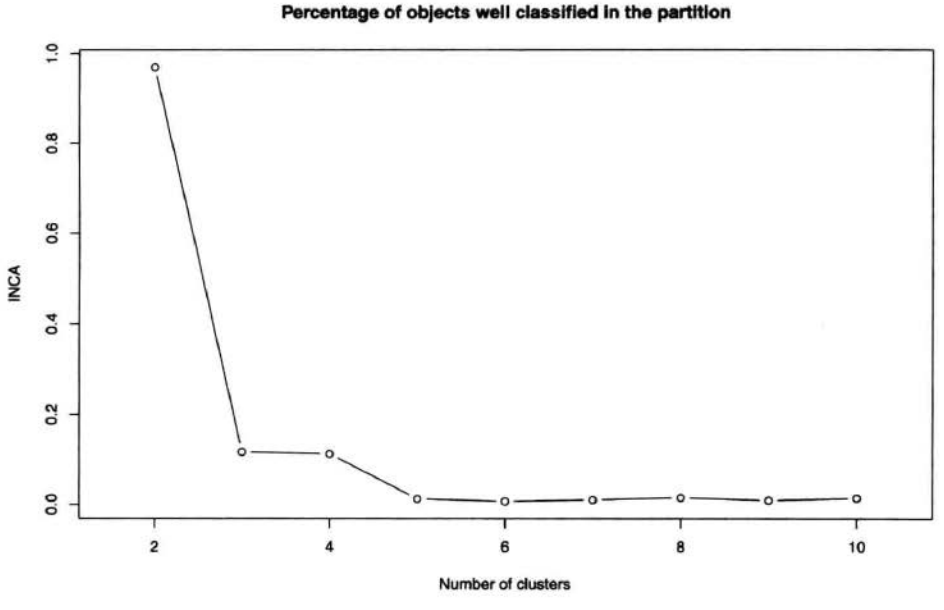


图 8.6 faithful数据的INCA指数

下面再对例8.1数据用上面两种方法来确定聚类的数目. 具体的计算和画图代码如下:

```
library(cluster)
#Clustering Gap statistic:
(z=clusGap(u[,-1], FUN=kmeans, 20))
library(ICGE)
d=dist(u[,-1]);m=5
T <- rep(NA, m)
for (i in 2:20){
part <- pam(d, i)$clustering
T[i] <- INCAindex(d, part)$Total}
par(mfrow=c(1,2))
plot(z, main = "Gap statistic for LA data")#图8.7(左)
plot(T, type="b", xlab="Number of clusters",
    ylab="INCA", xlim=c(1.5, 10.5))#图8.7(右)
title("Percentage of objects well classified
    in the partition")
```

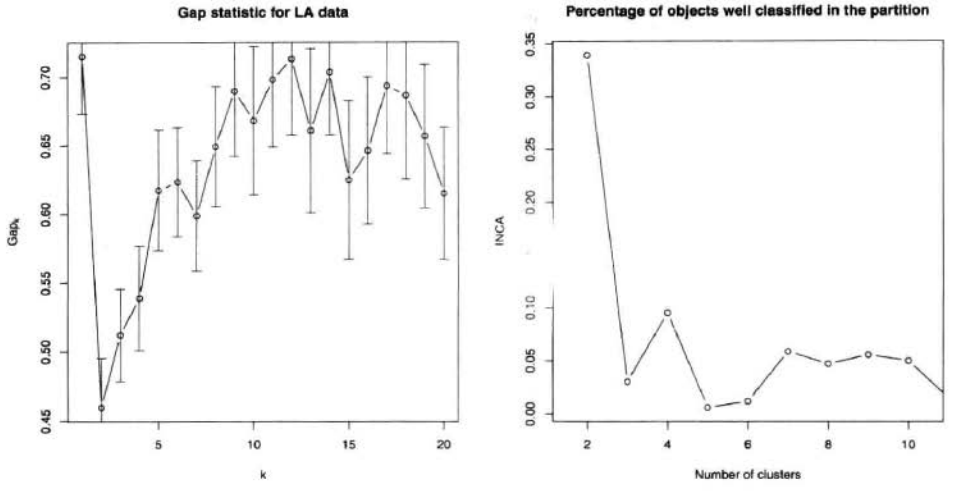


图 8.7 例8.1数据的Gap统计量(左)和INCA指数(右)

从图8.7的左图看来, 第一个局部极大值位于1, 如果不选择1, 则最近的极大值为6, 而在其一个标准差之内k的最小点是5, 因此, 可以选5类. 而右图的INCA指数的最大值则是2. 这就显示出各种方法的区别了. 分成5类或者2类都有道理. 人们可以根据不同的角度制定不同的数学指标, 说不定在分层聚类中用人眼直观判断会更加合适呢.

2. 分层聚类

聚类首先要确定点与点之间的距离, 这可以用欧氏距离、平方欧氏距离、绝对距离、夹角余弦、相关系数、Minkowski距离等. 此外, 还要定义类间距离, 采用的方法包括最短距离法、最长距离法、类平均法、重心法、离差平方和法等. 这些选项不同, 结果也可能不同. 当然, 选择参与聚类的变量是最重要的.

在确定了各种距离的选择之后, 就可以进行分层聚类了. 一开始每一个观测值点都算为一类, 这样, 有多少观测值就有多少类. 分层聚类首先把最近的两类(点)划分为一类, 这样总类数就少了一个, 然后把最近的两类划到一起, 这样类的数目又少了一个, 如此下去, 当只剩下一类时, 分层聚类就完成了. 在这过程中记录了所有类间的距离, 根据这些距离的远近, 可以画出树状图(dendrogram), 并根据图形决定分成几类.

下面就例8.1的数据做分层聚类, 代码如下:

```
w=read.csv("LA.Neighborhoods.csv")
```

```
w=data.frame(w,density=w$Population/w$Area)
u=w[,c(1,2,5,6,11,16)]#选择变量
#标准化数据，聚类方法="complete":
hh=hclust(dist(scale(u[,-1])), "complete")
plot(hh,labels=u[,1],cex=.6);id=identify(hh)#手工分成5份
```

由此得到树状图(图8.8). 其中有手工分成5类的标记, 也有每一类的街区名字. 根据所分类别和洛杉矶每个街区的经纬度, 把各个类用不同的符号点在图(图8.9)上, 画图代码如下:

```
ppp=c(7,17,19,21,24)
plot(w[id[[1]],14:15],pch=ppp[1],col=1,xlim=c(-118.7,-118.1),
    ylim=c(33.73,34.32),main="Los Angeles")
for(i in 2:5)
points(w[id[[i]],14:15],pch=ppp[i])
legend("bottomleft",pch=ppp,paste("Cluster",1:5))
```

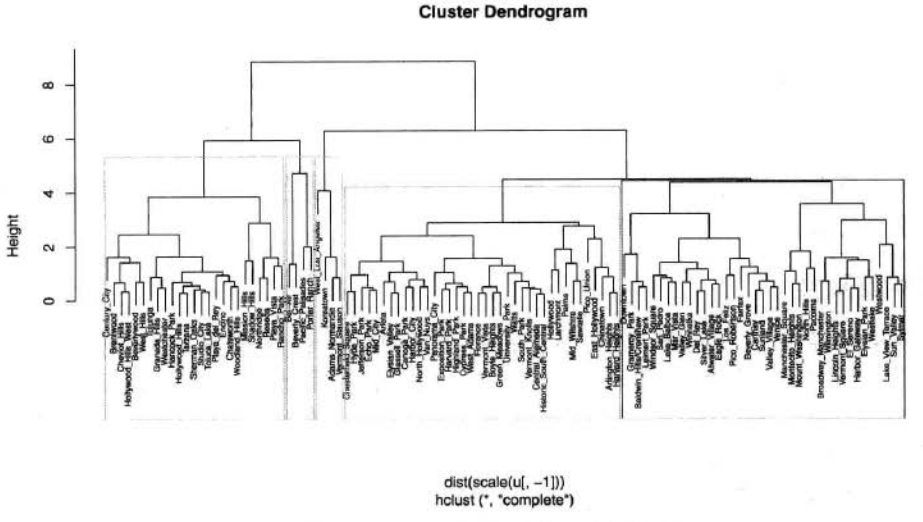


图 8.8 例8.1数据分层聚类树状图

对于分类地图(图8.9(左))中各类街区的地理位置及分布, 对洛杉矶比较熟悉的人可能会给出更加明确具体的解释. 图8.9(右)的洛杉矶地图是通过程序包**ggmap**[1]用下面语句产生的(计算机需要联网, 利用了Google Earth的资源):

[1]David Kahle and Hadley Wickham (2012). ggmap: A package for spatial visualization with Google Maps and OpenStreetMap. R package version 2.2. http://CRAN.R-project.org/package=ggmap.

```
library(ggmap)
hdf <- get_map(location = c(lon = -118.1, lat =33.9))#LA
ggmap(hdf)
```

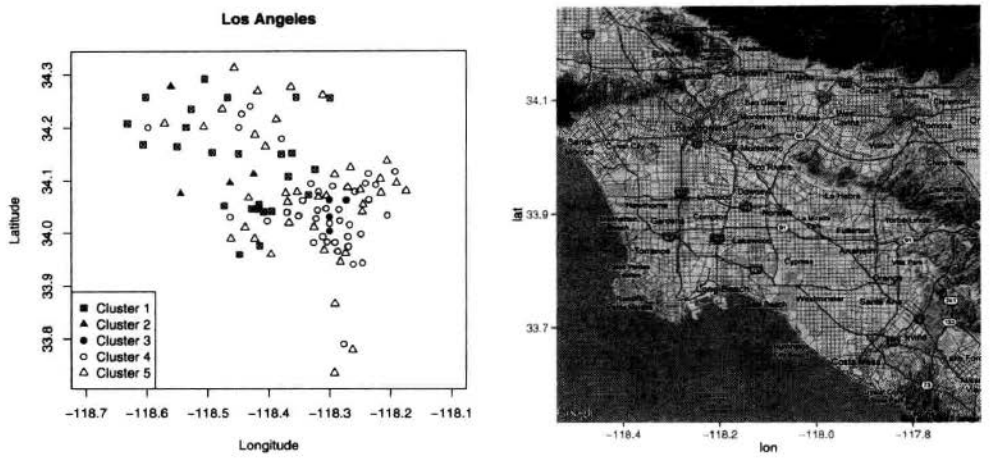


图 8.9 例8.1数据分层聚类所得5类街区按照经纬度在洛杉矶的点图(左)以及原地图(右)

分层聚类的树状图直观易懂, 容易帮助人们理解各类的特点和关系. 其缺点是对于大数据的计算速度相对较慢.

3. k均值聚类

较老的经典多元分析教科书上往往没有的k均值聚类又叫快速聚类. 假定有p个变量参与聚类, 如果要分为k类, 则在p个变量所形成的p维空间中选择k个不同的点, 称为种子, 然后按照各个观测值点距离这k个点的远近把所有观测值分成k类, 在每一类求出其中心, 作为k个新的种子, 再把所有观测值重新按照距离这k个种子的远近分成k类. 如此下去, 直到种子的位置基本不变为止. 初始种子的选取不必太认真, 通常计算机程序也会自动给出初始种子, 最终这些初始种子有可能会分到同一类中. 顾名思义, k均值聚类计算速度很快, 但其缺点是必须事先选择类别的数目. 目前也有利用一些准则自动选择类别数目的方法(见前面内容).

对例8.1数据做k均值聚类($k = 5$), 并像前面的图8.9那样按照经纬度把各个类用不同的符号点在图上(图没有在这里展示). 代码为:

```
a=kmeans(scale(u[,-1]),5);ppp=c(7,17,19,21)
```

```
plot(w[a$cluster==1,14:15],pch=1,col=1,xlim=c(-118.7,-118.2),
    ylim=c(33.73,34.32),main="Los Angeles")
for(i in 2:5)
points(w[a$cluster==i,14:15],pch=ppp[i-1])
legend("bottomleft",pch=c(1,ppp),paste("Cluster",1:4))
```

读者可以自己比较这两种聚类的异同. 实际上, k均值聚类的第4, 3, 1, 5, 2类大体上相当于前面分层聚类的1, 2, 3, 4, 5类(有27个点分的不同). 变量的选择、方法(包括细微选项)的选择以及是否把数据标准化都可以对聚类结果产生影响.

4. 含有聚类数目各种选择方法及各种聚类方法的综合程序包

这里介绍的聚类数目的选择实际上不是一种方法, 而是集合在一个程序包中的一系列方法, 也包括前面的Gap统计量. 该程序包为NbClust.[1] 它提供了30个指数用于确定在数据集中的最优聚类数目, 并根据不同的结果给用户提供最佳的聚类方案. 它不仅确定聚类数目, 而且根据选项实行具体的聚类, 包括前面的k均值聚类.

NbClust提供的聚类所基于的距离选择选项有

"euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski", "NULL".

如果选了NULL, 则必须选择不相似矩阵(如距离矩阵), 默认值为euclidean(欧氏距离). 它提供的方法有

"ward.D", "ward.D2", "single", "complete", "average", "mcquitty", "median", "centroid", "kmeans".

而判别数目的判别指标选项包括

"kl", "ch", "hartigan", "ccc", "scott", "marriot", "trcovw", "tracew", "friedman", "rubin", "cindex", "db", "silhouette", "duda", "pseudot2", "beale", "ratkowsky", "ball", "ptbiserial", "gap", "frey", "mcclain", "gamma", "gplus", "tau", "dunn", "hubert", "sdindex", "dindex", "sdbw", "all", "alllong".

其中, all指标包括除gap, gamma, gplus, tau之外的所有指数, 而alllong包括了all所没有包括的所有指标. 我们就例8.1(除了街区名字

[1] Malika Charrad, Nadia Ghazzali, Veronique Boiteau, Azam Niknafs (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *Journal of Statistical Software*, 61(6), 1-36. URL http://www.jstatsoft.org/v61/i06/.

和经纬度之外的原始数据)来说明这个强大程序包的应用. 下面是代码和部分输出(这里选了指标all):

```
library(NbClust)
a<-NbClust(w[,-c(1,14:15)], distance = "euclidean", min.nc=2,
max.nc=8,  method = "complete", index = "all")
*******************************************************************
* Among all indices:
* 7 proposed 2 as the best number of clusters
* 5 proposed 3 as the best number of clusters
* 6 proposed 4 as the best number of clusters
* 5 proposed 7 as the best number of clusters
* 1 proposed 8 as the best number of clusters

                   ***** Conclusion *****
* According to the majority rule,
    the best number of clusters is  2
*******************************************************************
```

输出表明, 关于聚类数目, 有8个指标说选2个, 有7个指标说选6个, 有4个指标说选3个, 另外有3个指标说选7个, 1个指标说选4个, 1个指标说选8个(也就是这里指定的最大数目max.nc=8), 看来很不一致. 输出按照"少数服从多数"标出了"最好的"聚类数目是2个. 还可以输出相应于2~8个聚类的所有指标(a$All.index)、某些指标(这里是3个)对于2~8个聚类分划的边界值(a$All.CriticalValues), 以及相应于"最好的"聚类数目的指标值(a$Best.nc). 最后还给出了说明各个观测值划分到哪一类的列表(Best.partition). 这个"少数服从多数"的结果把例8.1的原始数据(除了街区名字和经纬度)分成两类, 有一类只有3个成员, 而剩下的107个全部分到另一类. 我们也试了标准化之后的数据, 结果有9个指标选择分成两类.

无论如何, 在使用这个程序包时, 要充分利用其大量的选项来选择与问题相关的具体选项, 不一定使用其"少数服从多数"的自动化功能.

8.1.3 典型相关分析

数理统计教科书中都涉及单独两个变量之间的相关问题, 而这里的典型相关分析(canonical correlation analysis)则是研究两组变量之间的相关问题. 其方法是, 在两组变量中各自找一个代表, 即各自变量的线性组合, 使得这一对代表比其他对的代表更加相关, 然后在两组变量与这第一对代表正交的空间

中再各自找另一对最相关的代表, 如此下去, 直至找不到符合要求的代表为止. 其中使用的方法就是线性代数的特征值—特征向量及Lagrange乘数法. 每一对线性组合都称为典型变量(canonical variable). 它们之间的相关称为典型相关(canonical correlation). 假定两组变量为: $\boldsymbol{X}=(X_1,X_2,...,X_p)^\top$及$\boldsymbol{Y}=(Y_1,Y_2,...,Y_q)^\top$, 我们的目的是寻找系数$a_1,a_2,...,a_p$及$b_1,b_2,...,b_q$, 使得新的综合变量, 即典型变量

$$\begin{aligned} V &= a_1X_1+a_2X_2+\cdots+a_pX_p \\ W &= b_1Y_1+b_2Y_2+\cdots+b_qY_q \end{aligned}$$

之间的典型相关系数(canonical correlation coefficient)最大. 假定$(\boldsymbol{X},\boldsymbol{Y})^\top$的协方差矩阵分块为:

$$\boldsymbol{\Sigma}=\begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix},$$

可得

$$Cov(V,W)=\boldsymbol{a}^\top\boldsymbol{\Sigma}_{12}\boldsymbol{b}.$$

假定$Var(V)=1$, $Var(W)=1$, 使得相关系数(这时等于协方差)$\rho_{VW}=Cov(V,W)=\boldsymbol{a}^\top\boldsymbol{\Sigma}_{12}\boldsymbol{b}$最大的$\boldsymbol{a}$和$\boldsymbol{b}$为下式极大值问题的解:

$$\phi=\boldsymbol{a}^\top\boldsymbol{\Sigma}_{12}\boldsymbol{b}-\frac{\lambda}{2}(\boldsymbol{a}^\top\boldsymbol{\Sigma}_{11}\boldsymbol{a}-1)-\frac{\nu}{2}(\boldsymbol{b}^\top\boldsymbol{\Sigma}_{22}\boldsymbol{b}-1),$$

即Lagrange乘数法的解. 经过简单的求偏导运算解方程, 得到$\lambda=\nu=\boldsymbol{a}^\top\boldsymbol{\Sigma}_{12}\boldsymbol{b}=Cov(V,W)$及

$$\boldsymbol{A}\boldsymbol{a}=\lambda^2\boldsymbol{a},\ \boldsymbol{B}\boldsymbol{b}=\lambda^2\boldsymbol{b},$$

式中

$$\boldsymbol{A}\equiv\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21},\ \boldsymbol{B}\equiv\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}.$$

因此λ^2既为矩阵$\boldsymbol{A}$的特征值又为矩阵$\boldsymbol{B}$的特征值, 而相应的特征向量为$\boldsymbol{a}$和$\boldsymbol{b}$. $\boldsymbol{A}$和$\boldsymbol{B}$的特征值有如下性质:

(1) $\boldsymbol{A}$和$\boldsymbol{B}$有相同的非零特征值;
(2) 非零特征值的数目为$\min(p,q)$;
(3) $\boldsymbol{A}$和$\boldsymbol{B}$的特征值均在0和1之间.

不妨假定$p<q$ (则$p=\min(p,q)$), 这样, 那些称为典型相关系数的非零特征值和相应的特征向量为: $\lambda_1^2,\lambda_2^2,...,\lambda_p^2$ $(\lambda_1^2\geqslant\lambda_2^2\geqslant\cdots\geqslant\lambda_p^2>0)$; $\boldsymbol{a}_1,\boldsymbol{a}_2,...,\boldsymbol{a}_p$; $\boldsymbol{b}_1,\boldsymbol{b}_2,...,\boldsymbol{b}_p$. 由此可得到$p$对线性组合$V_i=\boldsymbol{a}_i^\top\boldsymbol{X},W_i=\boldsymbol{b}_i^\top\boldsymbol{Y}$, 称每一对变量为典型变量. 其相关系数极大值称为第一典型相关系数. 一般只取前几个影响大的典型变量和典型相关系数来分析. 典型变量的性质是:

(1) $\boldsymbol{X}$和$\boldsymbol{Y}$中的一切典型变量都不相关.

(2) $\boldsymbol{X}$和$\boldsymbol{Y}$的同一对典型变量V_i和W_i之间的相关系数为λ_i, 不同对的V_i和$W_j(i \neq j)$之间不相关.

当然, 在实际例子中一般并不知道$\boldsymbol{\Sigma}$, 因此在只有样本数据的情况下, 只要将$\boldsymbol{\Sigma}$用样本协差阵或样本相关阵$\boldsymbol{R}$代替即可, 但是这时的特征值可能不在0 ~ 1的范围内. 各种软件会给出调整后的相关系数. 我们来看下面的例子.

例 8.2 橄榄油数据(oliveoil.csv) 该数据来自Massart et al. (1998), 是针对来自希腊(第1 ~第5个)、意大利(第6 ~第10个)及西班牙(最后6个)的16个橄榄油样品得到的来自传感器的6个指标变量(sensory)及5个物理化学度量变量(chemical). 我们希望在sensory及chemical两组变量之间做典型相关分析. 在程序包pls中有这个数据. 全部数据展示在表8.4中.

表 8.4 例8.2 橄榄油数据

id	sensory						chemical				
	yellow	green	brown	glossy	transp	syrup	Acidity	Peroxide	K232	K270	DK
G1	21.4	73.4	10.1	79.7	75.2	50.3	0.73	12.7	1.900	0.139	0.003
G2	23.4	66.3	9.8	77.8	68.7	51.7	0.19	12.3	1.678	0.116	−0.004
G3	32.7	53.5	8.7	82.3	83.2	45.4	0.26	10.3	1.629	0.116	−0.005
G4	30.2	58.3	12.2	81.1	77.1	47.8	0.67	13.7	1.701	0.168	−0.002
G5	51.8	32.5	8.0	72.4	65.3	46.5	0.52	11.2	1.539	0.119	−0.001
I1	40.7	42.9	20.1	67.7	63.5	52.2	0.26	18.7	2.117	0.142	0.001
I2	53.8	30.4	11.5	77.8	77.3	45.2	0.24	15.3	1.891	0.116	0.000
I3	26.4	66.5	14.2	78.7	74.6	51.8	0.30	18.5	1.908	0.125	0.001
I4	65.7	12.1	10.3	81.6	79.6	48.3	0.35	15.6	1.824	0.104	0.000
I5	45.0	31.9	28.4	75.7	72.9	52.8	0.19	19.4	2.222	0.158	−0.003
S1	70.9	12.2	10.8	87.7	88.1	44.5	0.15	10.5	1.522	0.116	−0.004
S2	73.5	9.7	8.3	89.9	89.7	42.3	0.16	8.14	1.527	0.106	−0.002
S3	68.1	12.0	10.8	78.4	75.1	46.4	0.27	12.5	1.555	0.093	−0.002
S4	67.6	13.9	11.9	84.6	83.8	48.5	0.16	11.0	1.573	0.094	−0.003
S5	71.4	10.6	10.8	88.1	88.5	46.7	0.24	10.8	1.331	0.085	−0.003
S6	71.4	10.0	11.4	89.5	88.5	47.2	0.30	11.4	1.415	0.093	−0.004

下面利用程序包CCA[1]实行这两组变量的典型相关分析. 代码如下:

```
data(oliveoil,package="pls");attach(oliveoil)#数据
```

[1]Ignacio González and Sébastien Déjean (2009). CCA: Canonical correlation analysis. R package version 1.2. http://CRAN.R-project.org/package=CCA.

```
library(CCA)
(res.cc=cc(sensory,chemical))
```

由于两组变量中个数最少的一组有5个变量, 因此最多有5对典型相关变量, 输出中包括了5个典型相关系数: 0.9764811, 0.8397163, 0.8231290, 0.5730970, 0.2858563. 这5对典型相关变量为各自组中变量的线性组合, 其系数如下(注意: 输出中sensory组的典型相关变量称为X, 而chemical组的典型相关变量称为Y, 在其他软件中通常也是把一组称为"因变量", 另一组称为"自变量", 但实际上它们是对称的, 使用这种称谓仅仅是为了方便):

```
$xcoef
            [,1]    [,2]     [,3]     [,4]     [,5]
yellow -0.247101  0.3890  0.30391  0.10117  0.06152
green  -0.177212  0.3445  0.21346  0.06164  0.03562
brown   0.101121  0.2915 -0.06465  0.23142  0.06991
glossy  0.036967  0.1474 -0.46908  0.16579  0.60202
transp -0.005204 -0.1900  0.27997 -0.10092 -0.44396
syrup  -0.066060 -0.4347  0.43096 -0.17962  0.11612

$ycoef
               [,1]      [,2]     [,3]     [,4]     [,5]
Acidity     2.28933  -6.83138  -1.8312  -8.0146   1.5270
Peroxide    0.06877  -0.08713   0.2635   0.1413   0.5405
K232        4.31628  -6.11597  -1.6677  -5.1552  -8.8541
K270       -2.47051  83.26295 -16.0698  41.0047  25.1343
DK       -400.39944 636.94814 344.9703 152.6203 -64.4116
```

还可以从此得到各个观测值的得分(即**xscores**和**yscores**). 这些得分类似于因子得分, 为各个观测值做与典型变量同样的线性组合所得. 下面是输出中由典型相关变量得到的(sensory组的)x得分x-score和(chemical组的)y得分y-score分别与X组和Y组原始变量(即sensory和chemical)之间的相关系数(共有四种组合).

```
$scores$corr.X.xscores#即cor(sensory,res.cc$scores$xscores)
           [,1]     [,2]     [,3]    [,4]     [,5]
yellow -0.5523 -0.34911  0.02800  0.7169 -0.12568
green   0.4416  0.40651 -0.03411 -0.7439  0.13815
brown   0.8102  0.09532  0.35992  0.4415  0.09558
glossy -0.5475 -0.22618 -0.48003  0.4014  0.04302
```

```
transp -0.4898 -0.24276 -0.40015  0.4880 -0.18182
syrup   0.6588 -0.02562  0.50117 -0.2523  0.47694

$scores$corr.Y.xscores#即cor(chemical,res.cc$scores$xscores)
             [,1]    [,2]     [,3]      [,4]     [,5]
Acidity   0.02749 0.26653 -0.08334 -0.506353  0.09375
Peroxide  0.70226 0.09085  0.55018 -0.005638  0.04450
K232      0.77804 0.21840  0.38155 -0.084279 -0.07060
K270      0.72914 0.50102 -0.08460 -0.143035  0.03332
DK       -0.01254 0.37279  0.56679 -0.321762 -0.03308

$scores$corr.X.yscores#即cor(sensory,res.cc$scores$yscores)
          [,1]     [,2]     [,3]    [,4]     [,5]
yellow -0.5393 -0.29315  0.02305  0.4109 -0.03593
green   0.4312  0.34136 -0.02807 -0.4263  0.03949
brown   0.7912  0.08004  0.29626  0.2530  0.02732
glossy -0.5346 -0.18993 -0.39512  0.2301  0.01230
transp -0.4782 -0.20385 -0.32937  0.2797 -0.05197
syrup   0.6433 -0.02151  0.41253 -0.1446  0.13634

$scores$corr.Y.yscores#即cor(chemical,res.cc$scores$yscores)
             [,1]   [,2]    [,3]      [,4]    [,5]
Acidity   0.02815 0.3174 -0.1013 -0.883538  0.3280
Peroxide  0.71917 0.1082  0.6684 -0.009837  0.1557
K232      0.79678 0.2601  0.4635 -0.147060 -0.2470
K270      0.74671 0.5966 -0.1028 -0.249583  0.1166
DK       -0.01284 0.4439  0.6886 -0.561445 -0.1157
```

根据上面这些输出, 读者可以看出各个典型变量对各个原始变量的代表性以及各种变量之间的关系.

8.1.4 对应分析

对应分析是非常简单的数据描述方式, 在二维情况下特别直观, 但由于没有涉及分布及检验等经典统计要素, 因此有些多元统计的书不予介绍. 对应分析在概念上类似于主成分分析, 但应用于分类变量而不是连续变量, 传统上应用于列联表. 但由于其是描述性的, 只要意义清楚, 其他表格也可以应用. 所有

数据都应该是非负的, 并且行和列是平等的. 对应分析也有许多延伸, 比如应用于多维分类变量的情况(称为multiple correspondence analysis)等.

对应分析的数学不复杂. 先看下面来自程序包**MASS**(Venables and Ripley, 2002)的例子.

例 8.3 眼睛和头发颜色数据(caith.txt) 该数据是关于苏格兰Caithness地方人的眼睛颜色(变量eye)和头发颜色(变量hair)的列联表. 该地引人关注是因为那里混居着北欧日耳曼人(Nordic)、凯尔特人(Celtic)和盎格鲁–撒克逊人(Anglo-Saxon)的后裔. 数据见表8.5.

表 8.5 例8.3 眼睛和头发颜色数据

眼睛颜色(eye)	头发颜色(hair)				
	fair	red	medium	dark	black
blue	326	38	241	110	3
light	688	116	584	188	4
medium	343	84	909	412	26
dark	98	48	403	681	85

我们再看对应分析的结果(图8.10). 从图8.10可以看出, 颜色深的眼睛和颜色深的头发接近, 反之亦然. 这种直观易懂的图是由下面的代码作出的:

```
library(MASS);v=caith #数据来源
colnames(v)=paste(colnames(v),"hair")
rownames(v)=paste(rownames(v),"eye")
(cc=corresp(v,nf=2))#对应分析及结果输出
biplot(cc,xlim=c(-.55,.75))#画图
```

从原理上来说, 对应分析把一对列变量和一对行变量同时反映到同一张图上, 各自有其相应的坐标轴(因子轴). 下面从数学的角度解释对应分析. 对于二维列联表数据, 令$\boldsymbol{A}=\{a_{ij}\}$为$n\times m$矩阵, $\boldsymbol{x}=\{x_i\}$ 为n维向量, $\boldsymbol{y}=\{y_j\}$为m维向量. 那么满足下面条件的$(r,\boldsymbol{x},\boldsymbol{y})$称为对应分析问题$C_0(\boldsymbol{A})$的解:

$$rx_i=\sum_{j=1}^{m}\frac{a_{ij}y_j}{a_{i\cdot}},\quad i=1,2,...,n;$$

$$ry_j=\sum_{i=1}^{n}\frac{a_{ij}x_i}{a_{\cdot j}},\quad j=1,2,...,m.$$

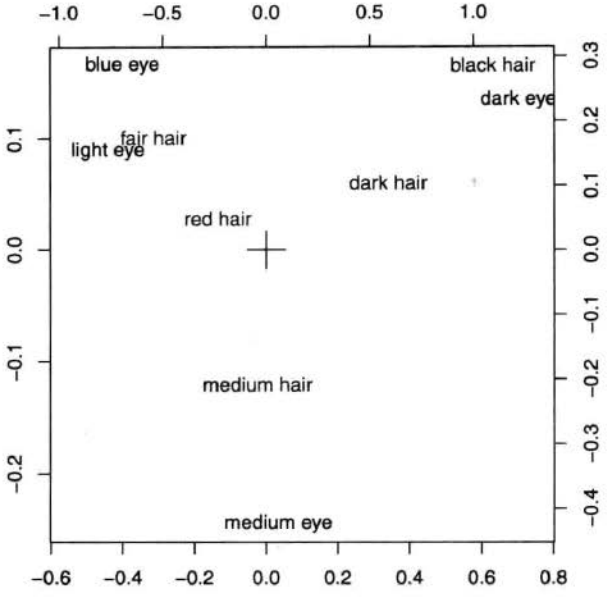


图 8.10 例8.3的对应分析结果

式中, $a_{i\cdot}=\sum_j a_{ij}, a_{\cdot j}=\sum_i a_{ij}$是各行及各列的元素和. 该式的意义为: 行记分(row score) x_i与列记分(column score) y_j的加权均值成比例, 列记分y_j则与行记分x_i的加权均值成比例. 数值r代表行列记分的相关(在典型相关的意义上). 对该问题解的数学推导也很简单, 对$\boldsymbol{A},\boldsymbol{x},\boldsymbol{y}$做某种变换之后得到新的变量, 相应地记为$\boldsymbol{Z},\boldsymbol{u},\boldsymbol{v}$, 上面两个方程可以写成等价的矩阵形式

$$r^2\boldsymbol{u}=\boldsymbol{Z}^{\top}\boldsymbol{Z}\boldsymbol{u};\quad r^2\boldsymbol{v}=\boldsymbol{Z}\boldsymbol{Z}^{\top}\boldsymbol{v}.$$

而从线性代数得知, 这是求两个有同样非零特征值的方程的特征值—特征向量问题. 每一个非零特征值对应于一个n维向量及一个m维向量, 通常对应分析把两个最大特征值所相应的两对主成分作为坐标, 得到两个因子载荷矩阵($m\times 2$和$n\times 2$). 把两个载荷图点在一起就完成了对应分析的演示. 显然r^2为对应于行和列的两个特征值问题的共同最大特征值的解(这里取最大的两个, 最多不超过$\min(m,n)$个). 这时, 如果用λ代表特征值, 则有

$$x_i=\frac{1}{\sqrt{\lambda}}\sum_{j=1}^{m}\frac{a_{ij}}{a_{i\cdot}}y_j,\quad i=1,2,...,n;$$

$$y_j = \frac{1}{\sqrt{\lambda}} \sum_{i=1}^{n} \frac{a_{ij}}{a_{\cdot j}} x_i, \quad j = 1, 2, ..., m$$

的形式, 这个形式是下一节公式的特殊情况.

下面是输出的典型相关系数、行记分和列记分:

```
First canonical correlation(s): 0.4463684 0.1734554

 Row scores:
                  [,1]       [,2]
blue eye    -0.89679252  0.9536227
light eye   -0.98731818  0.5100045
medium eye   0.07530627 -1.4124778
dark eye     1.57434710  0.7720361

 Column scores:
                   [,1]       [,2]
fair hair    -1.21871379  1.0022432
red hair     -0.52257500  0.2783364
medium hair  -0.09414671 -1.2009094
dark hair     1.31888486  0.5992920
black hair    2.45176017  1.6513565
```

8.2 非经典多元数据分析: 可视化

这一节介绍的多元分析方法包括主成分分析(PCA)、对应分析(CA)、多重对应分析(MCA)、多重因子分析(MFA)、分层多重因子分析(HMFA)、基于主成分分析的聚类(HCPC)等. 这些方法多数都可以很方便地处理各种定性和定量变量(PCA和HCPC只能处理定量变量, 但定性变量可以做补充变量)组成的数据, 无论数据是高维的还是分层分群的, 都可以用图形把它们的关系直观地展示出来, 这些特点都是经典多元分析所不具备的.

本节主要展示图形结果, 因为图形比其背后的数字更能够给人留下深刻印象. 所有图形所基于的数据都可以打印出来, 但由于数据复杂, 关系也复杂, 因此输出结果的数据量很大. 读者可以把需要的打印出来. 此外, 笔者尽量减少数学公式的堆砌, 以免读者被抽象的数学淹没.

在经典的多元分析中, 主成分分析、因子分析、典型相关分析及对应分析都是基于类似的降维思想, 用原始变量的线性组合作为新的变量, 而且在新变

量的选择上都用了代数的特征值—特征向量方法, 用特征值大小及相关性来体现所选择变量的重要性以及它们和原始变量或观测值的关系. 下面要介绍的"非经典"多元分析方法也利用了经典统计的思想, 但通过计算程序来改进结果, 主要强调的是几何直观演示. 因此, 所有方法都非常强调图形, 特别是二维图形的演示.

上一节介绍的对应分析的公式及演示实际上就是本节思想的最简单形式. 假定数据矩阵$\boldsymbol{X}$有I行K列, 那么$\boldsymbol{X}$的行(记为$\boldsymbol{N}_I$)就为在K维的R^K空间上的点群(称为云), 而$\boldsymbol{X}$的列(记为$\boldsymbol{N}_K$)就是在I维的R^I空间上的云. 假定$\boldsymbol{u}_s$代表$\boldsymbol{N}_I$云的第s大的主方向(即第s大的特征值对应的特征向量方向), 那么有(在标准化条件下)

$$\boldsymbol{X}^{\top}\boldsymbol{X}\boldsymbol{u}_s = \lambda_s \boldsymbol{u}_s, \quad \|\boldsymbol{u}_s\| = 1,$$

这里, λ_s就是$\boldsymbol{X}^{\top}\boldsymbol{X}$的第$s$大的特征值, 这也就是主成分分析的原理. 类似地, 假定$\boldsymbol{z}_s$代表$\boldsymbol{N}_K$云的第s大的主方向(即第s大的特征值对应的特征向量方向), 那么有(在标准化条件下)

$$\boldsymbol{X}\boldsymbol{X}^{\top}\boldsymbol{z}_s = \lambda_s \boldsymbol{z}_s, \quad \|\boldsymbol{z}_s\| = 1,$$

这里, λ_s就是$\boldsymbol{X}\boldsymbol{X}^{\top}$的第$s$大的特征值. 如果云$\boldsymbol{N}_I$及云$\boldsymbol{N}_K$投影到主方向, 则下面定义的$\boldsymbol{F}_s$和$\boldsymbol{G}_s$分别为云$\boldsymbol{N}_I$及云$\boldsymbol{N}_K$的第$s$个因子:

$$\boldsymbol{F}_s \equiv \boldsymbol{X}\boldsymbol{u}_s, \quad \boldsymbol{G}_s \equiv \boldsymbol{X}^{\top}\boldsymbol{z}_s.$$

$\boldsymbol{F}_s$在主成分分析中就称为主成分. 而云$\boldsymbol{N}_I$及云$\boldsymbol{N}_K$由下式关联起来:

$$\boldsymbol{z}_s = \frac{\boldsymbol{F}_s}{\|\boldsymbol{F}_s\|} = \frac{\boldsymbol{F}_s}{\sqrt{\lambda_s}}; \quad \boldsymbol{u}_s = \frac{\boldsymbol{G}_s}{\|\boldsymbol{G}_s\|} = \frac{\boldsymbol{G}_s}{\sqrt{\lambda_s}}.$$

可以马上推得

$$\boldsymbol{F}_s = \frac{1}{\sqrt{\lambda_s}}\boldsymbol{X}\boldsymbol{G}_s; \quad \boldsymbol{G}_s = \frac{1}{\sqrt{\lambda_s}}\boldsymbol{X}^{\top}\boldsymbol{F}_s.$$

如果不用矩阵形式, 考虑数据矩阵$\boldsymbol{X} = \{X_{ij}\}$, 基于第$s$方向的坐标轴的横坐标(代表个体$i$)为$F_s(i)$, 相应的纵坐标(代表变量$j$)为$G_s(j)$, 则有(在标准化条件下, 权重$m_j$及权重$p_i$为1)

$$F_s(i) = \frac{1}{\sqrt{\lambda_s}}\sum_j x_{ij} m_j G_s(j),$$

$$G_s(j) = \frac{1}{\sqrt{\lambda_s}}\sum_i x_{ij} p_i F_s(i),$$

这里, λ_s代表第s个特征值. 这个公式体现了广义因子分析的基本思想, 最初人们只认为其和对应分析有关, 但它也是主成分分析及其多重因子分析等各种新方法的基础.

本节介绍的新方法还引入了补充元素(变量或者个体), 它们不参加训练建模(没有参与估计权重m_j, p_i或λ_s), 但可以把补充的x_{ij}代入已经训练好的公式中, 以得到相应的坐标并点在图上, 就像机器学习中的测试集一样.

本节的一些内容请参看Lê et al. (2008), 更详细的内容请阅读他们2011年出版的书(Lê, et al. (2011)). 这里所用的程序包主要是**FactoMineR**. [1] 本节的主要内容是对各种图的展示以及对这些图的说明.

8.2.1 主成分分析

这部分与经典主成分分析的区别在于演示内容更加丰富, 并增加了补充元素作为测试集. 我们还是用例8.1的洛杉矶街区数据来做说明. 这里做主成分分析只用了六个变量(Income, Schools, Diversity, Age, Homes, Vets), 而把四个有关种族的变量(Asian, Black, Latino, White)拿来测试, 代码为:

```
w=read.csv("LA.Neighborhoods.csv")
u=w[,-c(1,12:15)]
library(FactoMineR)
#下面补充变量(测试)为7:10个:
res.pca <- PCA(u, quanti.sup = 7:10)
```

得到的图形如图8.11所示, 左图是各个街区(即个体或观测值)在两个主成分坐标系的位置, 右图则为各个变量在两个主成分坐标系的位置. 显然, 第一主成分较大的变量为比较富裕的(暂且称为"富裕主成分"), 而第二主成分较大的为学校成绩较好且种族多样性较高的. (读者看如何命名, 或者无法命名?) 在没有参加建模的种族变量中, 欧裔(White)在第一主成分中较大, 符合他们比较富裕的实际情况. 而拉美裔(Latino)在"富裕主成分"的负面, 也比较合乎实际. 非裔(Black)虽然也在"富裕主成分"的负面, 但箭头比较短, 说明在这个平面中(即在这两个成分中)被代表得不够, 越接近单位圆, 则被代表得越充分. 亚裔(Asian)主要被第二主成分代表, 但在这个平面中代表性不那么强(强于非裔). 左图仅仅给出街区的代码, 相信对这些街区的解释也可以参照对右图的理解. 从输出的`res.pca`中可以得到所有与这些图相关的数字, 但是, 这两个图总比大量的数字要令人感兴趣.

[1] Francois Husson, Julie Josse, Sebastien Le and Jeremy Mazet (2012). FactoMineR: Multivariate Exploratory Data Analysis and Data Mining with R. R package version 1.18. http://CRAN.R-project.org/package=FactoMineR.

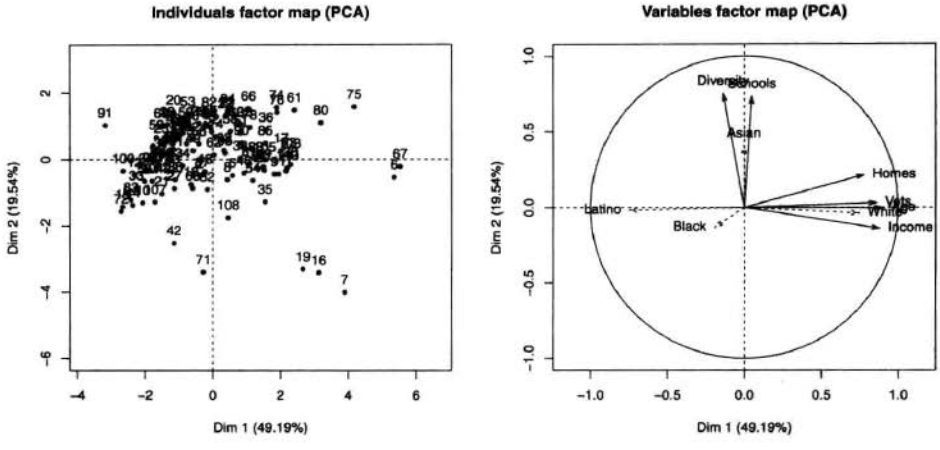


图 8.11　例8.1的主成分分析图

8.2.2　对应分析

这部分与经典对应分析的区别在于: 这里可以用补充元素作为测试集. 先来看一个例子.

例 8.4　父亲和儿子职业数据(glass.csv)　该数据是3497个英国家庭的父亲和儿子职业的列联表. 行和列分别为父亲和儿子职业种类, 为了识别, 行名字用大写, 列名字用小写. 代码意义为: PROF (专业人员和领导), EXEC (经理和执行人), HSUP (高级管理者), LSUP (低级管理者), SKIL (技术劳力和一般非劳力人员), SEMI (半技术劳力), UNSK (无技术劳工). 该数据来自Glass(1954), 在程序包anacor中可得到.

这里作了两个图(图8.12中的左图和右图), 一个用所有数据; 另一个把一个儿子的职业(skil)和一个父亲的职业(SEMI)排除于训练集之外但以测试集身份来测试结果. 看起来两个图的差别不大, 儿子的职业与父亲职业非常接近. 相关代码如下:

```
library(FactoMineR)
data(glass,package="anacor")
names(glass)=tolower(names(glass))
res.ca <- CA(glass)
res.ca <- CA(glass, col.sup = 5, row.sup = 6)
```

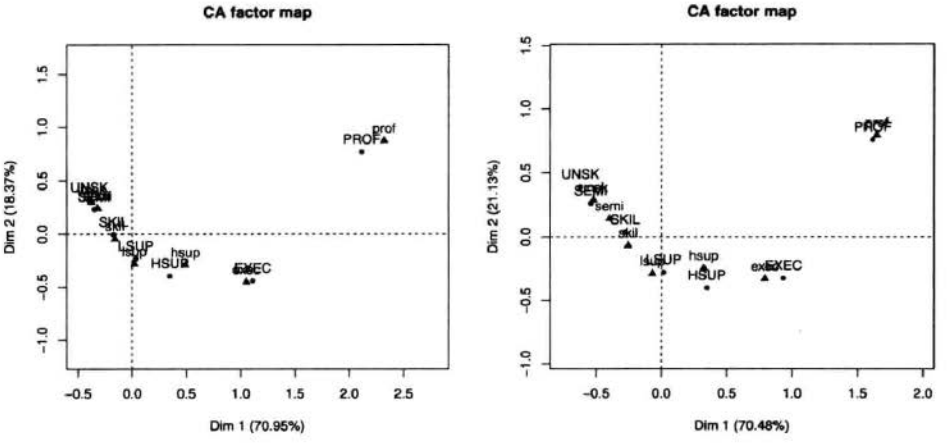


图 8.12 例8.4的对应分析图

如果数据有缺失值, 可以用程序包`missMDA`[1]中的函数`imputePCA()`来补上.

8.2.3 多重对应分析

这里的多重对应分析(multiple correspondent analysis)与8.2.2节内容的主要区别在于: 这里可以有很多变量, 包括数量变量和分类变量. 这是经典对应分析无法处理的. 先来看一个例子.

例 8.5 **汽车试验数据**(car.test.csv) 这是程序包`rpart`所带的汽车试验数据, 有60个观测, 8个变量: 价格(Price): 单位美元; 国家(Country): 产地名称; 可靠性(Reliability): 1 ~ 5分; 每加仑英里数(Mileage): 单位英里; 类型(Type):6种类型; 重量(Weight): 单位磅; 发动机容量(Disp): 单位升; 马力(HP). 该数据由John et al. (1992) 引自*Consumer Reports* (April, 1990, pp. 235 - 288).

这里用如下代码做多重对应分析, 得到的结果如图8.13所示.

```
data(car.test.frame,package="rpart")
library(FactoMineR)
#选择两个补充变量, 一个数量变量, 一个分类变量:
```

[1]Francois Husson and Julie Josse (2010). missMDA: Handling missing values with/in multivariate data analysis (principal component methods). R package version 1.2. http://CRAN.R-project.org/package=missMDA.

```
res.mca=MCA(car.test.frame,quanti.sup=7,quali.sup=2)
plot(res.mca,invisible=c("var","quali.sup",
    "quanti.sup"),cex=0.7)#选择不显示变量
plotellipses(res.mca,keepvar=c(2,5))#画置信椭圆
```

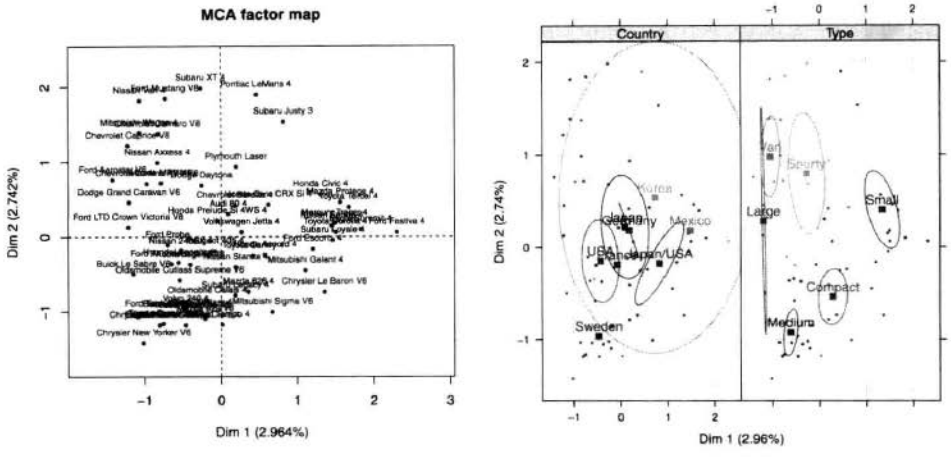


图 8.13　例8.5的多重对应分析图

图8.13中的左图为个体(各种汽车牌子和型号)在两个因子平面的点图, 可以选择点什么及不点什么, 如果不作选择, 则所有的内容都画在一张图上. 右图为两个分类变量上的其他变量的置信椭圆.

8.2.4　多重因子分析

多重因子分析(multiple factor analysis)意在处理分群结构的数据. 这些群都由不同类型的变量组成, 可以是数量变量, 也可以是分类变量. 它们同时都需要分析. 这种数据会出现在许多领域, 比如基因、传感、物理化学、代码的比较等. 对于结构的考虑使得每一群变量的影响得到平衡, 考虑了各组变量之间的联系, 既给出了传统的图形, 又标出了其特殊性. 这方面的细节可看一些参考资料, 如Jérôme Pagès的网上文章. 下面例8.6的数据属于分群数据. 对于不分群的数据, 如果既有分类变量, 又有数量变量, 可以简单地用函数AFDM()处理, 不必标明那些变量的类型.

多重因子分析对每个主轴都输出了其坐标、贡献及个体的方向余弦的平方, 在因子和连续变量之间的相关系数; 对每个分类变量, 输出了属于各水平(范畴)的个体的重心坐标和有关的检验值. 这里, 水平用重心代表, 对每个范畴能够计算相应的重心相对于总重要性的百分比, 称为贡献率. 变量k对轴s的

所有水平贡献率的和等于变量k和因子s的相关率. 这对所有分类变量都可以计算. 当然, 本书不会展示这些数字, 因为图形能够表明更多的含义. 前面所说的云$\boldsymbol{N}_I$和$\boldsymbol{N}_K$在这里可以分成很多子云, 多重因子分析的各种群都是子云. 我们来分析下面的例子.

例 8.6 **葡萄酒数据**(wine.csv) 这是程序包FactoMineR所带的数据[1], 有21个观测, 31个变量. 第一个变量为酒的(三种)来源标签, 第二个变量为土壤(四个水平), 其他变量都是酒的物理化学指标.

这个数据的变量是分组的, 一共有6组, 其中各组的变量数目分别为2, 5, 3, 10, 9, 2 (对应于下面代码的选项group=c(2,5,3,10,9,2)). 这6组中第1组为分类变量, 其余5组为数量变量, 对应于下面代码的选项type=c("n",rep("s",5)), 这6组的名字由代码name.group=c("orig","olf","vis","olfag","gust","ens")给出. 下面的代码给出若干图形(注意, 代码标明第1和第6组为补充群).

```
library(FactoMineR)
data(wine)
aa = MFA(wine,group=c(2,5,3,10,9,2),type=c("n",rep("s",5)),
    ncp=5,name.group=c("orig","olf","vis","olfag",
    "gust","ens"),num.group.sup=c(1,6),graph=FALSE)
#上面第1和第6组为补充群
par(mfrow=c(2,3))
plot(aa, choix = "ind") #图8.14上左
plot(aa, choix = "ind", partial="all")#图8.14上中
plot(aa, choix = "group")#图8.14上右
plot(aa, choix = "var", habillage="group")#图8.14下左
plot(aa, choix = "axes")#图8.14下中
plot(1:nrow(aa$eig),aa$eig[,1],type="o", pch=17,col=4,
    main="Eigenvalues", xlab="Dimension Number",
    ylab="Eigenvalues")#图8.14下右
```

图8.14的上左图及上中图是两个主因子的平面图, 其中, 上左图是个体及分类变量各水平在该平面的位置; 上中图是上左图的补充, 即个体及分类变量各水平(一共7个水平)位置以及对应于4个非补充群变量(分叉)的点图. 上右图为群变量点图(补充群特别标明); 下左图为各个变量的点图(包括了显示相关程度的单位圆), 但没有包括补充群的变量; 下中图为6个群变量在前5个因子的分量在第一、二因子平面中的地位图; 下右图为所有因子的地位排名(即相应的

[1] Centre de recherche INRA d'Angers.

特征值大小)图, 类似于崖底碎石图. 如要看清细节, 读者可按代码生成单独的图形.

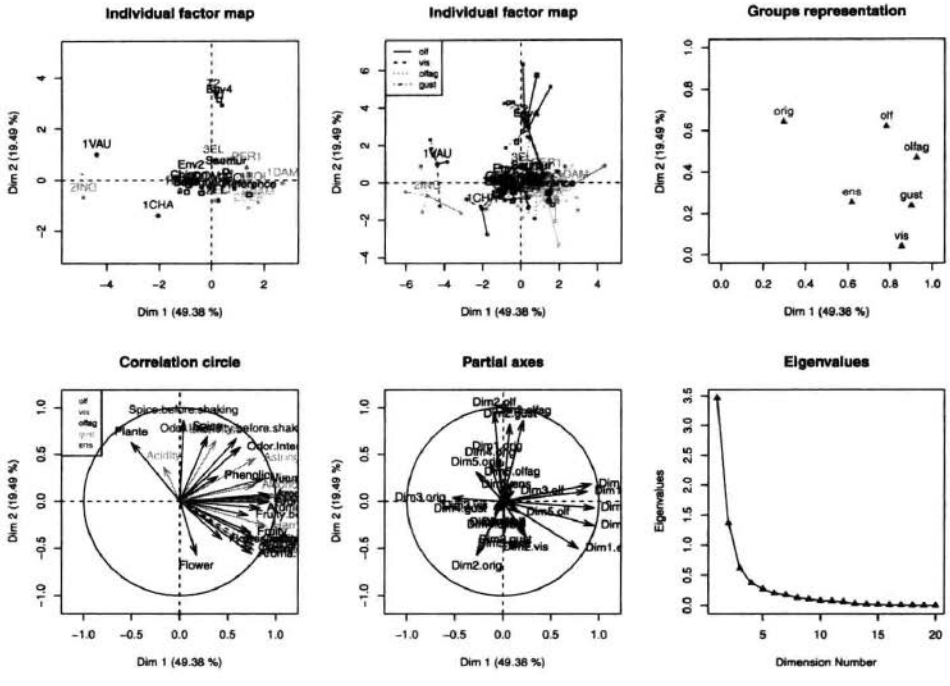


图 8.14　例8.6的多重因子分析

8.2.5　分层多重因子分析

分层多重因子分析(hierarchical multiple factor analysis)实际上就是在上面多重因子分析的各个群的基础上又分了群. 就拿前面的例8.6来说, 本来已经分了6组, 而这6组如果再分成两大组, 就成了分层(或多层)多重因子分析. 这里, 把这6组的前4组归为一群, 而后2组归为一群. 在输出的图中, L1代表第一层(底层), 一共有6组, 即L1:G1, L1:G2,...,L1:G6; L2代表第二层(高一层), 一共有两组, 即L2:G1, L2:G2. 计算代码为:

```
hierar <- list(c(2,5,3,10,9,2), c(4,2))
res.hmfa <- HMFA(wine, H = hierar, type=c("n",rep("s",5)))
```

这产生了一些图, 除了上面进行多重因子分析时已经有过的图之外, 主要有排列在图8.15中的两个图. 两图都标明了个体以及分类变量各水平的位置, 左图包括了它们与两层总共8组变量之间的关系, 右图仅包括它们与第二层的2组变量的直接关系.

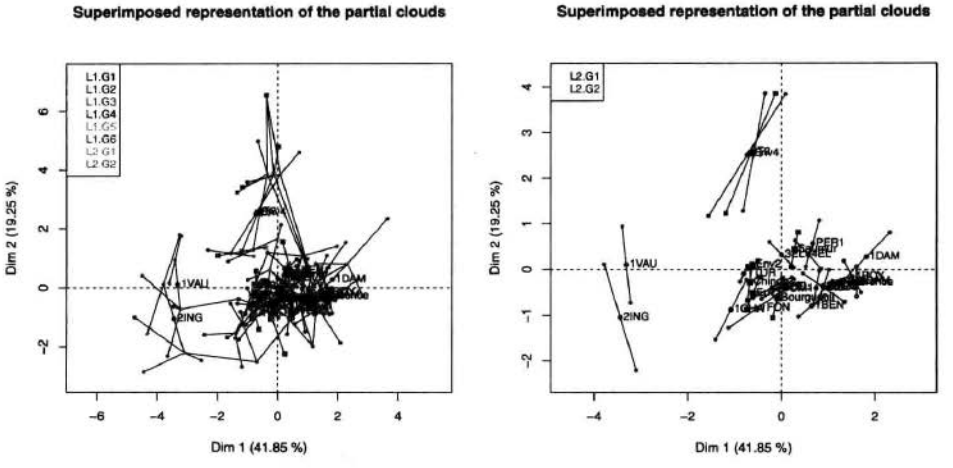


图 8.15 例8.6的分层多重因子分析

8.2.6 基于主成分分析的分层聚类

直接聚类的结果往往不太好, 原因是有噪声干扰, 但如果先进行主成分分析, 再进行聚类, 结果可能会好得多. 下面看一个例子.

例 8.7 生产数据(produc.csv) 这是美国48个州在1970—1986年长达17年的经济数据, 变量包括州名(state)、年份(year)、私人资本(pcap)、道路(hwy)、上下水系统(water)、其他公共设施(util)、公共资本(pc)、州生产总值(gsp)、非农就业(emp)、失业率(unemp). 该数据来源于Munnel (1990)及Baltagi and Pinnoi (1995), 为程序包Ecdat [1]的数据. 这里只用各州17年的平均值.

可用下面的代码先求各年的平均值, 创造新的数据集(z), 然后做主成分分析, 再在主成分分析的基础上做分层聚类, 最后画出图8.16和图8.17.

```
library(FactoMineR)
data(Produc,package="Ecdat")
aa=unique(Produc[,1]);z=NULL;
for(i in 1:48)
z=rbind(z,apply(Produc[Produc[,1]==aa[i],-(1:2)],2,mean))
row.names(z)=aa
res.pca=PCA(z,ncp=5,scale.unit=TRUE,graph = FALSE)#主成分分析
```

[1]Yves Croissant (2011). Ecdat: Data sets for econometrics. R package version 0.1-6.1. http://CRAN.R-project.org/package=Ecdat.

```
res.hcpc=HCPC(res.pca, nb.clust=-1,conso=0,min=3, max=10)
#上面nb.clust=-1意味着自动选聚类数目(0是手工选)
```

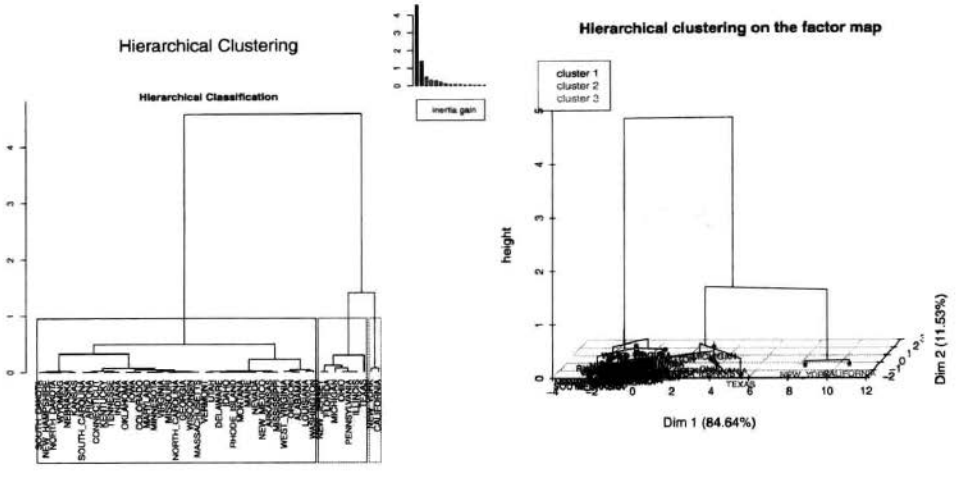


图 8.16 例8.7的分层聚类(1)

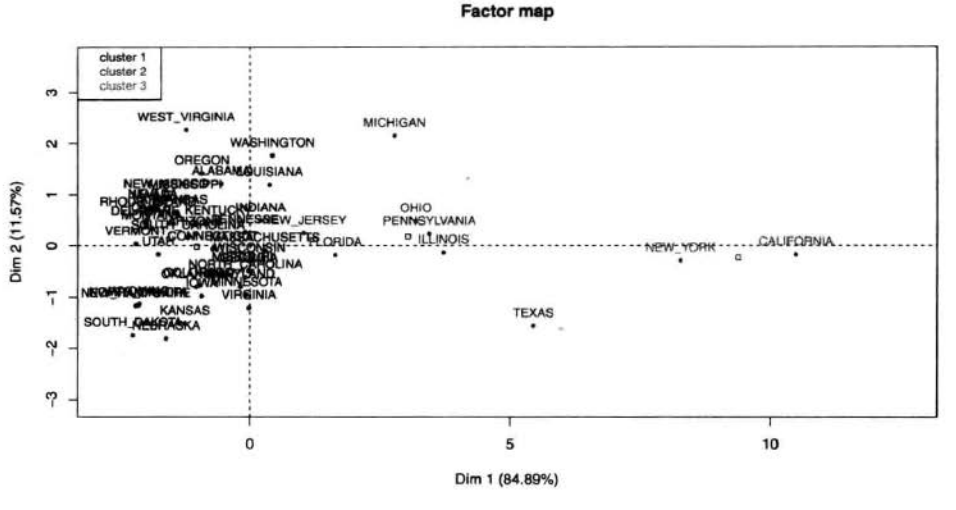


图 8.17 例8.7的分层聚类(2)

图8.16的左图是待选聚类数目的树状图(其右上角小图为特征向量的条形图), 右图是在两个主成分平面上的三维树状图. 图8.17为主成分平面上标明的自动分出的三类. 可以看出, 在主成分分析中, 已经把三类分得比较清楚了.

第九章　多元数据的关联规则分析

9.1　问题的提出

这一章很短, 放到多元分析一章似乎有些不合群, 于是单独列为一章.

关联规则分析(association analysis)也称关联规则挖掘(association rule mining), 它试图在一个很大的数据集中找出关联或相关的关系. 关联规则(association rules)显示出在一个给定的数据集中频繁地一同出现的变量. 关联规则挖掘由Agrawal, Imielinski & Swami(1993)最先提出.

最常用的例子是购物篮分析. 在超市, 收款台上的条码机通过扫描记录了每一个顾客所购买的全部商品, 这样, 商家就获得了什么商品是被最频繁购买的, 哪些商品被同时频繁购买等信息. 根据这些信息, 商家可用最优的方式安排货架, 确定哪些商品需要摆在一起, 哪些商品需要促销, 以及如何利用广告宣传商品.

来看啤酒和尿布的故事. 一个超市挖掘了所有顾客的交易, 商家吃惊地发现, 最经常和尿布一同售出的是啤酒; 商家还发现, 购买者主要是25~35岁、星期五下午购物的男性. 商家分析, 由于尿布体积大, 而平时购物的又是女性, 所以往往将购买尿布的责任交给当父亲的丈夫在周末完成. 而周末意味着喝啤酒, 因此, 啤酒就成为尿布的同时购买物. 于是超市把啤酒放在尿布旁, 结果由于方便, 新父亲买更多的啤酒, 就连平时不买啤酒的人也开始买了. 最终啤酒销量大增. 有人说这事发生在7-Eleven店, 有人说是在沃尔玛, 但这也有可能完全是口头文学. 无论是否有这回事, 数据挖掘得到的结果往往出人意料. 这就是科学探讨.

关联规则的原理很简单, 就是看一些事情同时出现的频率. 比如, 看买什么东西和买尿布同时出现的频率高. 原理虽然很简单, 但要在巨大的数据集中找出这些规律并和其他规律比较可不那么简单, 主要是对计算方法的挑战.

关联规则用途广泛, 除了用于交易数据、问卷调查等以分类变量为主的情况, 还可以利用把连续变量离散化来处理带有连续变量的情况. 先来看一个例子.

例 9.1　收入数据(IncomeESL.csv)　这是程序包**arules**[1]所带的数据, 来自Hastie et al. (2001), 有8993个观测值及14个变量. 该数据抽自美国普查数据, 变量为收入(income): 分为从$[0,10)$到$[75,\infty)$(千美元)的9个

[1]Michael Hahsler, Bettina Gruen and Kurt Hornik (2011). arules: Mining Association Rules and Frequent Itemsets. R package version 1.0-7.

区间; 性别(sex): 男(male)、女(female); 婚姻状态(marital status):已婚(married)、同居(cohabitation)、离异(divorced)、寡居(widowed)、单身(single); 年龄(age): 14 ~ 17, 18 ~ 24,..., 以后10年一段直到65+; 受教育程度(education): 9年级以下、9 ~ 11 年级、中学毕业(high school graduate)、大学肄业(college, 1 ~ 3 years)、大学毕业(college graduate)、研究生(graduate study); 职业(occupation):专业人员/经理(professional/managerial)、销售(sales)、劳工(laborer)、职员(clerical/service)、全职家务(homemaker)、学生(student)、军人(military)、退休(retired)、失业(unemployed); 居住在旧金山湾区的年限(years in bay area): 1, 1 ~ 3, 4 ~ 6, 7 ~ 10, > 10; 双份收入(dual incomes): 是(yes)、否(no)、未婚(not married); 家庭人口(number in household): 1, 2, ...,8, 9+; 孩子数量(number of children): 0, 1, 2, ...,8, 9+; 居住状况(householder status): 自有房产(own)、租房(rent)、和父母住或在别人家住(live with parents/family); 住房类型(type of home): 独栋房屋(house)、公寓(condominium)、单元房(apartment)、移动房屋(mobile home)、其他(other); 族裔(ethnic classification): 印第安人(american indian)、亚裔(asian)、非裔(black)、印度裔(east indian)、美洲西裔(hispanic)、太平洋岛民(pacific islander)、欧裔(white)、其他(other); 母语(language in home): 英语(english)、西班牙语(spanish)、其他(other).

9.2 数据的初步处理

这里用程序包arules来做关联规则分析, 首先对数据做整理, 下面的代码先把缺失数据删除, 剩下6876个观测值. 由于各个分类变量水平太多, 对它们做如下简化: 把收入从9个水平换成两个水平(0 ~ 40000美元及40000美元以上), 把年龄换成两个水平(14 ~ 35及35以上), 把受教育程度换成两个水平(大学没毕业及大学毕业), 把居住在旧金山湾区的年限换成1 ~ 9年和9年以上, 把家庭人口换成1个及多于1个, 把孩子个数换成没有孩子及有孩子. 注意, 关联规则处理的都是二分变量, 如果是连续变量, 则应该转换成离散的; 而如果不是二分变量, 则软件会把它们变成二分变量, 比如把有三个水平的变量, 自动转换成三个以水平为主的二分变量(每个都有"是"与"不是"两个水平).

```
library(arules)
data(IncomeESL)
#IncomeESL <- IncomeESL[complete.cases(IncomeESL), ]#删缺失值
IncomeESL <- na.omit(IncomeESL)#不要缺失值(等价于上一行)
identical(a,IncomeESL)
```

```
IncomeESL$income=factor(
    (as.numeric(IncomeESL$income)>6)+1,
    levels = 1 : 2 , labels = c("$0-$40,000", "$40,000+"))
IncomeESL$age=factor(
    (as.numeric(IncomeESL$age)> 3) +1,
    levels = 1 : 2 , labels = c("14-34", "35+"))
IncomeESL$education=factor(
    (as.numeric(IncomeESL$education)>4)+1,
    levels=1:2, labels=c("no college graduate",
    "college graduate"))
IncomeESL$"years in bay area"=factor(
    (as.numeric(IncomeESL$"years in bay area") > 4) +1,
    levels = 1 : 2 , labels = c("1-9", "10+"))
IncomeESL$"number in household"=factor(
    (as.numeric(IncomeESL$"number in household") > 3) +1,
    levels = 1 : 2 , labels = c("1", "2+"))
IncomeESL$"number of children"=factor(
    (as.numeric(IncomeESL$"number of children") > 1) +0,
    levels = 0 : 1 , labels = c("0", "1+"))
```

9.3 一些基本概念

由于程序包arules要求特别的数据形式("transactions"), 下面语句将其转换成"transactions"形式, 数据名为Income:

```
Income <- as(IncomeESL, "transactions")
summary(Income)
itemFrequencyPlot(Income,support=0.05,cex.names=0.8)#图9.1
```

图9.1显示了例9.1中出现最频繁的(频率在5%以上)项或项集, "项"和"项集"的定义包括在下面要给出的一些定义中.

在关联规则所使用的数据中, 每一个观测称为一个事务或交易(transaction), 每一个二分变量则称为一个项目或项(item). 一个数据为事务的集合, 称为事务数据集. 多个项目组成的集合称为项目集或项集(itemset). 用X表示一个项目或者项目集, Y表示与X没有交集的另一个项目或项目集, 那么记号$X => Y$ 表示X和Y同时出现的一个规则(rule). 在$X => Y$中, 称X为前项(也称为条件项(antecedent)或左项(left-hand-side, LHS of the rule)), 而称Y为后项(也称为结果项(consequent)或右项(right-hand-side, RHS of the

rule)). 而我们感兴趣的是涉及这个规则的有关频数或频率所导出或包含的信息. 这些信息包括：

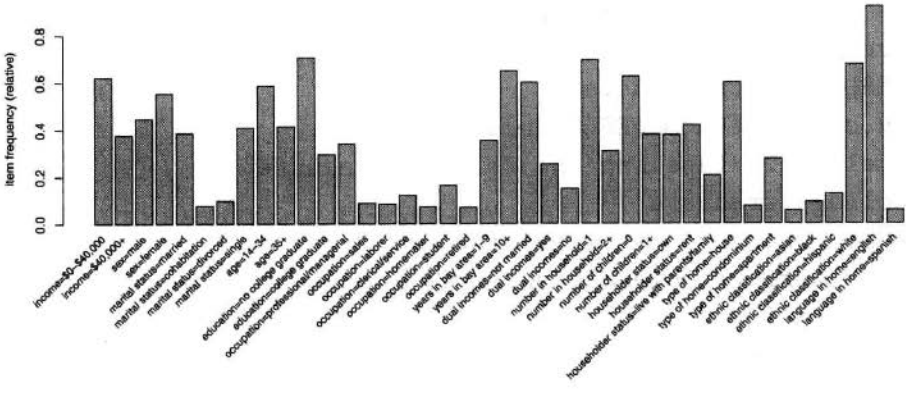


图 9.1 例9.1中出现最频繁的(频率在5%以上)项或项集

(1)$X => Y$的支持度(support)：定义为前项和后项在整个数据集中同时发生的频率. 假定整个事务数据集有N个事务, 用$\sigma(Z)$表示事务集Z的频数(Z中包含的事务个数), 用T_X表示包含项目X的事务集, 而T_Y表示包含项目Y的事务集, 则$X => Y$的支持度为:

$$\text{supp}(X => Y) \equiv \frac{\sigma(T_X \cap T_Y)}{N} = \frac{X\text{和}Y\text{同时出现的事务数目}}{\text{事务数总和}}.$$

显然, 支持度的前后项是对称的. 有的文献用频数而不是这里的比率(频率)来定义支持度.

(2) $X => Y$的置信度(confidence)(注意, 和统计中的置信度有区别)：定义为支持度和前项频率之比, 即

$$\text{conf}(X => Y) \equiv \frac{\text{supp}(X => Y)}{\text{supp}(X)} = \frac{\sigma(T_X \cap T_Y)}{\sigma(T_X)},$$

这里定义了$\text{supp}(X) \equiv \sigma(T_X)/N$. 置信度的前后项是不对称的.

(3) $X => Y$的提升(lift)：定义为置信度和后项频率之比, 即

$$\text{lift}(X => Y) \equiv \frac{\text{supp}(X => Y)}{\text{supp}(X)\text{supp}(Y)}.$$

提升的前后项是对称的.

9.4 频数和规则

用下面的语句求各个项或项集的频率并显示出来:

```
fsets=eclat(Income,parameter=list(support=0.05,maxlen=10))
inspect(sort(fsets, by = "support")[1:10])#只显示10项
```

输出如下, 显示了该数据集支持度最大的10个项(集)及其所占的比例:

```
   items                                  support
1  {language in home=english}             0.9128854
2  {education=no college graduate}        0.7052065
3  {number in household=1}                0.6918266
4  {ethnic classification=white}          0.6697208
5  {ethnic classification=white,
    language in home=english}             0.6595404
6  {number in household=1,
    language in home=english}             0.6495055
7  {years in bay area=10+}                0.6465969
8  {education=no college graduate,
    language in home=english}             0.6343805
9  {income=$0-$40,000}                    0.6224549
10 {number of children=0}                 0.6218732
```

从支持度的输出已经可以看出普查人口的大概构成. 下列语句中的第一条求各种规则(最主要的计算), 第二条则从把收入40000美元以上作为右项的规则中抽出提升大于1.2的那部分(各种规则的子集).

```
rules=apriori(Income,parameter=list(support=0.01,
    confidence=0.01))
x=subset(rules, subset = rhs %in%
    "income=$40,000+" &lift > 1.2)
```

然后可以用下面一行代码把规则按照支持度的降幂排列展示出来(头5个):

```
inspect(sort(x, by = "support")[1:5])
```

输出为:

```
  lhs                          rhs                    support conf lift
1 {type of home=house}      =>{income=$40000+} 0.28 0.46 1.23
2 {type of home=house,
```

```
   language in home=english}=>{income=$40000+} 0.26 0.48 1.27
3 {householder status=own}  =>{income=$40000+} 0.24 0.65 1.72
4 {marital status=married}  =>{income=$40000+} 0.24 0.61 1.63
5 {householder status=own,
   language in home=english}=>{income=$40000+} 0.23 0.66 1.74
```

这说明大多数收入超过4万美元的家庭都自己有房子, 说英文, 等等. 第一个规则说明: 住在独栋房屋(前项X)且收入在4万美元以上(后项Y)的人占总家庭数的27.7% ($\text{supp}(X => Y) = 0.277$), 而在住独栋房屋的人中, 收入超过4万美元的占46%($\text{conf}(X => Y) = 0.46$), 这个比例值为整个总体中相应比例值的1.23倍($\text{lift}(X => Y) = 1.231$). 下面的语句是按照提升(lift)的降幂排列展示(头3位)的:

```
inspect(sort(x, by = "lift")[1:3])
```

提升排前3位的输出为:

```
  lhs                                rhs      support conf lift
1 {education=college graduate,
   years in bay area=1-9,
   dual incomes=yes,
   number of children=0,
   householder status=own,
   language in home=english}=>{income=$40000+}0.01 0.98 2.59
2 {sex=male,
   age=35+,
   education=college graduate,
   occupation=professional/managerial,
   dual incomes=yes,
   number of children=0,
   householder status=own}  =>{income=$40000+}0.01 0.98 2.59
3 {marital status=married,
   education=college graduate,
   years in bay area=1-9,
   dual incomes=yes,
   number of children=0,
   householder status=own,
   language in home=english}=>{income=$40000+}0.01 0.98 2.59
```

从第一个规则可以看出, 如果同时满足下列条件(前项X): 大学毕业、居

住在旧金山湾区1年以上、有双份收入、没有孩子、拥有自己房子且说英语，而且同时(后项Y)收入在4万美元以上，则占总家庭数的1.3%(supp($X => Y$) = 0.013)，而收入在4万美元以上(Y)在上述人中(X)的百分比则高达97.8%(conf($X => Y$) = 0.978)，这个比例值为整个总体中的相应比例值的2.59倍(lift($X => Y$) = 2.59).

第十章　调查数据中垃圾比例的计算

10.1　动机

1.　吓人的巨型问卷

写这一章的动机来源于笔者遇到的许多令人汗颜的抽样调查实践. 比如, 有一个调查问卷比中国学生繁重的暑期作业厚得多, 有2400多个问题, 其中有1000多个数字问题, 有一页的问题是: 你有多少首饰、钟表、液化气灶具、缝纫机、自行车、电动自行车、摩托车、汽车、电风扇、空调、洗衣机、电脑、电冰箱、冰柜、照相机、收音机、录音机、收录机、组合音响、黑白电视机、彩色电视机; 每个用了多少年, 用什么方式得到的, 每个花了多少钱, 现在要卖值多少钱, 等等. 还有一页问题要求回答包括卫生纸、护肤化妆品、理发、美容、衣服鞋帽(分男女和儿童)、布料、家用工具(锤子、钳子、剪刀)、床上用品(被面、棉胎、毛毯、床垫、枕头、凉席、蚊帐)等物品过去一年中的花费. 在该问卷中还有约1400个是选答题, 选项总共有3700多个; 选答题的选项个数从(1900多问题的)2个到(56个问题的)14个.

对于这种问卷, 你能够回答吗? 完成这种问卷, 需要有极大的耐心、很好的记忆力、完美的修养以及充裕的时间. 但如此巨型的问卷只有几千个样本, 你能够相信这些问卷都会得到认真的回答吗? 更重要的是, 从这样的问卷中能够得到什么质量的结论呢? 这一类问卷很难用计算机进行分析(连录入数据都困难). 难道只计算一些简单的比例就可以把如此耗费资源的调查打发了吗?

2.　谁在生产这些问卷, 又产生了什么结论

有一项耗资巨大的问卷调查, 其领导小组有6人, 学术顾问3人, 项目办公室有8人, 学术委员会26人, 课题组12人, 调查报告主要撰稿人2人, 在这57个人中没有一个是学统计的, 仅在合作成员中有几个作为打工仔的统计师生.

根据这项问卷调查产生的200多页的调查报告(A4纸)展示了5800多个比例, 但是**这些比例没有任何一个有样本量、置信区间或置信度**. 不给出分母的比例是没有意义的. 难道一个人自己觉得应该当领导就可以说"100%的人支持我当领导"? (实际上却是分子分母均为1: $1/1 = 100\%$) 这不是笑话, 而是大多数比例所显示的赤裸裸的现实! 想想看, 这些比例中到底有多少垃圾?

这些调查问卷的设计者肯定不懂得如何处理调查数据, 他们往往为了自己的兴趣或者利益而随意增加问题的数量及每个问题的选项, 使得问卷极端不合理. 这些问卷调查在浪费了大量资源之后, 得到了质量低下且很难处理的数据.

3. 问卷调查数据处理中的弊端

由于某些人想突出政绩, 许多问卷调查数据在处理上受到很强的行政干预, 多数是要求"算出"权力持有者所需要的数字. 这使得诚实的数据处理人员苦不堪言.

此外, 大多数问卷调查都要求参加者签订保密协定, 这样, 对于任何过程中的猫腻, 没有人敢冒着破坏协定和得罪上司的风险来批评或曝光他们在调查中面对的问题. 这些调查耗费的是纳税者的血汗钱, 但却作为"政绩"或"科研成果"成为个人财产, 完全封闭在黑匣子中, 不但问卷数据不公开, 而且调查过程及数据处理过程完全对局外人保密. 有些甚至在调查之后连多余的问卷都要求收回或销毁, 以免被人抓住把柄.

10.2 我们的着眼点在于数据本身

本章不想谈论如何设计问卷这种在任何有关文献都能够找到的内容, 也无意讨论如何克服问卷调查弊端之类的既敏感又非学术性的问题. 这里仅仅谈论一个数据会产生多少垃圾这样的纯粹数据问题.

市面上有很多关于抽样调查的书籍, 但鲜有谈论抽样调查所产生的众多可能比例中有多少是没有价值的垃圾, 更少见关于如何计算这些垃圾比例的规模的讨论. 许多问卷的设计者为了得到自己感兴趣的比例, 在调查问卷中任意添加各种问题. 且不论大问卷使人厌倦而无法准确地做出回答, 本章主要想说明问卷中到底有多少没有意义的垃圾, 如何精确地计算出所有可能的比例及把数据中有用的和没用的比例识别出来.

事先我们需要考虑下面一些问题, 希望读者学完本章之后对这些问题会有一个较明确的理解.

- 是不是问卷的问题越多, 每个问题的选项越多, 得到的结果越多?
- 一个问卷的理论比例是多少?
- 一个问卷的各种有效比例中, 有多少有意义? 什么是有意义?
- 一个问卷中为什么会有那么多无意义的比例?

这里所说的有意义的比例是指一个比例的精确的95%置信区间必须足够窄. 所谓窄, 这里用的标准是0.06, 或者是$\pm$0.03, 即区间的半宽度是0.03. 根据下面的计算程序, 置信度不限于95%, 宽度也不限于0.06, 读者完全可以自己设定标准, 对于任何置信度的置信区间利用任何区间宽度对比例是否有意义(或者是不是垃圾)做出自己的判断.

本章将对简化的例9.1数据做分析, 看看该数据有多少比例有意义, 以及如何对此进行计算. 这个例子的所有问题都是单选题, 没有多选题, 而单选题比多

选题从逻辑和可答性来说要科学, 并且便于计算. 多选题数据当然也可以转换成多个单选题的数据来处理.

本章的程序全部由笔者编写, 错误及缺乏效率在所难免, 但愿能够抛砖引玉, 期待读者能够编出更好的程序.

10.3 解剖一个数据

例 10.1 收入数据(IncomeF.csv, Income2000.csv) 这两个数据是例9.1数据的两种简化版本, 其中IncomeF.csv是去掉缺失值的数据, 它和例9.1的数据有同样多的14个变量, 只不过去掉了缺失值, 因此只有6876个观测值. Income2000.csv是IncomeF.csv只保留5个变量(`income`, `education`, `occupation`, `type of home`, `ethnic classification`)以及随机抽取2000个观测值的数据. 数据各个变量的说明请参看例9.1.

我们首先看看数据IncomeF.csv的理论比例有多少个. 首先, 理论比例是以任何变量水平组合的交(集合$\mathcal{D}$)的观测值数目作为分母, 而以集合$\mathcal{D}$与其他任何变量的任何水平组合的交的观测值数目为分子. 为了解释这些概念, 假定一个数据有I个定性变量(记集合$\mathcal{I}=\{1,...,I\}$): $V_1,...,V_I$ (或$\{V_i\}_{i\in\mathcal{I}}$), 而变量$V_i$有$n_i$个水平, 这些水平记为$L_{ij}$ $(j=1,...,n_i)$. 记$\mathcal{O}$为所有观测值的集合. 定义$\mathcal{O}_{ij}$为$\mathcal{O}$中具有水平L_{ij}的观测值集合. 再用记号$\sigma(\mathcal{A})$(这里$\mathcal{A}$代表一个集合)表示一个集合$\mathcal{A}$中元素的个数. 以这样的记号, 任何一个比例的分母应该是下面类型观测值集合$\mathcal{D}$中观测值的总数$\sigma(\mathcal{D})$.

$$\mathcal{D}=\bigcap_{i\in\mathcal{I}_s;j\in\mathcal{J}_s}\mathcal{O}_{ij}$$

式中, $\mathcal{I}_s$表示下标集$\mathcal{I}$的某子集($\mathcal{I}_s\subset\mathcal{I}$), 这样的子集有$2^I-1$个; $\mathcal{J}_s$表示对每$n_i(i\in\mathcal{I}_s)$个下标取出一个j的j下标集合, 这样的集合的数目为:

$$\sigma(\mathcal{D})=\prod_{i\in\mathcal{I}_s}n_i.$$

而分子应该是任意类似的观测值集合$\mathcal{N}$与分母观测值集合$\mathcal{D}$的交(即与分子有关的观测值为与分母有关观测值集合的子集)$\mathcal{N}\cap\mathcal{D}$的观测值总数$\sigma(\mathcal{N}\cap\mathcal{D})$. 注意, 由于一个观测值只能取某变量的一个水平(由于调查问卷是单选题), 如果分子分母集合中出现了具有某个变量某水平的观测值, 那么, 具有该变量其他水平的观测值就不应该出现在比例中. 我们所有变量各水平的可能组合数目为:

$$\left[\prod_{i=1}^{I}(n_i+1)\right]-1. \tag{10.1}$$

式(10.1)中的(n_i+1)是计算了第i个变量没有被包括的可能, 比如, 第i个变量有3个水平, 每个观测值最多取一个水平, 因此算上没有这个变量的情况, 就有3+1=4种可能. 但还要减去观测值不属于任何变量的任何水平(不可能发生)的情况, 因此在式(10.1)中要减去1. 式(10.1)是针对所有观测值集合的, 类似的公式可用于任何有关的数据子集.

基于上面的思考, 下面给出计算理论比例的R函数:

```
TT=function(w){m=ncol(w)
Y=NULL;for(i in 1:m)Y=c(Y,length(levels(w[,i])))
Y1=Y+1;G=prod(Y1)-1
for(i in 1:m) G=G+Y[i]*(prod(Y1[-i])-1)#单独范畴为分母
for(k in 2:(m-1)){M=combn(1:m,k);J=ncol(M)#多范畴之交为分母
for(j in 1:J)G=G+prod(Y[M[,j]])*(prod(Y1[-M[,j]])-1)}
return(G)
}
w=read.csv("IncomeF.csv");n=nrow(w);m=ncol(w);dim(w)
TT(w)
```

数据IncomeF.csv的14个变量产生了一个$9\times2\times5\times7\times6\times9\times5\times3\times9\times10\times3\times5\times8\times3$列联表, 其格子数目为这些水平数的乘积$\prod_{i=1}^{I}n_i=16533720000$. 这些变量在理论上有各种变量各种水平的组合$\left[\prod_{i=1}^{I}(n_i+1)\right]-1=229920767999$个. 根据上面函数TT()计算出来一共有$1.089718\times10^{15}$个理论比例. 这的确是个天文数字. 当然, 由于样本量的限制, 这些比例绝大部分不存在, 也就是说, 上述列联表的格子中绝大多数都是0, 即绝大部分组合是空集. 因此变量再多, 或者问卷的问题数目再多, 也不见得能够得到想得到的比例或结果.

下面列举一些可能的比例, 并且"手算出"(也就是使用简单代码计算)在数据IncomeF.csv中的一些比例及它们的95%置信区间.

(1) **分母是观测值总数(问卷总数), 分子为所有可能的分母子集的观测值数.** 例子: 求收入7.5万美元以上的人(1-9: 第1个变量第9个水平)占所有人的比例是多少?

```
x1=table(w[,1])[9];x1/n;binom.test(x1,n)$con
```

得到样本比例为0.108057, 95%置信区间为(0.1008137, 0.1156346), 宽度为0.0148209. 这个置信区间宽度很窄.

(2) **分母是一个问题的某个水平的观测值数, 分子为所有可能的分母子集的观测值数.** 例子: 收入7.5万美元以上的人(1-9: 第1个变量第9个水平)在太平洋岛民后裔(13-6: 第13个变量第6个水平)中的比例是多少?

```
x1=table(w[,c(1,13)])[9,6];n1=table(w[,c(13)])[6]
x1/n1;binom.test(x1,n1)$con
```

得到样本比例为0.1022727, 95%置信区间为(0.04784291, 0.18528608), 宽度为0.1374432. 这个置信区间宽度有些太宽了.

(3) **分母是两个问题的水平的交集, 分子为所有分母子集的观测值数**. 例子: 太平洋岛民后裔(13-6)年龄大于65岁的人(4-7)中收入在7.5万美元以上(1-9)的比例是多少?

```
x1=table(w[,c(1,13,4)])[9,6,7];n1=table(w[,c(13,4)])[6,7]
x1/n1;binom.test(x1,n1)$con
```

得到样本比例为0, 95%置信区间为(0.000, 0.975), 宽度为0.975. 这个置信区间宽度接近1, 没有什么意义了.

(4) **分母是三个问题的水平的交集, 分子为所有可能的分母子集的观测值数**. 例子: 住公寓(12-2)的欧裔(13-7)大于65岁(4-7)的人群中收入在7.5万美元(1-9)以上的比例是多少?

```
x1=table(w[,c(12,13,1,4)])[2,7,9,7]
n1=table(w[,c(12,13,4)])[2,7,7];(p=x1/n1)
binom.test(x1,n1)$con
```

得到样本比例为0.09756098, 95%置信区间为(0.02722747, 0.23131455), 宽度为0.2040871. 这个置信区间宽度很宽了.

从这几个比例可以看出, 当比例的分母为全部观测值时(这里用n代表), 置信区间宽度很窄. 但当分母的数目变小时(由于只取部分有某些特征的数据), 置信区间会变大, 往往大大超过通常民调所习惯的宽度0.06(或±3%).

很多教科书谈论样本量该取多少时, 都以宽度0.06的置信区间有$100(1-\alpha)$%置信度的标准来确定样本量n的大小. 由于比例的$100(1-\alpha)$%置信区间近似为:

$$\hat{p} \pm t_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}},$$

所以, 半宽度为0.03的计算样本量n的公式为:

$$n \approx \frac{\hat{p}(1-\hat{p})}{(0.03/t_{\alpha/2})^2}.$$

对于95%置信区间, $\alpha/2 = 0.025$; 而为了使得风险最小, 公式中选择$\hat{p} = 0.5$(抽样之前不可能知道$\hat{p}$), 因此比较保险的样本量(整数)n约为:

$$n \approx \frac{0.5 \times 0.5}{(0.03/t_{0.025})^2} \approx 1067.072 \approx 1068.$$

这也是大多数调查公司的样本量都不少于这个数的原因. 注意, 这个样本量仅

仅是对以全部观测值数目n为分母的比例而估算的. 以一些变量水平的组合而得到的观测值数目作为分母来计算比例时, 必须确保那些分母成分大, 才会使得比例有意义(95%置信区间足够短, 比如小于0.06).

为了简化计算, 我们对例10.1的数据IncomeF.csv仅保留了5个变量, 并随机抽样得到2000个观测值, 这就是数据Income2000.csv. 其实不用下载数据, 可以直接从程序包arules得到(下面的w就和Income2000.csv一样):

```
data(IncomeESL,package="arules")
w=na.omit(IncomeESL);N=nrow(w);dim(w)
set.seed(1010);
w=w[sample(1:N,2000),c(1,5,6,12,13)];n=nrow(w);m=ncol(w)
```

通过前面TT()函数的计算, 可以得到数据Income2000.csv的理论比例为839791个. 这个数字当然和样本量无关, 只和变量有关, 真正得到的比例个数和种类与具体数据中的观测值有关.

10.4 计算例10.1Income2000.csv数据中各种比例及置信区间

10.4.1 一些计算准备工作

为了做后面的计算, 先做些准备工作. 首先, 把一些要素计算出来(利用前面生成的Income2000.csv数据, 或者读入数据, 名为w):

```
w=read.csv("Income2000.csv");n=nrow(w);m=ncol(w)
#下面的Y是各个变量的水平数:
Y=NULL;for(i in 1:m)Y=c(Y,length(levels(w[,i])))
Y1=Y+1 #Y+1是包括了不包含某水平的可能
#下面是分母和分子组合的所有可能数目:
prod(Y1)-1#=37799
```

为了编程方便, 把各个变量的水平数字化:

```
#用数字表示各个水平(为了编程方便):
for(i in 1:m)levels(w[,i])=1:Y[i]
#把每个变量水平加一个水平表示空水平(即没有该水平)存在P中
P=list()
for(i in 1:m)P[[i]]=as.character(c(levels(w[,i]),Y[i]+1))
#下面为各种变量水平的组合, 有prod(Y1)-1=37799行
pp=expand.grid(P);pp=pp[-nrow(pp),]#减去最后一行空集组合
```

```
for(i in 1:ncol(pp))pp[,i]=as.integer(pp[,i])#pp变成整数
```

上面的pp矩阵是寻求各种可能比例的分母和分子水平组合的基础, 它由目前数据的各种现存水平组合组成, 列数等于涉及的变量个数, 行数等于该数据存在的各种组合的数目. 这种矩阵也可以对任何数据子集计算, 这是我们计算程序的关键. 在后面的程序中, 符号PP1会在各种情况下重复出现, 并且根据具体问题而变化, 代表pp不同情况下的子集(最大等于pp).

10.4.2 主要函数

下面是计算各种比例的一个主要函数, 输出是一个数据框(R的一种对象格式), 其每一行代表一个比例, 一共四个列变量: n1为分母大小(和输入的一样), x1为分子大小, p为样本比例(p=x1/n1), len为95%精确置信区间的长度. 当然, 通过修改程序, 读者还可以增加或改变输出内容. 相信读者可以写出比这个函数更加有效的代码. 该函数的输入包括: PP1为pp中去掉了分母中包含的那些变量列的pp子集; Y8为Y中去掉了分母中包含的那些变量列相应的水平数的结果; w1为原始数据w中只包含分母水平组合的观测值集合, 且删除了分母的所有变量; n1为分母大小(和输出的一样). 这个函数是为了在已经确定了分母的情况下计算所有比例而编写的.

```
WW=function(PP1=PP1,Y8=Y8,w1=w1,n1=n1){
RR1=NULL;for(i in 1:nrow(PP1))
{
a=TRUE;for(j in 1:ncol(PP1))
{
 ind=PP1[i,j]
if(length(dim(Y8))==1)B=(ind!=Y8+1)else
    B=(ind!=Y8[j]+1)
if(prod(dim(w1))==1){if(B) a=(a&(w1==levels(w1)[ind]))} else
    if(B)a=(a&(w1[,j]==levels(w1[,j])[ind]))
 #上面a为PP1第i行所有非空列组合的TRUE-FALSE向量(和w1行数同)
    }
if (is.numeric(PP1))A=(sum(PP1[i]==Y8+1)!=length(Y8))else
    A=(sum(PP1[i,]==Y8+1)!=length(Y8))
if (A){
    p=sum(a)/n1;len=diff(binom.test(sum(a),n1,con=.95)$con)
RR1=rbind(RR1,c(n1,sum(a),p,len))}
}
```

```
RR1=data.frame(RR1);names(RR1)=c("n","x","p","len")
return(RR1)
}
```

为了熟悉如何使用函数WW(), 这里举一个简单的例子. 比如我们想要同时具有w的第3个变量(occupation)的第2个水平(homemaker)、第4个变量(type of home)的第3个水平(apartment)以及第2个变量(education)的第5个水平(college graduate)的观测值数目作为分子[1], 把这些变量和相应的水平号码放入一个两行矩阵x:

```
> x=c(3,4,2,2,3,5);x=matrix(x,ncol=length(x)/2,b=T);x
     [,1] [,2] [,3]
[1,]    3    4    2
[2,]    2    3    5
```

x的第一行为变量号码, 第二行为相应于第一行的某水平号码. 下面的a0则为满足分母条件观测值的由TRUE和FALSE组成的n维向量, 而w1为w数据中仅满足分母条件的观测值的子集, 而且删除分母所有的变量. 对于这个x, w1只有第1、第5个变量(income和ethnic classification)剩下了, w1的行数为n1, 这也是分母(这里只有9个观测值满足分母条件). 要计算的分子则从这两个变量水平的各种组合与分母的交中产生.

```
a0=TRUE;
for(i in 1:ncol(x))
a0=a0&(w[,x[1,i]]==levels(w[,x[1,i]])[x[2,i]])
w1=w[a0,-x[1,]]#保留a0的行, 去掉分母变量的列
n1=sum(a0)#w1的行数, 即比例的分母
```

然后, 把pp中相应的分母列去掉, 把重复的行去掉, 形成PP1; Y8则是把Y中相应于分母的列的元素去掉. 这样就可以算出前面x所定义的分母的全部比例及95%置信区间宽度(存在RT中).

```
PP1=unique(pp[,-x[1,]])
Y8=Y[-x[1,]]
RT=WW(PP1,Y8,w1,n1)
```

得到的RT为一个89 × 4的数据框, 也就是说, 可以算出的比例为89个, 其中只有13个不为0. 相应于这13个非零比例的RT为:

```
> RT[RT$p!=0,]
```

[1]注意这里变量水平的排名是按照原先水平字母顺序排列的.

```
    n  x          p       len
9   12  1 0.08333333 0.3826886
10  12  1 0.08333333 0.3826886
74  12  1 0.08333333 0.3826886
76  12  1 0.08333333 0.3826886
77  12  1 0.08333333 0.3826886
78  12  4 0.33333333 0.5518784
79  12  4 0.33333333 0.5518784
80  12 11 0.91666667 0.3826886
84  12  1 0.08333333 0.3826886
86  12  1 0.08333333 0.3826886
87  12  1 0.08333333 0.3826886
88  12  4 0.33333333 0.5518784
89  12  5 0.41666667 0.5716781
```

显然, 这些比例全部是垃圾比例, 它们的95%置信区间的长度最少也是0.3826886.

10.4.3 更多的计算

1. 分母为全部观测值的所有可能比例

先计算以总样本量n为分母的所有可能比例. 这时, 由于分母为全部数据的样本量, 因此在这里PP1=pp, Y8=Y, w1=w, n1=n.

```
R0=WW(pp,Y,w,n)#分母为n, 全部比例
dim(R0)#37799     4
summary(R0)
sum(R0$x==0)#比例中0的数目: 31453
sum(R0$x==0)/nrow(R0)#0.832112
sum(R0$x>0)#6346
sum(R0$x>0)/nrow(R0)#0.167888
```

其中, summary(R0)的输出表明, 95%置信区间最大的长度也只有0.043608. 因此, 所有的37799个比例都很"理想". 但再仔细观看, 比例中0的数目为31453个, 占总比例的83.2%, 这说明绝大多数分子的计数为0. 而只有6346个比例不为零, 占16.8%. 因此想要得到那些得不到的比例, 必须增大样本量(在抽样调查实践中, 谈何容易).

2. 分母为一个变量一个水平的所有可能比例

这里循环中的x是一个两行一列矩阵, 第一行是所有可能的一个变量号码, 第二行是这个变量的所有可能的水平号码之一. 把不属于这个范畴的观测值及相应的变量列从原数据剔除, 得到w1, 而把这个变量之外的所有其他可能变量的各种水平组合放入PP1中. n1为w1的行数(分母), Y8为剩下变量的各个水平数目.

```
R=NULL;for(i in 1:m)for(j in 1:Y[i]){
x=matrix(c(i,j),nrow=2,b=T)
a0=TRUE;for(k in 1:ncol(x))a0=a0&(w[,x[1,k]]==
    levels(w[,x[1,k]])[x[2,k]])
#保留a0的行, 去掉分母变量的列(剩下的列为setdiff(1:7,x[1,]))
w1=w[a0,-x[1,]];n1=sum(a0)
PP1=unique(pp[,-x[1,]])#PP1把相应的分母列去掉, 把重复的行去掉
Y8=Y[-x[1,]]#PP1的各列水平数(注意空列=(每一列水平)+1)
R=rbind(R,WW(PP1,Y8,w1,n1))#分母数为一个范畴的全部比例
}
```

下面程序代码汇总了计算得到的结果:

```
dim(R)#165503       4
t0=table(R$len<0.06, R$p>0)
dimnames(t0)[[1]]=c("len>0.06","len<0.06")
dimnames(t0)[[2]]=c("p=0","p>0")
t0
(t00=prop.table(t0))
(t01=prop.table(t0,1))
(t02=prop.table(t0,2))
```

输出为置信区间长度是否大于0.06及比例是否大于0的列联表:

```
> t0

             p=0    p>0
  len>0.06  34167   4288
  len<0.06 108234  18814
```

上述列联表的各个计数的比例:

```
> (t00=prop.table(t0))
```

```
                p=0       p>0
len>0.06 0.2064434 0.0259089
len<0.06 0.6539700 0.1136777
```

列联表按照行的比例(边缘分布之一):

```
> (t01=prop.table(t0,1))

                p=0       p>0
len>0.06 0.8884930 0.1115070
len<0.06 0.8519142 0.1480858
```

列联表按照列的比例(边缘分布之二):

```
> (t02=prop.table(t0,2))

                p=0       p>0
len>0.06 0.2399351 0.1856116
len<0.06 0.7600649 0.8143884
```

对于例10.1的Income2000.csv数据, 在一个变量某一个水平为分母时, `range(R$len)`代码显示, 95%置信区间的长度从0.00273到0.97500不等. 根据代码`apply(t00,1,sum)`的结果, 95%置信区间宽度小于0.06的占76.76%, 说明只有约23.24%(根据`apply(t0,1,sum)`有38455 个)的比例的置信区间大于0.06. 但值得注意的是, 那些"令人满意"的比例中, 有142401个为0(根据`apply(t0,2,sum))`, 根据代码`apply(t00,2,sum)`的结果, 等于0的比例占所有比例(165503个)的86.04%! 只有13.96%(23102个)的比例不等于0. 在比例不等于0的95%置信区间中, 宽度小于0.06的有18814个(上面输出的表`t0`), 占比例总数的11.37%(上面输出的表`t00`), 占大于0的比例的81.44%(上面输出的表`t02`).

总之, 这些比例中的垃圾所占的比例(`1-t00[2,2]`)为0.8863223! 注意, 这还仅仅是一个变量的一个范畴(水平)作为分母的情况.

3. 分母为至少两个变量的两个范畴组合的全部比例

下面计算分母为至少两个变量的两个范畴组合的全部比例. 这里的`list`类的对象`CC`一共有`m`个元素, 每个元素相应于`m`个变量(我们的数据`m=5`)的水平号码(可以用代码`CC`显示). 这里的`X`是相对于`m`个变量选取`I`个的所有组合, 比如, 在`I=3`时, 有(共有$\binom{5}{3}$[1]$= 10$种选择):

[1]组合数符号$\binom{n}{k}$的等价符号有C_n^k, $C_{k,n}$等.

```
> I=3;X=combn(1:m,I);X
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    1    1    1    1    1    2    2    2     3
[2,]    2    2    2    3    3    4    3    3    4     4
[3,]    3    4    5    4    5    5    4    5    5     5
```

而对于下面的每个j, 则选中了上面X的第j种组合. 例如, 当j=2时, CC[X[,j]]为第1, 2, 4个变量的各个水平号码(它们的水平数目分别为9, 6, 5), 而下面代码产生的MM为这三个变量各种水平组成的矩阵, 每一行为一种组合. 因此MM一共有$9 \times 6 \times 5 = 270$行.

```
j=2;MM=expand.grid(CC[X[,j]])
```

于是X的每一列及相应的MM的每一行就组成一个前面提到的两行矩阵x, 并据此(及前面的pp等)算出WW()函数需要的PP1, Y8, w1, n1.

下面是计算分母为至少两个变量的两个范畴组合的全部比例的全部代码.

```
#收集w的水平信息(每个变量水平的名字)
CC=list();for(i in 1:m)CC[[i]]=as.character(levels(w[,i]))
UU=NULL;
for(I in 2:(m-1)){
X=combn(1:m,I);
for(j in 1:ncol(X)){
MM=expand.grid(CC[X[,j]])
for(k in 1:nrow(MM)){
x=matrix(c(X[,j],as.integer(MM[k,])),nrow=2,b=T)
PP1=unique(pp[,-x[1,]])
#PP1把相应的分母列去掉, 把重复的行去掉
PP1=as.matrix(PP1)
Y8=Y[-x[1,]]#PP1的各列水平数(空列=(每一列水平)+1)
a0=TRUE
for(i in 1:ncol(x))
a0=a0&(w[,x[1,i]]==levels(w[,x[1,i]])[x[2,i]])
n1=sum(a0)#a0是w中等于x[2,]的行号(TRUE)
if(n1!=0){
w1=w[a0,-x[1,]]#;w1=as.matrix(w1)
#w1保留a0的行, 如果n1=sum(a0)=0则无w1
UU=rbind(UU,WW0(PP1,Y8,w1,n1))} else
{UU=rbind(UU,c(n1,n1,n1,1))}
```

```
}}}
dim(UU)#[1]  436305       4
```

在查看计算结果时, 要把那些空集(即程序中分母n1为零的)除掉, 然后形成列联表及列联表的分布, 分母为0的比例一共有12994个, 下面代码中的U是除去这些比例的UU. U包含剩下的423311个比例.

```
sum(UU$n==0)#12994
U=UU[UU$n>0,];dim(U)#423311       4
t1=table(U$len<0.06, U$p>0)
dimnames(t1)[[1]]=c("len>0.06","len<0.06")
dimnames(t1)[[2]]=c("p=0","p>0")
t1
(t10=prop.table(t1))
(t12=prop.table(t1,2))
```

输出为置信区间长度是否大于0.06及比例是否大于0的列联表:

```
> t1

             p=0    p>0
  len>0.06 320155  51687
  len<0.06  43087   8382
```

输出上述列联表的各个计数的比例:

```
> (t10=prop.table(t1))

                  p=0         p>0
  len>0.06 0.75631155 0.12210172
  len<0.06 0.10178568 0.01980104
```

输出列联表按照行的比例(边缘分布之一):

```
> (t11=prop.table(t1,1))

                 p=0       p>0
  len>0.06 0.8609974 0.1390026
  len<0.06 0.8371447 0.1628553
```

输出列联表按照列的比例(边缘分布之二):

```
> (t12=prop.table(t1,2))
```

```
                p=0       p>0
len>0.06 0.8813821 0.8604605
len<0.06 0.1186179 0.1395395
```

这一系列输出的结果说明, 在各(至少两个)变量各个水平(范畴)的组合为分母时, 95%置信区间宽度小于0.06的占12.16%(代码apply(t10,1,sum)), 只有51469个(代码apply(t1,1,sum)). 但值得注意的是, 这些"令人满意"的比例中, 有43087个为0(上面输出的表t1), 样本比例不等于0而且长度小于0.06的95%置信区间只有8382个(上面输出的表t1), 占总比例数(423311)的1.98%(上面输出的表t10). 在大于0的比例中有86.05%的置信区间宽度大于0.06(上面输出的表t12), 实际上在大于0的比例中有32.87%的置信区间宽度大于0.5(根据代码sum(U$len>0.5&U$p>0)/sum(U$p>0)).

总之, 这些比例中的垃圾占所有比例的98.02%! (代码1-t10[2,2]) 注意, 这个数据仅仅针对包含5个问题的问卷, 对于那些有几千个问题的巨型问卷, 垃圾之多可想而知.

程序上的注: 读者可能注意到了, 在计算MM的时候, 我们用的是各个变量的各个真实水平, 而在计算pp的时候, 我们用的是各个变量的各个真实水平加上一个空水平. 这是因为基于pp的PP1是在分母固定时所有可能的分子形式, 无论是几个范畴的交集(没有按照所有变量个数来循环), 因此, 每一行的空水平元素代表相应于空元素的变量水平不在分子中. 而MM是针对分母的, 计算时是按照分母变量个数及水平种类来循环的, 不需要空水平描述. 当然, 分母也可以按照pp的产生模式来产生, 但那样程序就要大改了. 这一章的程序改动了许多次, 都是为了一个目的, 但各个版本如此不同, 连笔者都记不得每一次的思路. 这就是编程序的特点及乐趣(或麻烦)所在.

10.5 通过一个简单人造数值例子解释本章代码的意义

例 10.2 信用例子 (cred100.csv) 为了逐步解释本章代码, 我们引进这个例子. 该例一共有100个观测值, 4个定性变量: age, edu, income, credit, 分别有2, 3, 3, 2个水平. 通过R代码, 得到变量具体情况为:

```
w=read.csv("cred100.csv");(n=nrow(w));(m=ncol(w))
summary(w)
    age       edu         income    credit
 Old  :47   Col:14   High  :47   Exc :22
 Young:53   El :53   Low   : 7   Fair:78
            HS :33   Middle:46
```

利用下面代码, 得到总的各种变量水平的理论组合数为143个(包括有些变量没有参加的情况), 而所有变量都在场的水平组合数只有$36 = (2 \times 3 \times 3 \times 2)$个. 这些数字和数据无关, 仅仅和变量及其水平数目有关.

```
Y=NULL;
for(i in 1:m)Y=c(Y,length(levels(w[,i])))
Y1=Y+1 #分母和分子组合:
prod(Y1)-1#=143
prod(Y)#=36
```

通过下面代码, 得到该数据的总理论比例数目为1081个, 这也和实际数据无关.

```
TT=function(w){m=ncol(w)
Y=NULL;for(i in 1:m)Y=c(Y,length(levels(w[,i])))
Y1=Y+1;G=prod(Y1)-1
for(i in 1:m) G=G+Y[i]*(prod(Y1[-i])-1)#单独范畴为分母
for(k in 2:(m-1)){M=combn(1:m,k);J=ncol(M)
for(j in 1:J)G=G+prod(Y[M[,j]])*(prod(Y1[-M[,j]])-1)}
return(G)
}
TT(w)
```

为了编程序方便, 用数字表示各个水平:

```
for(i in 1:m)levels(w[,i])=1:Y[i]
summary(w)
 age     edu     income credit
 1:47    1:14    1:47   1:22
 2:53    2:53    2: 7   2:78
         3:33    3:46
```

```
#把每个变量水平加一个水平表示空水平(即没有该水平)存在P中
#比如水平1,2,3->1,2,3,4
P=list();for(i in 1:m)
P[[i]]=as.character(c(levels(w[,i]),Y[i]+1))
#下面为各种变量水平的组合, 有prod(Y1)-1=37799行
pp=expand.grid(P);pp=pp[-nrow(pp),]
for(i in 1:ncol(pp))pp[,i]=as.integer(pp[,i])#pp变成整数
```

对于例10.2的人造数据, pp有143行(用dim(pp)可见), 每行代表4个变量

的某种组合(除了全部不在场的情况之外). 由于\verbpp+是自动生成的, 它没有用age, edu, income, credit的名字, 而是用Var1, Var2, Var3, Var4. 名字不要紧, 程序中又不用它们. 头几行和最后几行为(这里的4个变量的水平数目本来分别是2, 3, 3, 2, 但这里出现的是3, 4, 4, 3. 多出来的数目标志着该变量不参加):

```
> head(pp)
  Var1 Var2 Var3 Var4
1    1    1    1    1
2    2    1    1    1
3    3    1    1    1
4    1    2    1    1
5    2    2    1    1
6    3    2    1    1
> tail(pp)
    Var1 Var2 Var3 Var4
138    3    2    4    3
139    1    3    4    3
140    2    3    4    3
141    3    3    4    3
142    1    4    4    3
143    2    4    4    3
```

我们现在把年轻高收入者人数作为分母, 这是变量age(Var1)为Young(水平2)及income(Var3)为High(水平1)的组合. 于是用x来表示:

```
x=matrix(c(1,3,2,1),ncol=2,b=T)
> x
     [,1] [,2]
[1,]    1    3
[2,]    2    1
```

其中第一行代表分母涉及的变量号码(这里是第1和第3), 第二行为分母中相应变量的水平(第2和第1水平). 下面的a0为年轻高收入者在数据中的行号, 为由100个TRUE-FALSE构成的向量(TRUE表示该行是这样的组合, FALSE则表示不是):

```
a0=TRUE;for(i in 1:ncol(x))
a0=a0&(w[,x[1,i]]==levels(w[,x[1,i]])[x[2,i]])
sum(a0)#有23个TRUE, 即满足这样的数据只有23个观测值
```

a0标志的观测值组成的数据为:

```
summary(w[a0,])
 age     edu     income credit
 1: 0    1: 4    1:23   1: 2
 2:23    2:11    2: 0   2:21
         3: 8    3: 0
```

显然, 该数据没有老人和非高收入的了(年龄和收入都只有一个水平存在了). 我们去掉变量1和变量3, 剩下变量2(edu)和变量4(credit)作为在此分母下可能的分子组合:

```
#w1为保留a0的行, 去掉分母的列(剩下的列号为setdiff(1:m,x[1,]))
> w1=w[a0,-x[1,]]
> n1=sum(a0)#23
> summary(w1)
 edu     credit
 1: 4    1: 2
 2:11    2:21
 3: 8
> w1
    edu credit
1     2      2
2     2      1
3     2      2
4     3      1
5     2      2
16    1      2
19    2      2
34    3      2
35    3      2
46    3      2
52    1      2
53    3      2
55    3      2
56    2      2
59    3      2
70    1      2
```

```
72    1       2
75    2       2
76    3       2
79    2       2
90    2       2
97    2       2
100   2       2
```

下面的PP1把相应的两个分母列去掉, 把重复的行去掉.

```
PP1=unique(pp[,-x[1,]])
dim(PP1)#12 2 #4*3=12
PP1
Y8=Y[-x[1,]]#Y8=3 2
```

PP1就是第2和第4个变量的各种组合(包括某一个不参加的情况, 即Var2等于4, Var4等于3):

```
> PP1
    Var2 Var4
1      1    1
4      2    1
7      3    1
10     4    1
49     1    2
52     2    2
55     3    2
58     4    2
97     1    3
100    2    3
103    3    3
106    4    3
```

通过主要函数WW()(这里对WW()为这个数据稍微修改了一下, 使它输出了分子变量(Var2和Var4)的水平号. 注意: 水平号为0代表相应的变量没有参加分子)可得

```
> RT=WW(PP1,Y8,w1,n1)
> RT
    n  x Var2 Var4          p       len
1  23  0    1    1 0.00000000 0.1481851
```

```
2  23  1   2   1 0.04347826 0.2183864
3  23  1   3   1 0.04347826 0.2183864
4  23  2   0   1 0.08695652 0.2696694
5  23  4   1   2 0.17391304 0.3383042
6  23 10   2   2 0.43478261 0.4231392
7  23  7   3   2 0.30434783 0.3970888
8  23 21   0   2 0.91304348 0.2696694
9  23  4   1   0 0.17391304 0.3383042
10 23 11   2   0 0.47826087 0.4259258
11 23  8   3   0 0.34782609 0.4088924
```

这个可以和w1逐个验证.

第十一章　路径建模(结构方程建模)数据的PLS分析

11.1　路径模型概述

路径建模(path modeling)也称结构方程建模(structural equation modeling, SEM). 与回归分析及因子分析相比, 路径建模是一个年轻的方向, 从20世纪60年代后期开始出现在论文中, 其方法还在发展, 基本的概念正面临挑战和修正. 路径建模是一种验证性的方法, 其基本思想是根据想象、已有假说或已有经验事先构造一个模型, 先画路径图并写出关系, 然后根据这个模型收集数据, 再用该模型来拟合. 由于没有涉及其他模型, 因此无法通过比较来确定该模型是否比未知模型更合适, 只能在构造的模型框架内部加以调整. 因此, 不能期望用路径建模来发现一个更适合的模型(它不是一种探索性方法). 但是, 这种方法对于比较成熟的模型(如瑞典、西欧和美国的顾客满意度模型)还是很有效的.

11.1.1　路径模型

图11.1是一个欧洲顾客满意度(European customer satisfaction index, ECSI)的路径模型. 其中的左图为结构模型(structural model), 也称内部模型或内部关系, 结构模型中(在椭圆中)的变量都是不可观测或度量的变量, 称为隐变量或潜变量(latent variable, LV). 这个模型中的隐变量包括: 企业形象(corporate image, IMAG)、顾客期望(customer expectations, EXPE)、感知质量(perceived product quality, QUAL)、感知价值(perceived value, VAL)、顾客满意度(customer satisfaction index, SAT)、顾客忠诚度(customer loyalty, LOY). 图中的箭头意味着因果关系. 其中没有被箭头所指的隐变量只有IMAG, 称为外生变量(exogenous variables), 它是模型中其他变量的原因变量(其他变量不是它的原因). 而其他5个隐变量都有箭头所指, 称为内生变量(endogenous variables). 对内生变量来说, 模型中有一个或更多的隐变量为其原因, 内生变量也可能是其他隐变量的原因.[1] 右图为该路径模型的测量模型(measurement model), 也称外部模型或外部关系, 其中矩形中的变量称为显变量(manifest variable)或可观测变量, 也称隐变量的指标(indicator). 测量模型把(椭圆中的)隐变量和(矩形中的)显变量联系起来. 在这个模型中显变量分别为与那6个隐变量有关的imag1, imag2, imag3, imag4, imag5, expe1, expe2, expe3, expe4, expe5, qual1, qual2, qual3, qual4, qual5, val1, val2,

[1] 内生变量和外生变量的概念不仅仅限于这里的隐变量.

val3, val4, sat1, sat2, sat3, sat4, loy1, loy2, loy3, loy4等27个变量. 图11.1中的左右两图结合起来就是完整的路径模型或结构方程模型.

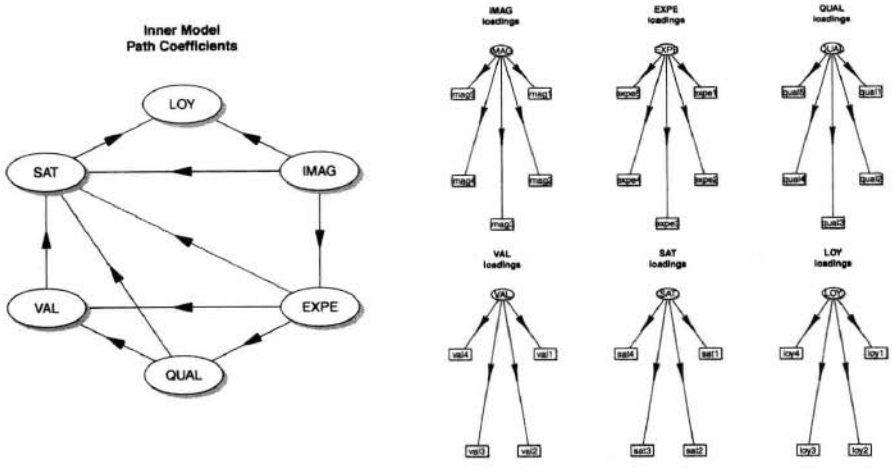


图 11.1　路径模型

11.1.2　路径模型的两种主要方法

路径模型基本上有两种分析方法: 偏最小二乘法(partial least squares, PLS)及协方差方法, 后者又称最大似然法(ML)、LISREL(软件名字)方法、AMOS(软件名字)方法或者ML-LISREL方法. 中国许多冠以"结构方程模型"之名的教材和文章都以后者为主要内容, 因此人们往往认为ML-LISREL方法等价于结构方程模型, 这是一种误解. 这两种方法除了分析目标都是路径模型之外, 没有任何内容上的交叉, 很难比较它们的结果. 它们的模型假定、计算过程及输出均有很大区别.

偏最小二乘法(PLS)完全根据原始数据通过迭代算法找到各个变量之间完全的线性关系, 而且估计出全部隐变量的值. ML-LISREL方法则在对数据的多元正态性的假定之下, 加上图模型所确定的协方差关系(而PLS方法总是假定变量之间相关, 图上及公式中无须注明), 得到似然函数, 然后以最大似然法来得到各种估计, 但最大似然法无法估计出隐变量的值, 也无法得到线性关系的绝对系数(只有相对系数). ML-LISREL方法可以仅用均值、协方差矩阵及样本量来得到所有结果, 这与多元分析中的因子分析类似. 实际上, 从某种意义来说, 因子分析为结构方程模型的一个特例, 各个因子就是隐变量.

下面列举它们的一些区别:

- PLS方法不用对数据做任何分布假定, 而ML-LISREL方法必须假定数据服从多元正态分布.
- PLS方法假定所有隐变量都是相关的(即使在图模型中它们之间没有箭头), 而ML-LISREL方法, 只要隐变量之间没有箭头, 就假定它们之间的相关严格为零, 并体现在后续的计算之中.
- PLS方法用全部数据建模, 而ML-LISREL方法由于假定了分布, 只要有各变量的均值、协方差矩阵和样本量就可以计算.[1]
- 如果假定了数据变量的正态性, 则ML-LISREL方法可以输出多达40多种不同的统计量、检验的p值等, 而PLS方法无法做这些检验, PLS有一些其他指标.
- PLS适用于关注隐变量得分的情况, 比如满意度指数, 而ML-LISREL方法无法直接得到隐变量得分, 因此各国在计算满意度指数时都用PLS方法.
- PLS适用于小样本情形.
- PLS由于收敛速度快, 因此适用于较大、较复杂的结构方程模型, 计算效率比ML-LISREL更高.
- 由于长期没有商业软件支持, PLS的知名度不如ML-LISREL方法高, 而后者由于有LISREL, AMOS(SPSS公司产品), EQS, COSAN, EZ-PATH等软件的支持和商业宣传, 知名度较高. 但现在有多个R程序包可以使用, 商业软件也开始开发PLS方法.
- 由于数学上的复杂性使其有数学文章可做, ML-LISREL方法有很多数学研究论文, 而PLS主要是欧洲大陆(所谓非"盎格鲁–撒克逊"统计传统[2])的人在研究.
- 由于软件成熟, 而且只要有协方差阵就可以计算, 一些社会学、心理学、教育学等方面的学者喜欢用ML-LISREL方法, 但现在随着PLS相关软件的普及, 情况也有所改变.
- 关于ML-LISREL方法在正态假定不满足时的稳健性, 许多论文都予以否认, 这些结论往往与LISREL, AMOS软件商的宣传大相径庭.
- 在人文学科的实际应用中, 往往把统计不显著的结果都判定为显著. 这实际上不是软件的问题, 而是使用者对数据的假定太强, 路径模型的设定太主观, 不能体现数据产生的背景规律, 才导致出现如此削足适履的做法.
- ML-LISREL的支持者总是用PLS方法的有偏性来做文章. 在正态假定

[1]这意味着, 假定有4个可观测变量, 则只要有均值、协方差下三角矩阵(10个数)及样本量等总共12个数字就可以得到填满几页纸的大量输出. 这意味着先验的假定对结果的强大影响.

[2]前面介绍的非经典多元分析、偏最小二乘回归、对应分析以及这里介绍的PLS方法等都是欧洲大陆学者开发的, 与欧洲大陆人所说的"盎格鲁–撒克逊"统计传统(即英美统计传统)有所区别.

下, PLS估计是有偏的, 但这个偏差造成的误差远远小于由不正确假定造成的误差.

- ML-LISREL方法是理论导向的, 强调从探索到确认性分析的转换, PLS主要是在高度复杂但又没有什么理论信息时做因果预测分析.
- 两种方法不应该是互斥的, 而应该是互补的, 除了软件商之外, 学者之间的争论应该保持在学术层面.

11.2 PLS方法: 顾客满意度的例子

本节就以欧洲顾客满意度为例, 说明PLS方法.

例 11.1 欧洲顾客满意度(satisfaction.csv) 这是程序包plspm[1]自带的数据, 来自Laboratory of Information Analysis and Modeling (LIAM). Facultat d'Informatica de Barcelona, Universitat Politecnica de Catalunya, 有250个观测值及28个变量. 该数据的变量在前面介绍路径模型图11.1时已经介绍过, 都是显变量, 隐变量因为不可观测, 只出现在模型之中. 除了那27个显变量之外, 还有1个没有被包括在下面分析中的性别(gender)变量. 图11.1完全描述了该路径模型.

除了图11.1之外, 该模型还有等价的方程表示, 假定这27个显变量用$x_1, x_2, ..., x_{27}$表示, 外生隐变量用ξ表示(代表IMAG), 内生隐变量用$\eta_1, \eta_2, ..., \eta_5$表示(依次代表EXPE, QUAL, VAL, SAT, LOY), 则结构模型(图11.1中的左图)等价于带有误差项的方程

$$\begin{pmatrix} \xi \\ \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \\ \eta_5 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \beta_{21} & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta_{32} & 0 & 0 & 0 & 0 \\ 0 & \beta_{42} & \beta_{43} & 0 & 0 & 0 \\ \beta_{51} & \beta_{52} & \beta_{53} & \beta_{54} & 0 & 0 \\ \beta_{61} & 0 & 0 & 0 & \beta_{65} & 0 \end{pmatrix} \begin{pmatrix} \xi \\ \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \\ \eta_5 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \xi + \begin{pmatrix} 0 \\ \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \\ \zeta_5 \end{pmatrix}.$$

该方程也可以写成

$$\begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \\ \eta_5 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \beta_{32} & 0 & 0 & 0 & 0 \\ \beta_{42} & \beta_{43} & 0 & 0 & 0 \\ \beta_{52} & \beta_{53} & \beta_{54} & 0 & 0 \\ 0 & 0 & 0 & \beta_{65} & 0 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \\ \eta_5 \end{pmatrix} + \begin{pmatrix} \beta_{21} \\ 0 \\ 0 \\ \beta_{51} \\ \beta_{61} \end{pmatrix} \xi + \begin{pmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \\ \zeta_5 \end{pmatrix}.$$

[1]Gaston Sanchez (2012). plspm: Partial Least Squares Data Analysis Methods. R package version 0.2-1. http://CRAN.R-project.org/package=plspm.

一些文献将内生变量和外生变量之间的系数用单独一套字母(比如$\{\gamma_{ij}\}$)表示, 其矩阵形式为$\boldsymbol{\Gamma}$, 这样上式的矩阵形式为:

$$\boldsymbol{\eta} = \boldsymbol{B\eta} + \boldsymbol{\Gamma\xi} + \boldsymbol{\zeta}.$$

而测量模型(图11.1中的右图)等价于下面带有误差项的方程. 其中涉及外生变量的为:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} \lambda_{11} \\ \lambda_{12} \\ \lambda_{13} \\ \lambda_{14} \\ \lambda_{15} \end{pmatrix} \xi + \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \end{pmatrix},$$

其矩阵形式为:

$$\boldsymbol{x}_\xi = \boldsymbol{\Lambda}_\xi \boldsymbol{\xi} + \boldsymbol{\delta}_\xi,$$

而涉及内生变量的为:

$$\begin{pmatrix} x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \\ x_{17} \\ x_{18} \\ x_{19} \\ x_{20} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{25} \\ x_{26} \\ x_{27} \end{pmatrix} = \begin{pmatrix} \lambda_{21} & 0 & 0 & 0 & 0 \\ \lambda_{22} & 0 & 0 & 0 & 0 \\ \lambda_{23} & 0 & 0 & 0 & 0 \\ \lambda_{24} & 0 & 0 & 0 & 0 \\ \lambda_{25} & 0 & 0 & 0 & 0 \\ 0 & \lambda_{31} & 0 & 0 & 0 \\ 0 & \lambda_{32} & 0 & 0 & 0 \\ 0 & \lambda_{33} & 0 & 0 & 0 \\ 0 & \lambda_{34} & 0 & 0 & 0 \\ 0 & \lambda_{35} & 0 & 0 & 0 \\ 0 & 0 & \lambda_{41} & 0 & 0 \\ 0 & 0 & \lambda_{42} & 0 & 0 \\ 0 & 0 & \lambda_{43} & 0 & 0 \\ 0 & 0 & \lambda_{44} & 0 & 0 \\ 0 & 0 & 0 & \lambda_{51} & 0 \\ 0 & 0 & 0 & \lambda_{52} & 0 \\ 0 & 0 & 0 & \lambda_{53} & 0 \\ 0 & 0 & 0 & \lambda_{54} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{61} \\ 0 & 0 & 0 & 0 & \lambda_{62} \\ 0 & 0 & 0 & 0 & \lambda_{63} \\ 0 & 0 & 0 & 0 & \lambda_{64} \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \\ \eta_5 \end{pmatrix} + \begin{pmatrix} \delta_6 \\ \delta_7 \\ \delta_8 \\ \delta_9 \\ \delta_{10} \\ \delta_{11} \\ \delta_{12} \\ \delta_{13} \\ \delta_{14} \\ \delta_{15} \\ \delta_{16} \\ \delta_{17} \\ \delta_{18} \\ \delta_{19} \\ \delta_{20} \\ \delta_{21} \\ \delta_{22} \\ \delta_{23} \\ \delta_{24} \\ \delta_{25} \\ \delta_{26} \\ \delta_{27} \end{pmatrix},$$

其矩阵形式为:

$$\boldsymbol{x}_\eta = \boldsymbol{\Lambda}_\eta \boldsymbol{\eta} + \boldsymbol{\delta}_\eta.$$

偏最小二乘的计算步骤为:

(1) 用迭代(包括一系列OLS回归、线性运算及抽取平方根)来得到隐变量$\boldsymbol{\xi}$和$\boldsymbol{\eta}$的表示式(通常把$\boldsymbol{\eta}$也写成$\boldsymbol{\xi}$的分量, 放到一起, 用$\boldsymbol{\xi}$表示)

$$\hat{\xi}_j = \sum \tilde{w}_{jh} x_{jh},$$

它们是显变量$\boldsymbol{x}$的线性组合(通过加权$\boldsymbol{w}$), 实际上对每个观测都得到隐变量的值.

(2) 用普通最小二乘法得到内部关系的那些系数$\boldsymbol{\beta}$的估计, 也就是得到隐变量之间关系的度量.

(3) 用普通最小二乘法得到外部关系的那些系数$\boldsymbol{\lambda}$的估计, 它们代表了隐变量和显变量之间的关系.

(4) 最后得到各种指数. 以满意度隐变量η_4为例, 由于每个隐变量都有和观测值一样多的估计值, 假定一共有n个观测值, 则有估计的$\hat{\eta}_{4i}(i = 1, 2, ..., n)$. 基于这些估计值, 顾客满意度指数习惯用下面公式来得到:

$$\hat{\eta}_4 \equiv \frac{\frac{1}{n}\sum_i^n \hat{\eta}_{4i} - \min_i(\hat{\eta}_{4i})}{\max_i \hat{\eta}_{4i} - \min_i \hat{\eta}_{4i}} \times 100.$$

下面就是例11.1的计算代码, 利用的是程序包`plspm`的函数`plspm()`. 先把上面的模型输入:

```
library(plspm)
data(satisfaction)
IMAG <- c(0,0,0,0,0,0)
EXPE <- c(1,0,0,0,0,0)
QUAL <- c(0,1,0,0,0,0)
VAL  <- c(0,1,1,0,0,0)
SAT  <- c(1,1,1,1,0,0)
LOY  <- c(1,0,0,0,1,0)
sat.mat <- rbind(IMAG, EXPE, QUAL, VAL, SAT, LOY)
#上面定义的矩阵与内部关系方程中的矩阵相同
#下面语句把显变量分配到与其有关的隐变量确定测量模型
sat.sets <- list(1:5,6:10,11:15,16:19,20:23,24:27)
sat.mod <- rep("A",6)
#上面"A"说明6组显变量都是反映型的("B"意味着显变量是影响型的)
```

上面所谓的“反映型”(reflective)显变量是以隐变量为原因的, 关系为$x_h =$

$\pi_{h0}+\pi_h\xi+\epsilon$, 因此在图模型上箭头是指向显变量的. 而"影响型"(formative)显变量是隐变量的原因, 关系为$\xi=\sum_h \omega_h x_h+\delta$, 在图模型中的箭头是指向隐变量的. 还有混合型的, 即与一个隐变量相关的一部分显变量是反映型的, 另一部分是影响型的. 在满意度模型中, 所有的显变量都是反映型的. 把上面的模型准备好之后, 用下面的语句来计算得到所有的输出:

```
res=plspm(satisfaction,sat.mat,sat.sets,sat.mod,scaled=F)
summary(res)
plot(res, what="all")
```

输出的基本结果(即被估计的各个系数)都显示在图11.2的三个图的相应箭头旁, 这里图太小, 图中的数字不易看清, 读者可自己在R中产生, 看原图, 或者看单独输出的各项结果(后面列出了输出各种信息的相关语句). 其中, 左图为估计的结构模型的路径系数, 中图和右图是估计的测量模型的载荷及权重. 载荷为显变量(的条件期望)用隐变量表示时的系数, 则权重为隐变量(的条件期望)用显变量表示时的系数. 用数学语言表述为: 如果记任意的一个隐变量为ξ, 载荷为满足$E(x_h|\xi)=\pi_{h0}+\pi_h\xi$的$\pi_h$, 则权重为满足$E(\xi|x_1,x_2,...,x_p)=\sum_h w_h x_h$的$w_h$.

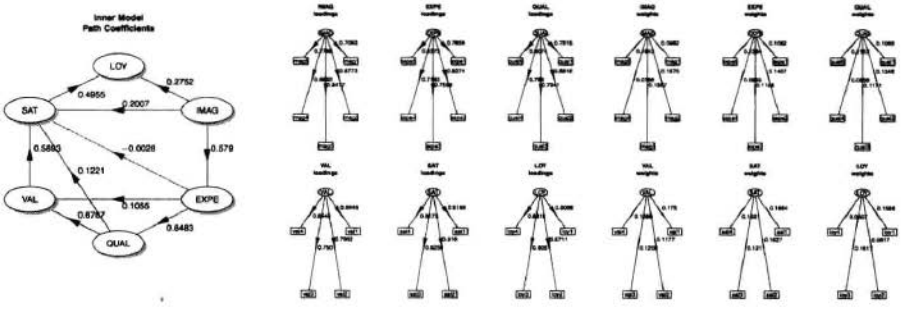


图 11.2 例11.1的PLS解

从图11.2中左图的数字可以看出, 从隐变量EXPE到SAT的路径系数太小, 只有-0.0028, 这说明该模型(至少在这个数据上)的这个关系并不显著(但间接正面效应是显著的, 见后面关于结构模型的效应表评论). 此外, 在实际的关于顾客满意度的应用中, 调查问卷中要回答的问题(也就是显变量)很可能远比这个数据复杂. 越复杂的商品, 问题就越复杂. 比如汽车, 仅仅是感知质量方面就可能有操控性能、节能、刹车、电路、加速等诸多问题. 在满意度调查中, 对每一项的打分习惯上都采用10分制. 在模型设计上, 因果关系一般都是正面关系(即在设计上, 尽量不涉及负面因果关系). 因此如果产生了负数, 就要核对模型是否适合该数据.

除了图11.2中展示的数字之外, 还可以输出涉及相关、拟合度量的各种统计量, 相信读者根据模型方程及图形可以理解它们的意义. 如果计算输出存在res下, 那么, 除了summary(res)语句之外, 得到的所有信息也可以分别以下面的语句输出(表11.1).

表 11.1 输出信息的语句

代码	意义
res$outer.mod	测量模型的结果: 外部权重、标准化载荷、共同度、冗余等
res$inner.mod	结构模型的结果: 路径系数、R^2等
res$latents	标准化隐变量的估计值矩阵
res$scores	隐变量用于估计内部模型的值矩阵
res$out.weights	外部权重向量
res$loadings	标准化载荷向量, 即与隐变量的相关系数
res$path.coefs	路径系数矩阵
res$r.sqr	R^2向量
res$outer.cor	隐变量和可测变量的相关
res$inner.sum	隐变量的汇总表
res$effects	各个隐变量的直接和间接效应
res$unidim	核对隐变量的反映型显变量组的单向度(unidimensionality)
res$gof	绝对、相对、外部模型、内部模型的拟合优度
res$data	原始数据
res$model	结构方程模型(dataset=TRUE时有效)
res$boot	自助法(bootstrapping)结果(boot.val=TRUE时有效)

有几点需要说明:

- 表11.1的单向度定义为一个隐变量为与其正相关的显变量的单一背景变量. 一个关于单向度的度量包括Cronbach's alpha($\alpha > 0.7$时认为该组显变量是单向的), 该度量的公式为(p为与该隐变量相关的显变量个数):

$$\alpha = \frac{Cov(x_h, x_{h'})}{Var\left(\sum_{h=1}^{p} x_h\right)} \times \frac{p}{p-1}.$$

还有Dillon - Goldstein's rho($\rho > 0.7$时认为该组显变量是单向的), Dillon - Goldstein's rho的估计为:

$$\rho = \frac{[\sum_{h=1}^{p} Cor(x_h, t_1)]^2}{[\sum_{h=1}^{p} Cor(x_h, t_1)]^2 + \sum_{h=1}^{p}[1 - Cor^2(x_h, t_1)]}.$$

隐变量单向度的另一个标准为相应显变量组相关阵的第一特征值显著大于第二特征值.

- 对反映型显变量的权重估计为$w_{jh} = Cov(\boldsymbol{x}_{jh}, \boldsymbol{z}_j)$, 这里$\boldsymbol{z}_j$为标准化的隐变量的估计, 而对影响型的显变量的权重估计为$W_j = (\boldsymbol{X}_j^\top \boldsymbol{X}_j)^{-1}\boldsymbol{X}_j^\top \boldsymbol{z}_j$.

作为例子, 表11.1可输出的内容之一为用`res$effects`得到的下面的结构模型的效应表(表11.2), 表中后面三列数字分别代表直接效应、间接效应和总效应. 其中唯一的负值是EXPE->SAT的直接效应(dir.effects), 为-0.0028, 似乎不合乎逻辑, 但是数值很小, 可以忽略. 再看间接效应(ind.effects)及总效应(tot.effects), EXPE对SAT的影响还是正的, 但主要来自间接影响. 这个问题说明直接效应的箭头有些多余. 这是模型和数据不一致的例子.

表 11.2 用res$effects得到的结构模型效应

	relationships	dir.effects	ind.effects	tot.effects
1	IMAG->EXPE	0.5790	0.0000	0.5790
2	IMAG->QUAL	0.0000	0.4912	0.4912
3	IMAG->VAL	0.0000	0.3934	0.3934
4	IMAG->SAT	0.2007	0.2902	0.4910
5	IMAG->LOY	0.2752	0.2433	0.5184
6	EXPE->QUAL	0.8483	0.0000	0.8483
7	EXPE->VAL	0.1055	0.5740	0.6795
8	EXPE->SAT	−0.0028	0.5041	0.5013
9	EXPE->LOY	0.0000	0.2484	0.2484
10	QUAL->VAL	0.6767	0.0000	0.6767
11	QUAL->SAT	0.1221	0.3988	0.5209
12	QUAL->LOY	0.0000	0.2581	0.2581
13	VAL->SAT	0.5893	0.0000	0.5893
14	VAL->LOY	0.0000	0.2920	0.2920
15	SAT->LOY	0.4955	0.0000	0.4955

利用输出信息的语句还可分别输出一些更详细的内容, 比如, 可用`res$outer.mod$SAT`输出`res$outer.mod`中的一条有关变量SAT的内容.

11.3 协方差方法简介

由于协方差方法在各个商业软件中都有, 而且直观方便, 因此这里就不着重介绍其在R上面的计算了, 仅仅给出一个例子来说明.

由于模型的假定, 协方差方法有模型的确认和识别问题, 也就是说, 不是画出图模型, 用数据一算就能够得出结果. 除了分布之外, 还必须有其他假定, 比如, 隐变量方差为1, 均值为0(AMOS默认值). 误差项路径系数为1(AMOS默

认值)或方差为1(可自行设定), 一个隐变量的一个测量变量的路径系数必须为1(AMOS默认值), 内生因变量要有误差项(自己设定). 如果未知参数个数等于已知数据量, 则称为"恰好识别" (just identified, saturated), 这时自由度为0; 如果参数个数小于已知数据量, 则称为"过度识别"(over identified), 这时, 自由度为正, 参数完全可以估计出来; 如参数个数大于数据量, 则称为"不可识别"(under identified), 这时无法做任何事情. 由于路径模型的协方差方法需要涉及许多参数, 判断模型是否可识别并不那么简单, 有大量判断识别状况的方法, 这里不一一介绍. 但最简单的方法是用软件计算一下, 如果可以得出结果, 肯定是可识别的.

除了识别问题, 对于结果的评价也很不简单, 各个软件所提供的和文献所推荐的也不尽相同. 篇幅有限, 不能详述.

协方差方法, 主要称为ML-LISREL方法, 其主要思想是对要预测的协方差阵和样本协方差阵进行比较, 预测的值要使得这两个矩阵的差越小越好. 假定$\boldsymbol{S}$为样本协方差阵, $\boldsymbol{\Sigma}(\theta)$为预测的协方差阵, 则目标的似然函数为:

$$F_{ML} = \ln|\boldsymbol{\Sigma}(\theta)| + \mathrm{tr}[\boldsymbol{S}\boldsymbol{\Sigma}^{-1}(\theta)] - \ln|\boldsymbol{S}| - (p+q),$$

这里, p及q分别为内生观测变量及外生观测变量的数目. 还有一些其他方法, 比如使得下式最小的方法(不加权最小二乘)

$$F_{ULS} = \frac{1}{2}\mathrm{tr}[(\boldsymbol{S} - \boldsymbol{\Sigma}(\theta))^2],$$

或者使得下式最小的方法(广义最小二乘)

$$F_{GLS} = \frac{1}{2}\mathrm{tr}\{[(\boldsymbol{S} - \boldsymbol{\Sigma}(\theta))W^{-1}]^2\}.$$

还有利用工具变量的两步最小二乘法等. 它们都基于使得$\boldsymbol{S}$和$\boldsymbol{\Sigma}(\theta)$尽可能接近的原理.

下面就例11.1的欧洲顾客满意度数据(satisfaction.csv)来演示如何在R中用结构方程模型拟合数据. 用协方差方法解结构方程模型的商业软件很多, 在R中类似的程序包却不多, 这里用的是程序包`lavaan`.[1] 该程序包比程序包`sem`要好用.

我们用和前面11.2节使用PLS方法时完全一样的结构方程模型, 仅有的差别是增加少许相关关系(协方差关系). 为此, 在建立模型时使用的代码为:

```
w=read.csv("satisfaction.csv")
library(lavaan)
```

[1] Yves Rosseel (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software*, 48(2), 1-36. URL http://www.jstatsoft.org/v48/i02/.

```
model<-"
#外部关系:
IMAG=~imag1+imag2+imag3+imag4+imag5
EXPE=~expe1+expe2+expe3+expe4+expe5
QUAL=~qual1+qual2+qual3+qual4+qual5
VAL=~val1+val2+val3+val4
SAT=~sat1+sat2+sat3+sat4
LOY=~loy1+loy2+loy3+loy4
#内部关系:
EXPE~IMAG
QUAL~EXPE
VAL~EXPE+QUAL
SAT~EXPE+QUAL+VAL+IMAG
LOY~IMAG+SAT
#协方差关系:
imag1~~imag2+imag3+imag4+imag5
imag2~~imag3+imag4+imag5
imag3~~imag4+imag5
imag4~~imag5"
```

代码中的内部关系和外部关系与前面11.2节使用PLS方法时相同, 只有后面的协方差关系是加上去的, 它意味着所有和隐变量IMAG有关的可观测变量是两两相关的. 注意, PLS方法假定所有的变量都相关, 因此不存在这个问题. 这里还不能加太多协方差关系, 否则会算不出来, 这也是协方差方法的缺陷之一.

下一步是用这样定义的模型model拟合这个数据, 并打印出各种结果. 相应的代码为:

```
fit <- sem(model, w)
summary(fit, fit.measures = TRUE)
```

得到输出:

```
> fit <- sem(model, w)
> summary(fit, fit.measures = TRUE)
lavaan (0.5-17) converged normally after 130 iterations

  Number of observations                           250
```

```
  Estimator                                      ML
  Minimum Function Test Statistic          1096.353
  Degrees of freedom                            304
  P-value (Chi-square)                        0.000

Model test baseline model:

  Minimum Function Test Statistic          5261.366
  Degrees of freedom                            351
  P-value                                     0.000

User model versus baseline model:

  Comparative Fit Index (CFI)                 0.839
  Tucker-Lewis Index (TLI)                    0.814

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)          -12337.606
  Loglikelihood unrestricted model (H1)  -11789.430

  Number of free parameters                      74
  Akaike (AIC)                            24823.212
  Bayesian (BIC)                          25083.800
  Sample-size adjusted Bayesian (BIC)     24849.214

Root Mean Square Error of Approximation:

  RMSEA                                       0.102
  90 Percent Confidence Interval      0.096   0.109
  P-value RMSEA <= 0.05                       0.000

Standardized Root Mean Square Residual:

  SRMR                                        0.066
```

Parameter estimates:

Information Standard Errors	Estimate	Std.err	Z-value	Expected Standard P(>|z|)
Latent variables:				
IMAG =~				
imag1	1.000			
imag2	1.688	0.211	8.011	0.000
imag3	1.627	0.224	7.249	0.000
imag4	0.803	0.139	5.795	0.000
imag5	2.122	0.339	6.262	0.000
EXPE =~				
expe1	1.000			
expe2	1.291	0.113	11.462	0.000
expe3	1.096	0.116	9.445	0.000
expe4	0.873	0.086	10.122	0.000
expe5	1.278	0.118	10.822	0.000
QUAL =~				
qual1	1.000			
qual2	1.279	0.094	13.592	0.000
qual3	1.092	0.098	11.110	0.000
qual4	0.917	0.075	12.200	0.000
qual5	1.093	0.089	12.297	0.000
VAL =~				
val1	1.000			
val2	0.736	0.055	13.353	0.000
val3	0.735	0.069	10.726	0.000
val4	0.953	0.066	14.330	0.000
SAT =~				
sat1	1.000			
sat2	0.962	0.044	21.709	0.000
sat3	0.759	0.052	14.707	0.000
sat4	0.776	0.054	14.243	0.000
LOY =~				

```
    loy1              1.000
    loy2              0.530    0.056    9.538    0.000
    loy3              1.008    0.064   15.726    0.000
    loy4              0.582    0.062    9.412    0.000

Regressions:
  EXPE ~
    IMAG              1.108    0.468    2.365    0.018
  QUAL ~
    EXPE              1.133    0.107   10.598    0.000
  VAL ~
    EXPE             -5.346   26.720   -0.200    0.841
    QUAL              5.756   23.515    0.245    0.807
  SAT ~
    EXPE             -3.776   34.178   -0.110    0.912
    QUAL              3.280   33.447    0.098    0.922
    VAL               0.771    3.225    0.239    0.811
    IMAG              0.438    0.405    1.083    0.279
  LOY ~
    IMAG              0.482    0.385    1.254    0.210
    SAT               0.804    0.144    5.565    0.000

Covariances:
  imag1 ~~
    imag2             0.721    0.542    1.331    0.183
    imag3             0.385    0.518    0.744    0.457
    imag4             0.677    0.293    2.313    0.021
    imag5            -0.321    0.667   -0.481    0.630
  imag2 ~~
    imag3             0.442    0.842    0.525    0.599
    imag4             0.217    0.437    0.497    0.619
    imag5            -0.681    1.087   -0.626    0.531
  imag3 ~~
    imag4             0.240    0.421    0.571    0.568
    imag5            -0.830    1.047   -0.793    0.428
  imag4 ~~
```

imag5	-0.430	0.542	-0.793	0.428

Variances:

imag1	2.110	0.383
imag2	1.237	0.883
imag3	1.133	0.820
imag4	1.809	0.274
imag5	1.591	1.387
expe1	2.060	0.196
expe2	1.714	0.176
expe3	3.013	0.284
expe4	1.459	0.140
expe5	2.304	0.227
qual1	1.902	0.184
qual2	1.562	0.167
qual3	2.860	0.272
qual4	1.392	0.137
qual5	1.912	0.188
val1	1.390	0.155
val2	1.508	0.148
val3	2.764	0.259
val4	2.017	0.204
sat1	0.704	0.090
sat2	0.749	0.091
sat3	1.579	0.153
sat4	1.792	0.172
loy1	1.673	0.246
loy2	2.748	0.262
loy3	1.901	0.263
loy4	3.427	0.326
IMAG	0.775	0.377
EXPE	0.805	0.395
QUAL	0.014	0.064
VAL	0.342	1.801
SAT	0.157	0.759
LOY	1.802	0.284

可以用下面代码打印出各种需要的输出结果(比如R^2、系数、均值、协方差矩阵, 等等):

```
inspect(fit, "free")
inspect(fit, "start")
inspect(fit, "rsquare")
inspect(fit, "fit")
fitted.values(fit)
coef(fit)
resid(fit, type="normalized")
```

协方差方法模型的评价指标有很多类, 每类又有许多个, 比如各种吝啬度量、各种吝啬调整的度量、各种信息度量、ECVI的置信区间、各种与基线模型比较的指标以及至少10种其他未归类的指标等. 对于各种指标需要注意:

- 指标没有强有力的理论基础来支持数据的意义和原理.
- 不同指标的优劣一直有争议.
- SEM模型检验是基于理论的, 最佳指标完全依赖于理论上的假定.
- 不适当的参数应事先考察.
- 初始模型必须具有合理的理论基础, 否则再多修饰也于事无补.

11.4 结构方程模型的一些问题

结构方程模型有许多特有的、与其他统计模型不同的问题, 下面就其中的一些问题予以讨论.

构造图模型的时候, 必须进行因果关系的论证. 没有任何根据就在图上画几个椭圆和矩形, 再想当然地加几个箭头必然会失败. 结构方程模型分析的推论有很大的局限性, 这是因为结构方程模型本身无法创造理论或知识, 一定要避免过度依赖技术指标进行过度推断. 此外, 解释结构方程模型时要明白, 各参数显著和拟合很好等是变量因果关系的必要条件而不是充分条件, 应该避免根据参数的显著性盲目乐观地给出因果结论.

在作决策时, 对模型优劣避免武断和绝对化, 应该尽可能检验不同的路径模型, 先进行测量模型的检验, 然后进行结构模型的参数估计, 再做模型评估. 模型是否适当, 不能只考虑统计意义, 还要考虑概念的本质与意义, 注意隐变量的多重指标是否有理论和实际意义.

对协方差方法来说, 对于模型拟合的评估, 应参考各种指标, 不能仅选取符合自己想象的. 应尽可能事先检验数据的分布特征, 测量变量应尽可能符合正态分布的假定, 选择简单、参数较少的模型, 样本量则越大越好.

在使用结构方程模型时, 处处都可能出现问题. 比如, 在收集数据后才确定模型是一个疏忽; 隐变量有可能缺乏足够的测量变量的支持; 因果关系可能完全弄颠倒了; 为追求拟合, 而改变模型形式, 使其复杂化; 使用协方差方法时, 在没有充分理由的情况下, 将测量残差或解释残差的相关性纳入估计或者使用多维假定(一个测量变量受一个以上隐变量的影响).

在使用结构方程模型作结果报告时, 绝不能仅关心模型的拟合, 忽视了其他指标和模型的合理性. 把模型拟合解读为模型被支持的唯一或主要证据, 或者把模型拟合解读为内生变量被有效解读的证据是最常见的错误. 在使用协方差方法时, 往往过度依赖显著性检验结果, 错误解释标准化参数估计. 人们经常误认为结构方程模型可以弥补设计不周或考虑不周的研究. 我们应该把充分的信息和正确的解读提供给服务对象, 使他人能够理解或复制研究结果. 科学应该是可以重复的, 研究结论的推导过程应该是完全透明的.

第十二章　贝叶斯网络

12.1　引言

前一章讨论了图模型(结构方程模型)及基于完全不同理念的两种处理方法, 这里讨论属于概率图形(probabilistic graphical models, GM)族的另一种网络模型, 即**贝叶斯网络**(Bayesian network, BN).[1] 贝叶斯网络包括图理论、概率论、计算机科学及统计的概念.

虽然贝叶斯网络并不属于严格意义上的贝叶斯统计范畴, 比如它经常采用频率派的方法来估计条件概率分布的参数, 但贝叶斯网络是多层贝叶斯模型的非常有用的表示, 而多层贝叶斯模型又是应用贝叶斯统计的基础.

贝叶斯网络包含类似于图模型的有向无回路图(directed acyclic graph, DAG). DAG是由节点和有向边(箭头)组成的. 无回路意味着没有节点是它自己的祖先或后代, 也就是顺着箭头不会回到已经走过的节点. DAG的每个节点(node)都相应于一个变量, 箭头代表着变量之间的相依关系. 每个箭头的终点称为"子节点"(child), 起点称为"父节点"(parent). 和图模型类似, 没有父节点的节点称为根节点(root node), 没有子节点的节点称为叶节点(leaf node), 其他节点称为中间节点(intermediate node). 每个变量都附有给定其父节点的(条件)局部概率, 因此贝叶斯网络也称为概率有向无回路图模型(probabilistic directed acyclic graphical model), 这就区别于其他具有DAG的图模型.

贝叶斯网络广泛应用于统计、机器学习、人工智能等领域. 贝叶斯网络数学上严格而又易于理解. 虽然箭头代表变量间直接的因果关系, 但推理能够作用于任何方向(无论有没有箭头). 这和图模型/结构方程模型很不一样.

虽然贝叶斯网络看上去有些类似于图模型, 也有些像神经网络. 但实际上, 贝叶斯网络和前面所说的各种图形网络模型完全不同. 从方法上来说, 也没有什么共同点. 贝叶斯网络不像完全主观建造而又很难更新或修改的结构方程模型, 它根据数据中变量之间的相关来构建图中的变量联系, 同时也可以加以主观的介入. 因此, 贝叶斯网络从某种意义上来说比图模型更加客观, 与数据所反映的现实世界的关系也更加密切.

例 12.1　这是一个假想的描述性例子. 一共有5个变量, 它们的取值均为"是"或"非": 变量P代表X先生所住区域停电; 变量I代表互联网公司故障; 变量W表示X先生和外面世界失去了互联网联系; 变量J代表X先生失去正在

[1] 贝叶斯网络有很多英文名称: Bayesian network, Bayes network, belief network, Bayes(ian) model.

等待的某工作机会; 变量M代表X先生失去某网上投资机会. 这5个变量都是二分变量, 取哑元值1(代表"是")或者0(代表"非"). 图12.1为这样一个例子的贝叶斯网络的DAG图形.

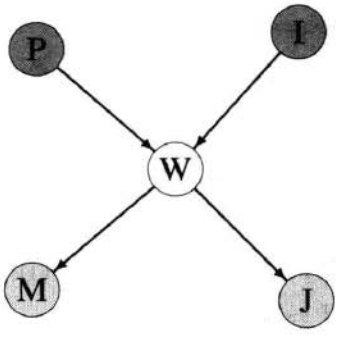


图 12.1 例12.1的贝叶斯网络

在图12.1中, 一共有5个节点, 每个节点代表一个变量. 仅仅这个图本身还不足以构成贝叶斯网络, 对于每个节点, 还必须有给定其父节点(上级节点)时的条件概率. 在这个例子中, 四个箭头显示了下面所谓的"因果关系"[1]:

- 停电(P)可能造成X先生失去互联网联系(W), 即$P \to W$.
- 互联网故障(I)可能造成X先生失去互联网联系(W), 即$I \to W$.
- 失去互联网联系(W)可能造成X先生失去某工作机会(J), 即$W \to J$.
- 失去互联网联系(W)可能造成X先生失去某网上投资机会(M), 即$W \to M$.

所有的事件都不是确定性的, 比如停电, 但可能会有备用电源; 互联网不通, 可通过手机或其他无线方式上网; 即使真的失去互联网联系, 也不一定会失去某工作机会或者失去某投资机会. 这些都用局部概率表示, 即给定父节点后一个节点取值的条件概率. 对于我们的例子, 可以设定局部概率(根节点的概率不是条件概率):

- 节点P的分布$P(P)$: $P(P=1)=0.01$.
- 节点I的分布$P(I)$: $P(I=1)=0.03$.
- 节点W的条件分布$P(W|P,I)$:

$$P(W=1|P=1,I=1)=0.98; \quad P(W=1|P=1,I=0)=0.95;$$

[1]在大多数情况下, 并不一定是真正的因果关系.

$$P(W=1|P=0,I=1)=0.05;\quad P(W=1|P=0,I=0)=0.001.$$

- 节点J的条件分布$P(J|W)$:

$$P(J=1|W=1)=0.60,\ P(J=1|W=0)=0.50.$$

- 节点M的条件分布$P(M|W)$:

$$P(M=1|W=1)=0.10,\ P(M=1|W=0)=0.08.$$

从独立性来说, 显然, 例12.1的变量P和I是独立的, 在给定W之后, J和M应该是独立的.

我们可以算联合分布, 比如对于例12.1, 一个联合分布的值为:

$$\begin{aligned}&P(J=1,M=1,W=1,P=0,I=0)\\=&P(J=1|W=1)P(M=1|W=1)P(W=1|P=0,I=0)\\&\times P(P=0)P(I=0)\\=&0.60\times 0.10\times 0.001\times 0.99\times 0.97\\=&5.7618\times 10^{-5}\end{aligned}$$

根据例12.1模型的联合分布可以由贝叶斯定理回答诸如"如果找到工作, 没有停电的概率是多少(即$P(P=0|J=1)$)"这样的问题. 这需要计算:

$$P(P=0|J=1)=\frac{P(P=0,J=1)}{P(J=1)}=\frac{\sum_{M,W,I}P(P=0,J=1,M,W,I)}{\sum_{M,W,I,P}P(J=1,P,M,W,I)},$$

式中, 分子、分母是对所有多余变量可能取的值(这里是0及1)求和. 而联合概率又是由贝叶斯网络所设定的变量之间的关系计算出来的, 比如分子:

$$\begin{aligned}&\sum_{M,W,I}P(P=0,J=1,M,W,I)=\sum_{W,I}P(P=0,J=1,W,I)\\=&\sum_{W,I}\{P(J=1|W)P(W|P=0,I)P(P=0)P(I)\}\\=&\sum_{W\in\{0,1\}}\sum_{I\in\{0,1\}}\{P(J=1|W)P(W|P=0,I)P(P=0)P(I)\}.\end{aligned}$$

注意, 上式第一个等式利用了J和M在给定W时的条件独立性. 分母的运算也类似.

显然, 利用贝叶斯网络变量的已有关系来计算联合分布会有较高的效率. 比如对于例12.1的联合分布

$$P(J,M,W,P,I)=P(P)P(I)P(W|P,I)P(J|W)P(M|W),$$

一共有$1+1+4+2+2=10$个参数, 而在没有这些关系时, 有$2^5=32$个参数.

12.2 贝叶斯网络涉及的一些基本概念

12.2.1 贝叶斯网络

形式上贝叶斯网络定义为$\boldsymbol{B} = (\boldsymbol{G}, \Theta)$, 这里$\boldsymbol{G}$为DAG, 节点为随机变量$X_1, X_2, ..., X_n$, 边代表变量之间的直接相依关系. 贝叶斯网络满足因果**Markov假定**(Markov assumption), 也就是说, **贝叶斯网络中的每个变量在给定其父节点的条件下, 独立于其祖先**. 因而, 图形$\boldsymbol{G}$刻画了**独立性假定**(independence assumptions), 即**给定$\boldsymbol{G}$中的父节点后, 每个变量独立于其$\boldsymbol{G}$中的非后代**. Θ代表了网络参数集, 包括对于X_i的每个实现x_i以π_i为条件的参数, $\theta_{x_i|\pi_i} = P_B(x_i|\pi_i)$, 这里$\pi_i$为$\boldsymbol{G}$中$X_i$的父节点集合, 因此, 贝叶斯网络$\boldsymbol{B}$定义了所有变量集合的唯一联合概率分布, 按照独立性假定:

$$P_B(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P_B(X_i|\pi_i) = \prod_{i=1}^{n} \theta_{X_i|\pi_i}.$$

如果没有独立性假定, 根据条件分布的链原理,

$$P_B(X_1 = x_1, ..., X_n = x_n) = \prod_{i=1}^{n} P_B(X_i = x_i|X_{i+1} = x_{i+1}, ..., X_n = x_n).$$

而根据独立性假定,

$$P_B(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = \prod_{i=1}^{n} P_B(X_i = x_i|X_j = x_j, \forall X_j \in \pi_i).$$

显然独立性假定大大简化了联合分布.

12.2.2 贝叶斯网络中的条件独立性概念

下面要介绍的d分离概念对于理解贝叶斯网络各个节点之间的条件独立或相依关系十分有帮助.

因果Markov假定建立了贝叶斯网络中的条件独立性. 据此, 对于贝叶斯网络中的一般条件独立性和依赖性, Pearl (1988)提出了**d分离**(d-separation, direction-dependent separation)的概念. 它有下面的含义: 如果两个节点集合X和Y在贝叶斯网络中被不包括X和Y的第三个节点集合Z 造成d分离, 那么, 对于给定的Z中的变量, 对应的变量集合X和Y是独立的. 这里d分离的定义为: 两个节点集合X和Y在贝叶斯网络中被不包含X和Y的第三个集合d分离, 当且仅当X和Y之间的每个路径被"阻塞"(blocked), 这里"阻塞"是指有一个不来自X和Y的中间变量V, 使得

(1) X和Y通过中间点V被连接(如$X \to V \to Y$或$X \leftarrow V \leftarrow Y$), 或者

(2) X和Y被一个共同原因连接, 而V是这个共同原因($X \leftarrow V \rightarrow Y$), 或者
(3) X和Y被一个共同效果连接, 但V不是共同效果或共同效果之一($X \rightarrow \{\text{不是}V\} \leftarrow Y$).

(1)(2)比较好理解, 如果X和Y不独立, 这是由于V造成的. 如果控制了V, 即以V为条件, 则去掉了X和Y的相依性, 它们是条件独立的. 换言之, V阻塞了X和Y之间的通路.

至于(3), 在V给定时, X和Y独立(即V阻塞了它们间的通路)只有当V不是它们共同效果的函数时才成立. 这是因为, 如果V是共同效果或其函数, 那么以V为条件将导致X和Y变得相依. 如果X和Y原本独立, 而且$X \rightarrow V \leftarrow Y$(或者$X \rightarrow U \leftarrow Y$而且$U \rightarrow V$等), 那么在知道$V$和$X$的信息时, 人们就知道了$Y$的信息, 这意味着给定$V$时, X和Y条件不独立. 图12.2给出了V节点阻塞(未阻塞)X和Y的几种情况.

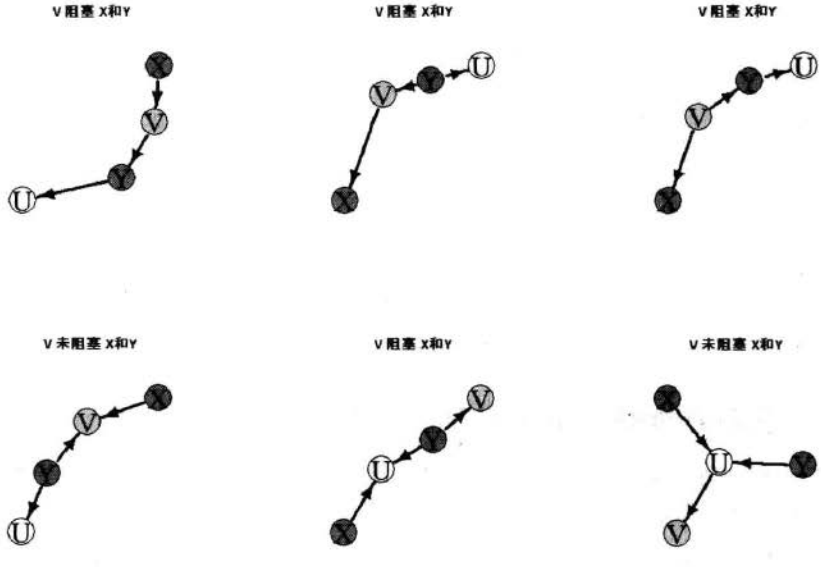


图 12.2 V节点阻塞(未阻塞)X和Y的几种情况

以例12.1来说, 本来P(停电)和I(互联网公司故障)互相独立, 当一个事件发生时, 人们并不知道另一个事件是否发生. 但如果知道W(X先生的互联网断了)发生了, 又知道P没有发生(未停电), 这就意味着I很可能发生了. 这说明在给定W时, P和I就不独立了.

Verma and Pearl (1988)证明, 如果集合V在一个贝叶斯网络中d分离X和Y, 那么给定V时, X和Y条件独立. 而最小的使得节点A和网络中其他节点d分离的节点集合是A的**Markov毯**(Markov blanket, MB). 贝叶斯网络中节点A的Markov毯为A的父节点、子节点及子节点的父节点的集合. 也就是说, 给定其Markov毯时, 节点独立于所有其他点.

图12.3显示了贝叶斯网络中节点E的Markov毯, 它包括E的父节点节点C和D, 子节点节点G和F, 以及子节点G的父节点节点H.

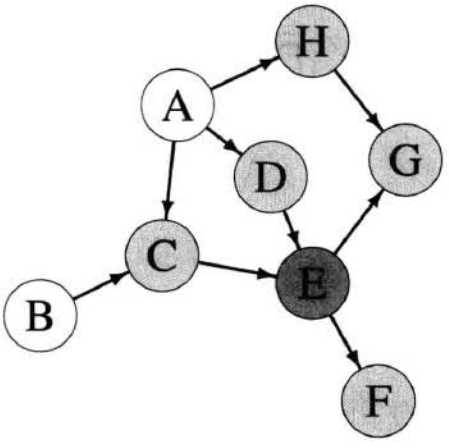


图 12.3 贝叶斯网络节点E的Markov毯

12.3 通过贝叶斯网络做推断

给出贝叶斯网络的因子化形式的联合概率分布, 就可以通过对所有"无关"变量求和来得到边缘分布以做推断. 一般考虑两种推断:

- 通过其父节点(自上而下推理)的证据对X_i节点的预测.
- 通过其子节点(自下而上推理)的证据对X_i节点的诊断.

前面一开始已经介绍过利用贝叶斯网络做推断的例子, 这里再看一个例子.

例 12.2 驾车(Driver.csv) 考虑几个二分变量: Y (Young, 是否年轻), D (Drink, 是否饮酒), A (Accident, 是否有过事故), V (Violation, 是否违规过), C (Citation, 是否收到过罚单), G (Gear, 是否自动档). 数据

中用的是0,1哑元变量, 对每个变量, "是"相应于1, "否"相应于0. 图12.4为相应的DAG图. 这张DAG图说明了各个点存在的独立性和相关性, 而箭头表示假定的因果关系. 这里变量Accident, Citation, Violation的父节点均为Young和Drink.

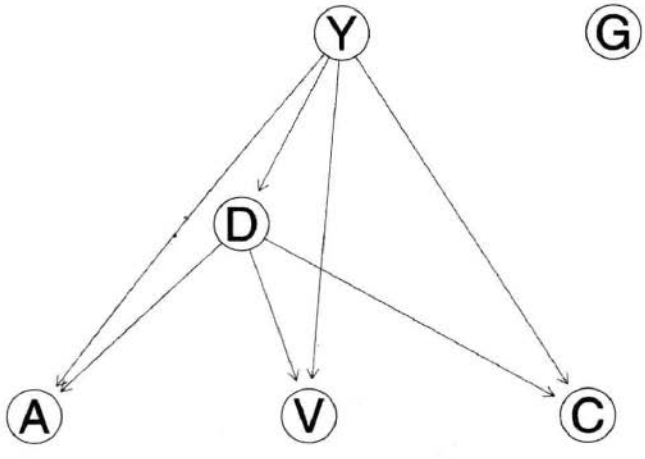


图 12.4 例12.2的DAG图

下面对于例12.2做自下而上的一个推断:

$$P(Y=1|A=1)=\frac{P(Y=1,A=1)}{P(A=1)}$$

$$P(Y=1,A=1)=\sum_{G,D,V,C\in\{0,1\}} P(Y=1)P(G)P(D|Y=1)$$
$$\times P(V|Y=1,D)P(A=1|Y=1,D)P(C|Y=1,D)$$

$$P(A=1)=\sum_{Y,G,D,C,V\in\{0,1\}} P(Y)P(G)P(D|Y)P(C|Y,D)$$
$$\times P(V|Y,D)P(A=1|Y,D)$$

这里利用的贝叶斯网络推断相对一般情况的推断节省了许多资源. 如果不知道诸如贝叶斯网络这样的结构, 即使是这里的二分变量例子, 对于n个变量, 联合概率分布的可能为$O(2^n)$的. 如此计算称为精确推断(exact inference), 一般称为NP-hard (Non-deterministic Polynomial-time hard)问题. 对于一些有约束的网络, 可解决精确推断问题的算法包括消息传递算法(message passing algorithm), 它只有$O(n)$种可能, 在两个节点中限制最多一条路径, 参见Pearl

(1988)和Pearl and Russel (2001). 该方法被推广到一般网络, 参见Lauritzen and Spiegelhalter (1988). 其他精确推断包括循环子集抽样(cyclecutset conditioning)(Pearl, 1988)和变量删除(variable elimination)(Zhang and Poole, 1996).

还有近似方法, 比如Monte Carlo抽样给出逐渐改进的估计(Pearl, 1987). MCMC方法包括Gibbs抽样和Metropolis - Hastings算法(Griffiths and Yuille, 2006). 其他方法包括信念传递(loopy belief propagation)和利用大数定律通过均值近似大量随机变量的和的变分方法(variational methods)(Jordan, et al., 1998).

12.4 学习贝叶斯网络

实际上, 贝叶斯网络往往是未知的, 需要从数据学习. 首先给出训练集和先验信息(比如专家知识、因果关系), 然后估计图形拓扑(网络结构)和贝叶斯网络中的联合概率分布的参数. 学习贝叶斯网络结构比学习贝叶斯网络的参数更难, 而且还可能有部分可观测的障碍, 比如有隐变量和缺失值.

学习贝叶斯网络不是一个单纯的数学或概率过程, 还要了解数据所代表的研究对象领域的知识, 要寻求最有效的方法和途径, 要考虑概率的变化方向以及如何处理缺失数据. 由于贝叶斯网络是用有向箭头联系的, 学习中必须考虑可能的因果关系.

12.4.1 网络学习算法的种类

从算法的角度看, 贝叶斯网络结构学习算法能够分成两类. 第一类是基于约束的算法(constraint-based algorithms), 利用分析贝叶斯网络Markov性的条件独立性检验来分析其概率关系, 比如把搜索局限于节点的Markov毯来优化, 然后构造统计上相应于d分离的图形. 往往把边(弧或箭头)的所有方向都看成三元V结构(如$X_j \to X_i \to X_k$, $X_j \to X_i \leftarrow X_k$, $X_j \leftarrow X_i \to X_k$)的一部分. 出于主观的经验, 或者为了确保无回路条件满足, 可能还要增加一些约束. 最终, 即使是从观测数据来学习, 得到的模型也常常解释为因果模型.

第二类是基于记分的算法(score-based algorithms). 这个算法给每个候选的贝叶斯网络一种记分, 这些记分的定义多种多样, 但都是按照某些标准衡量网络. 有了记分标准之后, 就可以利用诸如吝啬搜索(greedy search)、登山(hill-climbing) 或禁忌搜索(tabu search)等直观搜索方法来得到使记分最大的网络结构. 记分函数通常是记分等价的(score-equivalent), 也就是说, 具有同样概率分布的网络有同样的记分. 记分种类很多, 比如似然或对数似然记分、AIC及BIC记分、离散变量的贝叶斯Dirichlet后验密度记分、K2记分, 以

及对连续变量的正态的Wishart后验密度记分, 等等.

12.4.2　几种可能面对的问题

一般将可能需要面对的问题划分成四种情况(表12.1), 每种情况都可能相应于上面提到的不同种类的学习方法. 下面对各种情况做出有限的概括介绍.

表 12.1　四种可能面对的问题

图形	数据	学习方法举例
已知	完全	最大似然法(ML)、最大后验概率法(MAP)
已知	部分	EM算法、梯度上升法、MCMC
未知	完全	模型空间搜索
未知	部分	EM+搜索模型空间

1.　已知图形及完全的数据

表12.1中第一种情况最容易处理. 假定学习目标是找到使得训练数据的似然函数最大化的每个条件概率分布的参数值, 这也是最简单的最大似然法. 考虑有m个独立观测的训练集$\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m\}$, 这里$\boldsymbol{x}_\ell = (x_{\ell 1}, x_{\ell 2}, ..., x_{\ell n})^\top$. 参数集$\Theta = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_n)$, 这里$\boldsymbol{\theta}_i$为变量(节点)$X_i$的条件分布的参数集. 训练集的对数似然为(和号中每一项对应一个节点):

$$\ln L(\boldsymbol{X}|\Theta) = \sum_{\ell=1}^{m} \sum_{i=1}^{n} \ln P(x_{\ell i}|\pi_i, \boldsymbol{\theta}_i).$$

该对数似然记分函数按照已知的图结构分解, 然后独立地最大化每个节点对对数似然记分的贡献. 当然这里要假定每个节点的参数独立于其他节点的参数.

下面考虑例12.2中的贝叶斯网络. 如果我们有一系列训练数据, 而且想要估计节点C的条件概率表. 我们记录各种可能的数据, 比如$\#(C=1, D=1, Y=1), \#(C=1, D=0, Y=1), \#(C=1, D=0, Y=0)$等(一共有8个), 这里符号$\#(C=c, D=d, Y=y)$表示事件$(C=c, D=d, Y=y)$出现的次数. 有了这些数据, 我们可以找到$C$的条件概率分布参数$\theta_{c|\pi_i} \equiv P(C=c|\pi_i)$的最大似然估计.

$$\begin{aligned} P(C=c|D=d, Y=y) &\approx \frac{\#(C=c, D=d, Y=y)}{\#(D=d, Y=y)} \\ &= \frac{\#(C=c, D=d, Y=y)}{\sum\limits_{c\in\{0,1\}} \#(C=c, D=d, Y=y)} \end{aligned}$$

一般来说, $\theta_{x|\pi_X}$的最大似然估计为:

$$ML(\theta_{x|\pi_X}) = \frac{\#(x|\pi_X)}{\sum_x \#(x|\pi_X)}.$$

另一种方法是最大后验概率法(maximum a posteriori, MAP). 在最大似然法中, 考虑的是$\max_\Theta \ln P(\boldsymbol{X}|\Theta)$, 而在MAP中, 考虑的是$\max_\Theta \ln P(\boldsymbol{X}|\Theta)P(\Theta)$, 这里的$P(\Theta)$是先验分布. 和最大似然法一样, 这里也要假定各个节点参数的独立性, 以便于把联合概率分布分解. 对于多项分布$Mu(N, p_1, ..., p_k)$的例子, 先验分布一般取与其共轭的Dirichlet分布$Di(\theta_{\pi_i}|\alpha_{1|\pi_i}, ..., \alpha_{k|\pi_i})$. 这时

$$MAP(\theta_{x|\pi_X}) = \frac{\#(x|\pi_x) + \alpha_{x|\pi_X}}{\sum_x \#(x|\pi_X) + \sum_x \alpha_{x|\pi_X}},$$

这里的$\{\alpha_{x|\pi_X}\}$可以是等价的样本量, 算是先验知识.

对于这种离散的多项分布情况, 仅仅计数就可以得到结果. 而对于连续的情况, 通常节点分布为正态分布, 那时计算样本均值和样本方差, 再通过线性回归来估计权数矩阵. 还可以给每个参数向量设定先验概率密度, 并利用训练集计算后验参数分布和贝叶斯估计.

2. 已知图形但数据不完全

在已知结构及部分观测的情况, 能够用EM(expectation maximization)算法找到参数的局部最优最大似然估计. EM算法从包括初始参数的目前图形$(\boldsymbol{G}, \Theta)$开始, 然后利用贝叶斯网络的推断来计算期望(在多项分布情况下为期望计数)

$$E_X[\#(x|\pi_x)] = \sum_k P(x|\pi_x, \boldsymbol{G}, \Theta),$$

这是E步骤, 再把它通过ML或者MAP方法最大化以更新参数(M步骤), 之后回到E步骤继续迭代直到收敛.

就例12.2网络C节点的情况, 把观测的事件计数用期望计数来代替:

$$P(C = c|D = d, Y = y) = \frac{E[\#(C = c, D = d, Y = y)]}{E[\#(D = d, Y = y)]}.$$

对于一个事件(比如A), 这里$E[\#(A)]$的计算为:

$$E[\#(A)] = E\left[\sum_i I(A|D(i))\right] = \sum_i P(A|D(i)),$$

指标函数$I(A|D(i))$定义为:

$$I(A|D(i)) = \begin{cases} 1, & \text{事件}A\text{在第}i\text{个观测值出现;} \\ 0, & \text{事件}A\text{在第}i\text{个观测值未出现.} \end{cases}$$

给定期望计数后通过最大化似然记分估计参数, 然后再次计算期望计数, 如此迭代, 最终收敛到似然表面的局部最大值. 也可以在似然表面用梯度上升法, 这需要调整步长和约束参数的取值, 不如EM算法简单.

3. 未知图形但数据完全

对于寻求未知结构但有完全观测的情况, 最一般的是基于记分的搜寻方法(score+search paradigm), 它把学习网络看成优化问题的组合. 搜寻方法是在与贝叶斯网络关联的空间中搜寻, 而记分函数(scoring function)则是评估该空间对数据的拟合程度的度量或准则. 形式上, 记分搜寻方法可描述为: 给定完全的m个独立观测的训练集$\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m\}$, 这里$\boldsymbol{x}_\ell = (x_{\ell 1}, x_{\ell 2}, ..., x_{\ell n})^\top$, 在有$n$个节点的所有可能网络图形$\mathcal{G}_n$中, 寻求一个DAG $\hat{\boldsymbol{G}}$, 满足

$$\arg\max_{\boldsymbol{G}\in\mathcal{G}_n} g(\boldsymbol{G}|\boldsymbol{X}).$$

这里的$g(\boldsymbol{G}|\boldsymbol{X})$是度量候选$\boldsymbol{G}$对数据拟合程度的记分函数, 而$\mathcal{G}_n$为定义在这$n$个节点或变量的所有DAG.

最大化似然记分是最直接的想法, 它有大量的参数, 可以最好地拟合数据, 但可能会有过拟合. 为避免过拟合, 在模型上加先验分布$P(\Theta)$, 这样, 根据贝叶斯定理, 我们要最大化后验分布

$$P(\Theta|\boldsymbol{X}) = \frac{P(\boldsymbol{X}|\Theta)P(\Theta)}{P(\boldsymbol{X})} \propto P(\boldsymbol{X}|\Theta)P(\Theta).$$

取对数, 得到

$$\ln P(\Theta|\boldsymbol{X}) \propto \ln P(\boldsymbol{X}|\Theta) + \ln P(\Theta).$$

这里$P(\Theta)$的作用就像复杂模型的惩罚项一样.

一种由Cooper and Herskovits(1992)建议的记分函数称为K2, 它依赖于几个假定: 多项分布, 没有缺失值, 参数独立, 参数模块化及在给定网络结构时参数先验分布均匀化. 而K2记分又是Heckerman et al. (1995)提出的BD(Bayesian Dirichlet)记分的一个特例. 在这里我们使用de Campos(2006)的符号: X_i的取值有r_i个, 其父节点π_i设置有q_i个, $q_i = \prod_{X_j\in\pi_i} r_i$; $w_{ij}(j = 1, 2, ..., q_i)$代表$\pi_i$的设置; m_{ijk}为数据集$\boldsymbol{X}$中的X_i取值x_{ik}以及π_i的变量取值w_{ij}的观测值数目; m_{ij}是π_i的变量取其第j个设置w_{ij}的观测值数目;

$m_{ij}=\sum_{k=1}^{r_j} m_{ijk}$; $m_{ik}=\sum_{j=1}^{q_i} m_{ijk}$. 下面是BD记分形式上的定义:

$$g_{BD}(\boldsymbol{G}|\boldsymbol{X})=\ln(p(\boldsymbol{G}))+\sum_{i=1}^{n}\left\{\sum_{j=1}^{\eta_i}\left[\ln\left(\frac{\Gamma(\eta_{ij})}{\Gamma(m_{ij}+\eta_{ij})}\right)+\sum_{k=1}^{r_i}\ln\left(\frac{\Gamma(m_{ijk}+\eta_{ijk})}{\Gamma(\eta_{ijk})}\right)\right]\right\},$$

这里$\Gamma(\cdot)$为Gamma函数, 如果c为整数, 则$\Gamma(c)=(c-1)!$, 如果再加上$\eta_{ijk}=1$, 则称为K2记分.

上面确定η_{ijk}很困难, 但Heckerman et al.(1995)提出的似然等价的记分则简化很多, 称为BDe记分, 那时$\eta_{ijk}=\eta\times p(x_{ik},w_{ij}|B_0)$, 这里$p(\cdot|B_0)$代表与一个先验贝叶斯网络$B_0$相关的概率分布, 而$\eta$代表代表等价样本量的一个参数. BDe记分的一个特例为Buntine (1991)提出的BDeu记分, 那里$p(x_{ik},w_{ij}|B_0)=\frac{1}{r_iq_i}$, 即对于每个设定$\{X_i\}\cup\pi_i$, 先验网络假定为均匀分布.

结构的学习过程是学习出能最好解释数据的DAG. 这是NP-hard问题, 因为n个变量的DAG的个数是n的超指数数量级的. 如果知道变量的次序并且假定节点之间的参数独立, 事情就好办得多了, 因为可以独立地逐个学习每个节点的父节点, 而不必为无回路的约束操心. 对于每个节点最多有$\sum_{k=0}^{n-1}\binom{n}{k}=2^n-1$个可能父节点的集合. 这可以逐个集合地搜寻最高得分. 可以先从没有父节点、有1个父节点搜寻到有$n-1$个父节点的方向, 也可以反过来搜寻, 还可以从中间搜寻.

比如, 我们从最少父节点开始往上搜索, 这就需要确定增加父节点按照某种准则是否划算. 重构分析(reconstructibility analysis, RA)的标准做法是利用交互信息(mutual information, MI). X和Y之间的交互信息定义为:

$$MI(X,Y)=\sum_{x,y}p(x,y)\ln\left(\frac{p(x,y)}{p(x)p(y)}\right).$$

而X和Y之间在给定Z时的条件交互信息定义为:

$$MI(X,Y|Z)=\sum_{z}\left\{p(z)\sum_{x,y}p(x,y|z)\ln\left(\frac{p(x,y|z)}{p(x|z)p(y|z)}\right)\right\}$$

在搜索过程中, 有可能需要做一系列条件独立性检验. 根据de Campos(2006), 交互信息可以用渐近χ^2检验来评估以确定MI记分在搜索过程中是增加还是减少. 此外, 它和对数似然比检验G^2 成比例(相差$2n$因子, n是样本量), 而且和被检验模型的偏差(deviance)有关. 与此相关的条件独立性检验还包括Monte Carlo置换检验和对离散变量的经典的Pearson χ^2检验.

还有很多方法, 原理上都差不多, 但具体细节和效果依数据所反映的情况而定.

4. **未知图形及不完全的数据**

这是最困难的情况. 通常用对后验分布的渐近近似, 即BIC(Bayesian information criterion), 定义为:

$$\ln P(X|G) \approx \ln P(X|G, \hat{\Theta}_G) - \frac{\ln m}{2} \#G$$

这里, m为样本量; $\hat{\Theta}_G$为参数的最大似然估计; 而$\#G$为模型的维数. 在完全可观测的情况, 模型维数等于自由参数的数目, 在有隐变量的情况, 维数要少些. 上面第一项是似然函数, 第二项为模型复杂性的惩罚项. BIC得分和最小描述长度记分(minimum description length (MDL) score)等价: MDL= −BIC. 虽然BIC可分解成局部项, 但也很费时, 因为必须对每个节点利用EM算法来计算$\hat{\Theta}$. 也可以在EM算法的M步内做局部搜索, 这称为结构EM(structureal EM).

12.5 贝叶斯网络的数值例子及计算

12.5.1 全部变量是离散变量的情况

首先考虑例12.2的驾车数据(Driver.csv), 那里有6个二分变量: Y (Young, 是否年轻), D (Drink, 是否饮酒), A (Accident, 是否有过事故), V (Violation, 是否违规过), C (Citation, 是否收到过罚单), G (Gear, 是否自动档). 数据中用的是0,1哑元变量, 对每个变量, "是"相应于1, "否"相应于0. 实际上, 那里的图12.4是用登山算法算出来的DAG图. 这里用的程序包为`bnlearn`.[1] 下面首先采用程序包`bnlearn`中目前对于离散变量所用的几种基于记分的方法来做计算. 这里的记分包括:

- 多项对数似然记分, 代码为`loglik`, 也称为熵度量.
- AIC记分(Akaike information criterion score), 代码为`aic`.
- BIC记分(Bayesian information criterion score), 代码为`bic`.
- 贝叶斯Dirichlet等价记分的对数(logarithm of the Bayesian Dirichlet equivalent score), 代码为`bde`.
- 改进的贝叶斯Dirichlet等价记分的对数(logarithm of the modified Bayesian Dirichlet equivalent score), 代码为`mbde`.
- K2记分的对数, 代码为`k2`. 它是Dirichlet后验密度, 但不是记分等价的.

输入数据并把哑元设置为因子的代码如下:

```
w=read.csv("driver.csv")
```

[1] Radhakrishnan Nagarajan, Marco Scutari, Sophie Lebre (2013). *Bayesian Networks in R with Applications in Systems Biology*. Springer, New York. ISBN 978-1461464457.

```
for(i in 1:ncol(w))w[,i]=factor(w[,i])
names(w)=c("Y","D","A","V","C","G")#为简单用缩写变量名
```

然后用登山法对各种记分度量进行搜寻. 程序包bnlearn目前只有登山法一种记分搜索方法.

```
library("bnlearn")
(w.hc.aic=hc(w, score = "aic"))
(w.hc.bic=hc(w, score = "bic"))
(w.hc.loglik=hc(w, score = "loglik"))
(w.hc.bde=hc(w, score = "bde"))
(w.hc.mbde=hc(w, score = "mbde"))
(w.hc.k2=hc(w, score = "k2"))
```

对每一种记分都打印出学习的模型形式, 比如, 对于AIC的方法, 打印出

```
library("bnlearn")
  Bayesian network learned via Score-based methods

  model:
   [Young][Gear][Drink|Young][Accident|Young:Drink]
   [Violation|Young:Drink][Citation|Young:Drink]
  nodes:                                6
  arcs:                                 7
    undirected arcs:                    0
    directed arcs:                      7
  average markov blanket size:          2.33
  average neighbourhood size:           2.33
  average branching factor:             1.17

  learning algorithm:                   Hill-Climbing
  score:                                AIC (disc.)
  penalization coefficient:             1
  tests used in the learning procedure: 50
  optimized:                            TRUE
```

这里面包括搜索出来的模型、节点之间有向和无向联系的个数、平均Markov毯的大小、平均邻近点个数, 等等. 当然, 画出这些图来会更直观. 我们用函数graphviz.plot()来画图(图12.5, 可能需要安装程序包Rgraphviz), 这比单纯用代码plot()好看些. 代码如下:

```
library("Rgraphviz")
par(mfrow = c(2, 3))
graphviz.plot(w.hc.aic,main="Score-based hill-climbing: aic")
graphviz.plot(w.hc.bic,main="Score-based hill-climbing: bic")
graphviz.plot(w.hc.loglik,
    main="Score-based hill-climbing: loglik")
graphviz.plot(w.hc.bde,main="Score-based hill-climbing: bde")
graphviz.plot(w.hc.mbde,main="Score-based hill-climbing: mbde")
graphviz.plot(w.hc.k2,main="Score-based hill-climbing: k2")
```

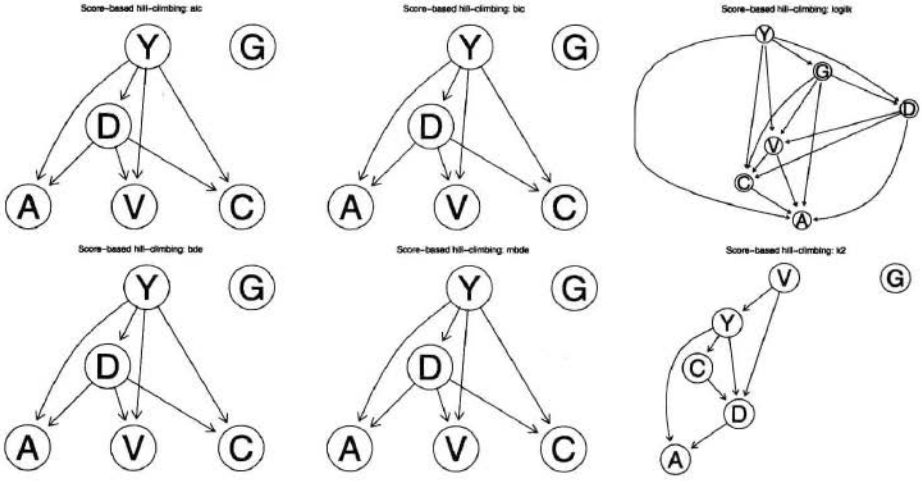


图 12.5 对例12.2使用6种记分的登山搜索方法结果图

图12.5为对例12.2使用6种记分的登山搜索方法结果图. 看得出来, 除了对数似然和k2得到的结果之外, 另外四种记分(aic,bic, bde, mbde)的结果相同, 而且比较合理.

我们也可以用基于约束的方法. 程序包bnlearn目前一共有下面几种方法(最后一种方法不是基于约束的):

- 增长收缩(grow-shrink): 最简单的基于增长收缩Markov毯(grow-shrink Markov blanket)搜索算法. 代码为gs.
- 增量关联(incremental association): 基于增量关联的Markov毯算法(IAMB), 基于两相搜索方法(向前选择及除掉假阳性). 代码为iamb.
- 快速增量关联(fast incremental association): 利用推测的向前步骤选择以减少条件独立检验数目的各种IAMB. 代码为fast.iamb.
- 交替增量关联(interleaved incremental association): 另一种IAMB, 向

前步骤选择以在Markov毯搜寻步骤(相)中减少条件独立检验数目. 代码为inter.iamb.

- 最大最小父节点子节点(max-min parents and children): 在前一次迭代所选择的任何节点子集中把观测的最小关联最大化作为邻域向前搜索. 它学习贝叶斯网络的背景结构(所有的弧为无向的, 不企图测试方向). 代码为mmpc.

下面是用这些方法对例12.2做结构搜索的代码:

```
w.gs <- gs(w,debug=T)#成为bn类型object
(w.iamb=iamb(w))
(w.fiamb=fast.iamb(w))
(w.iiamb=inter.iamb(w))
(w.mmpc=mmpc(w))
```

同样打印出各种结果, 包括非常详细的搜寻步骤. 但都没有给出有向线段. 这可以从图12.6看出. 绘制该图的代码如下:

```
par(mfrow = c(2, 3))
graphviz.plot(w.gs,main="Constraint-based algorithms: gs")
graphviz.plot(w.iamb,main="Constraint-based algorithms: iamb")
graphviz.plot(w.fiamb,main="Constraint-based algorithms: fiamb")
graphviz.plot(w.iiamb,main="Constraint-based algorithms: iiamb")
graphviz.plot(w.mmpc,main="mmpc")
```

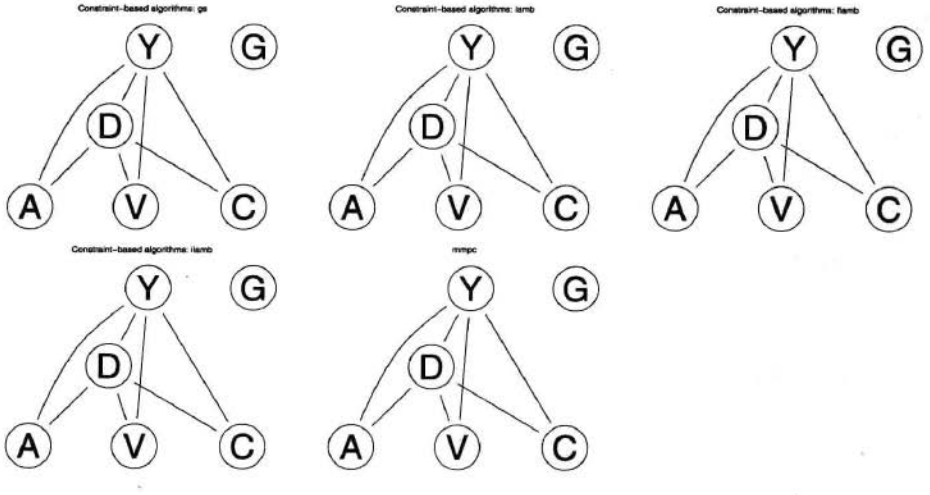


图 12.6 对例12.2使用5种搜索方法(其中4种是基于约束的)结果图

由于节点间的线段没有方向, 可以使用“白名单”和“黑名单”来人为控制方向. 白名单就是设定某些箭头的方向, 黑名单就是阻止设定某些箭头方向. 下面是代码和结果. 所生成的图和图12.4相同, 这里就不展示了.

```
WL=data.frame(from=c(rep("Y",4),rep("D",3)),
    to=c("D",rep(c("A","C","V"),2)))
w.gs <- gs(w, whitelist=WL1);graphviz.plot(w.gs1)
> WL
  from to
1    Y  D
2    Y  A
3    Y  C
4    Y  V
5    D  A
6    D  C
7    D  V
```

12.5.2 全部变量是连续变量的情况

我们通过一个例子来描述在全部变量都是连续变量的情况下网络学习的计算.

例 12.3 自我评分(score.csv) 这里是关于理解力(U)和记忆力(M)以及对各科喜恶程度的自我评分, 科目包括MATH(数学), PHYS(物理), CHEM(化学), LITE(语文), HIST(历史)等.

这里先用不加黑白名单的增长收缩法(gs)和登山法(hc)来学习网络, 选项用的都是缺省值. 然后对增长收缩法加白名单, 而对登山法既加白名单又加黑名单, 最后得到四个网络. 四个图形在图12.7中, 由于人工干预, 下面两个图是相同的. 图12.7的上面两个图未加黑白名单, 下面两个加了黑白名单.

```
library(bnlearn);U=read.csv("score.csv")
for(i in 1:ncol(U))U[,i]=as.numeric(U[,i])
U.gs0=gs(U);U.hc0=hc(U)#未加黑白名单
U.w=data.frame(from=c("M","M","LITE","U","U","U","MATH"),
    to=c("LITE","HIST","HIST","MATH","CHEM","PHYS","PHYS"))
U.gs=gs(U,whitelist=U.w)#加白名单
U.hc=hc(U,whitelist=U.w,blacklist=c("LITE","MATH"))#加黑白名单
```

```
par(mfrow = c(2, 2))
graphviz.plot(U.gs0,main="Original gs")
graphviz.plot(U.hc0,main="Original hc")
graphviz.plot(U.gs,main="Gs with whitelist")
graphviz.plot(U.hc,main="Hc with white list & blacklist")
```

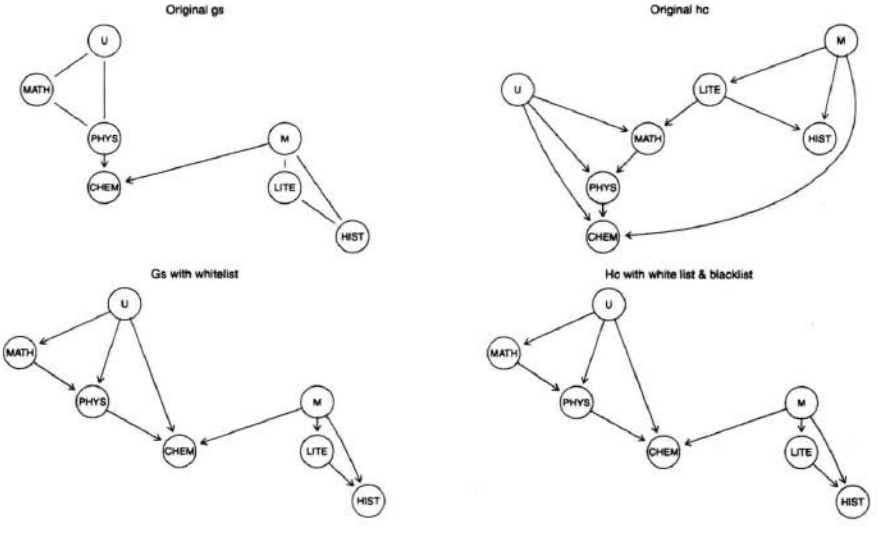


图 12.7 对例12.3使用增长收缩法和登山法的网络学习效果图

12.5.3 连续变量和离散变量混合的情况

这里我们利用可以处理变量类型混合情况的混合程序包deal[1]中自带的函数及数据ksl来应对连续和离散变量混合情况的网络学习.

deal允许贝叶斯网络既有离散变量又有连续变量, 于是, 网络中所有变量集合$X = I \cup Y$, 这里I和Y分别为离散变量和连续变量的集合. 在deal中, 为确保局部计算方法存在, 不允许离散变量有连续的父节点, 但连续变量可以有离散的父节点. 于是变量子集合$x = (i, y)$ $(x \in X, i \in I, y \in Y)$的联合概率分布可以因子化成为离散和混合两部分, 因此

$$p(x) = p(i, y) = \prod_{i \in I} p(i|\pi_i \in I) \prod_{y \in Y} p(y|\pi_y \in I \cup Y).$$

由于这里既有连续变量, 又有离散变量, 对于连续变量, 这里假定是正态分布;

[1]Susanne Gammelgaard Bottcher and Claus Dethlefsen (2013). deal: Learning Bayesian Networks with Mixed Variables. R package version 1.2-37. http://CRAN.R-project.org/package=deal.

而对于离散变量, 假定服从多项分布, 先验分布就更复杂了, 包括正态分布、逆Wishart分布、逆Gamma分布、Dirichlet分布, 等等.

deal是基于记分的搜索算法, 原则上能够关于所有可能的DAG评估网络, 但随着节点增加, 可能的DAG数目呈指数增长. 寻找高记分的DAG是需要的. deal的战略是随机重新开始吝啬搜索. 为比较不同的DAG结构, 比如结构G和G^*(这里结构在下面的概率表示中也代表该结构所对应的参数, 而x代表数据), 利用后验优势(posterior odds)

$$\frac{p(G|x)}{p(G^*|x)} = \frac{p(G,x)}{p(G^*,x)} = \frac{p(G)}{p(G^*)} \times \frac{p(x|G)}{p(x|G^*)},$$

这里, $p(G)/p(G^*)$为先验优势; $p(x|G)/p(x|G^*)$为贝叶斯因子(Bayes factor). 目前, 在deal仅有的对各种DAG设置先验分布选项是让所有DAG同等可能, 先验优势总等于1. 因此利用贝叶斯因子来比较两个不同的DAG. 在吝啬搜索中仅仅比较有一个箭头不同的模型, 或者加入、删除, 或者反向. 在这些情况中, 贝叶斯因子因为网络记分的可分解性而特别简单.

例 12.4 健康和社会特征(ksl.csv) 该数据来自Badsberg (1995), 是一项度量健康和社会特征的研究, 涉及丹麦70岁的人, 样本取自1967和1984年, 总共1083个观测值, 每个观测包含9个不同变量的观测. 其中的定性变量用哑元表示($\{0,1\}$或$\{1,2\}$). 具体变量情况见表12.2.

表 12.2 例12.4的变量说明

节点	指数	变量及说明
1	Fev	强迫吹气量: 肺功能
2	Kol	胆固醇
3	Hyp	高血压(0, 1)
4	logBMI	对数体质指数(Body Mass Index, BMI)
5	Smok	吸烟(1, 2)
6	Alc	酒精摄入程度(1, 2)
7	Work	工作(1, 2)
8	Sex	性别(1,2)
9	Year	调查年(1,2)

按照原作者的说法, 分析的目的是看哪些变量影响血压. 可以猜测某些连续变量, 诸如Fev, Kol 和BMI会有影响. 但是, 程序包deal目前还不允许离散节点有连续的父节点, 因此, 只能把血压变量Hyp当成连续变量.

下面是学习网络所用的代码. 首先读入数据, 把最后5个变量设为定性变

量, 前面4个(包括血压)为数量变量.

```
w=read.csv("ksl.csv")
for(i in c(5:9))w[,i]=factor(w[,i])
```

然后, 确保均匀的先验分布的超参数:

```
library(deal)
w.nw <- network(w) #创造一个空的框架
w.prior <- jointprior(w.nw)#默认想象样本量64(可自己设)
#下面查看默认的超参数值(也可以修改)
w.prior$jointalpha#2^5=32种 均匀离散 Dirichlet参数
w.prior$jointnu #2^5 均匀离散 正态协方差的除数
w.prior$jointrho #2^5 均匀离散 逆Wishart自由度
w.prior$jointmu #32乘4 每种离散设置4个连续(样本均值)正态均值
w.prior$jointsigma#32个4乘4样本方差对角阵 正态协方差
w.prior$jointphi  #32个4乘4样本方差对角阵 IW协方差
```

由于离散变量水平组合有$2^5 = 32$个, 上面的jointalpha(α)给出了Dirichlet的参数, 是32个同样的值. jointnu(ν)给出了正态分布协方差参数的作为尺度调整的参数, 是32个同样的值. jointrho(ρ)给出了正态分布协方差参数的逆Wishart分布的自由度, 是32个同样的值. jointmu(μ)对于32种离散水平组合, 给出了正态分布均值参数的先验分布, 每种用4个变量的样本均值代替. jointsigma(σ)对于32种离散水平组合, 给出了正态分布方差的先验分布, 每种用4×4的样本方差对角矩阵代替. jointphi(ϕ)对于32种离散水平组合, 给出了逆Wishart分布方差的先验分布, 每种用4×4的样本方差对角矩阵代替.

这里可以设"黑名单", 称为禁止名单(ban list), 但没有"白名单". 下面就是搜索代码, 输出了搜索中途和最终的结构图. 其中, 函数autosearch()给出了图12.8的左图, 这里没有用禁止名单; 函数heuristic()给出了图12.8的右图, 这里使用了禁止名单.

```
mybanlist <- matrix(c(5,5,6,6,7,7,9,8,9,8,9,8,9,8),ncol=2)
banlist(w.nw) <- mybanlist
w.nw <- learn(w.nw,w,w.prior)$nw
# 结构搜索:
w.search <- autosearch(w.nw,w,w.prior,trace=TRUE)
# 扰动最好的, 重复搜寻2次:
w.heuristic <- heuristic(w.search$nw,w,w.prior,
    restart=2,degree=10,
```

```
    trace=TRUE,trylist=w.search$trylist)
thebest2 <- w.heuristic$nw
```

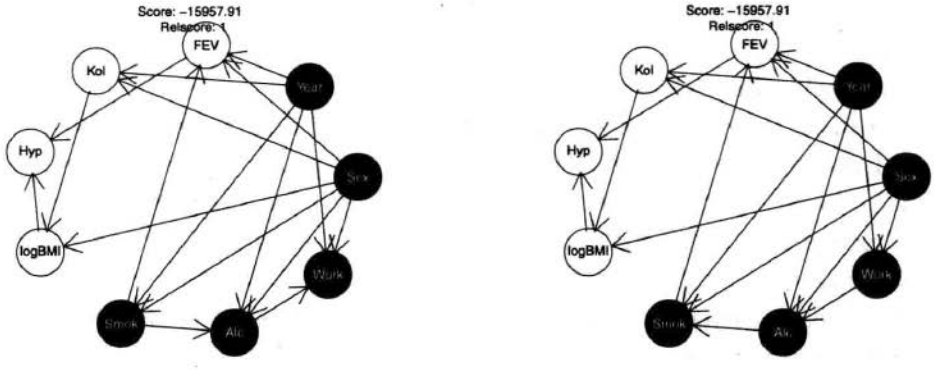


图 12.8 对例12.4的网络学习效果图

从图12.8可以看出, 禁止名单加入后, 把"吸烟导致饮酒"和"饮酒导致工作"的箭头颠倒过来了. 此外, 图中的黑底节点代表离散变量, 白底节点代表连续变量.

第十三章　多元时间序列数据

时间序列是按照一定时间间隔收集的一系列数据. 时间序列可以是一维或一元的, 即只有一个按照时间记录的变量, 比如一个气象站点获取的降水量数据; 也可以是多维的, 即变量是多维向量, 比如一个气象站点按照同样时间间隔收集的气压、气温、降水量、风力等多个指标. 研究时间序列的一个主要目的是对同样变量的未来值做预测. 对于一元时间序列来说, 这就意味着下面的假定必须成立: **这个变量的未来值能够完全由该变量的现在和过去值预测, 而不受任何其他变量的影响**. 这个假定很强, 通常不能满足, 但人们往往又有意无意地无视这个假定. 当然也有加入其他变量的模型. 通常的时间序列教科书的主要内容都是关于一元时间序列的, 这是因为其数学推导和结论研究得比较透彻, 比较容易讲解. 到了高维, 一切都不再那么清晰和漂亮, 一元时间序列的一些数学结论和公式很难推广到高维的情形.

多元时间序列在预测上是否就比一元强呢? 不一定. 时间序列, 特别是经济领域的时间序列受到大量其他因素的影响, 比如传染病(如"非典")、自然灾害(如地震、海啸)、法律政策的改变、全球的金融危机等. 这些经济领域的时间序列严重受制于那些大环境因素的变化, 而后者几乎是无法用数学方法预测的.

本章首先回顾基于时间序列的一些基本概念和计算, 然后介绍多元时间序列的单位根检验、协整检验、Granger因果检验、VAR模型、VARX模型及状态空间模型等内容. 此外, 还将介绍非线性时间序列, 包括各种门限自回归模型(SETAR, LSTAR)、神经网络模型、可加AR模型、向量误差修正模型(TVECM)等内容.

13.1　时间序列的基本概念及单变量时间序列方法的回顾

学过时间序列的读者可以不看这一节. 没有学过时间序列的读者, 也最好仅仅在不清楚某些术语及概念的定义时参阅这一节. 时间序列是统计之中数学概念最多的领域之一, 为此专门设一节来回顾.

13.1.1　时间序列的一些定义和基本概念

本小节中关于时间序列的定义和概念完全是数学上的, 为了以后叙述及读者查阅方便而汇集在这里. 时间序列领域的数学是很漂亮的, 但是, 在使用任何现有模型去拟合数据之前必须明白, 手中的数据是否近似满足必要的条件及数学假定.

1. 平稳、自协方差函数、偏自相关函数

假定$X(t_1), X(t_2), ...$是在时间$t_1, t_2, ...$上的时间序列. 所谓的严平稳时间序列(strictly stationary)要满足下列条件: 对于所有$t_1, t_2, ..., t_k$, τ, $X(t_1)$, $X(t_2)$, ..., $X(t_k)$ 与$X(t_1+\tau), X(t_2+\tau), ..., X(t_k+\tau)$的联合分布相同. 弱平稳(二阶平稳)(weakly stationary, second-order stationary)时间序列要满足: $E[X(t)] = \mu$ 及$Cov[X(t), X(t+\tau)] = \gamma(\tau)$. 该协方差仅依赖于滞后$\tau$, 与绝对时间无关. 严平稳和弱平稳的概念没有一个无条件地包含另一个的关系.

一般教科书都仅仅研究二阶平稳时间序列, 因此, 一般文献中所谓的"平稳"就意味着"二阶平稳"或"弱平稳". 一般时间序列的自协方差函数(autocovariance function, acvf)定义为:

$$\gamma(t_1, t_2) = E\{[X(t_1) - \mu(t_1)][X(t_2) - \mu(t_2)]\}.$$

平稳时间序列的自协方差函数满足

$$\gamma(\tau) = E\{[X(t) - \mu][X(t+\tau) - \mu]\} = Cov[X(t), X(t+\tau)],$$

仅仅依赖于滞后τ, 与绝对时间无关. 如果$X(t)$有均值μ, 方差σ^2, 自协方差函数(acvf) $\gamma(\tau)$及自相关函数(acf)$\rho(\tau)$, 则

$$\rho(\tau) = \gamma(\tau)/\gamma(0) = \gamma(\tau)/\sigma^2.$$

说明: $\rho(\tau) = \rho(-\tau)$; $\|\rho(\tau)\| \leqslant 1$. 一般来说, acf并不唯一识别模型. 而其偏自相关函数(partial autocorrelation function, pacf)在正态假定下定义为:

$$\phi(k) = Cor(Y_t, Y_{t+k} | Y_{t+1}, Y_{t+2}, ..., Y_{t+k-1}).$$

2. 纯随机过程

满足下面条件的序列$\{Z_t\}$称为纯随机过程(purely random process): Z_t为iid, 而且$Z_t \sim N(0, \sigma_Z^2)$.[1] 则有

$$\gamma(k) = Cov(Z_t, Z_{t+k}) = \begin{cases} \sigma_Z^2, & k = 0; \\ 0, & k = \pm 1, \pm 2, ... \end{cases}$$

$$\Rightarrow \rho(k) = \begin{cases} 1, & k = 0; \\ 0, & k = \pm 1, \pm 2, ... \end{cases}$$

纯随机过程是二阶平稳的, 也是严平稳的.

3. 差分算子和后移算子

序列的差分: 对于序列$\{X_1, X_2, ..., X_N\}$, 其一阶(向后)差分(first-order (backward) difference)为$\{y_2, y_3, ..., y_N\}$, 这里$y_t = \Delta x_t \equiv x_t - x_{t-1}$.

[1]纯随机过程不一定都定义为正态分布, 但至少是独立同分布并具有常数均值和方差, 这也称为白噪声(white noise).

二阶差分定义为:

$$\begin{aligned}\Delta^2 x_t &= \Delta(\Delta x_t) = \Delta x_t - \Delta x_{t-1} \\ &= (x_t - x_{t-1}) - (x_{t-1} - x_{t-2}) = x_t - 2x_{t-1} + x_{t-2},\end{aligned}$$

p阶差分为:

$$\Delta^p x_t = \Delta^{p-1}(\Delta x_t) = \Delta^{p-1} x_t - \Delta^{p-1} x_{t-1},$$

s滞后差分为:

$$\Delta_s x_t = x_t - x_{t-s}.$$

差分的性质还包括: $\Delta\Delta_s \equiv \Delta_s\Delta$; $\Delta c = 0$.

后移(滞后)算子(backshift (lag) operator)L(一些文献用B):

- $Ly_t \equiv y_{t-1}$, $L^s y_t = y_{t-s}$, $\Delta \equiv 1 - L$
- $Lc = c$
- $Lcy_t = cy_{t-1} = cLy_t$
- $(a_1 L^i + a_2 L^j)y_t = a_1 L^i y_t + a_2 L^j y_t = a_1 y_{t-i} + a_2 y_{t-j}$
- $L^i L^j y_t = L^{i+j} y_t = y_{t-i-j}$
- $\frac{1}{(1-aL)} y_t = \{1 + aL + a^2 L^2 + \cdots\} y_t = y_t + a y_{t-1} + a^2 y_{t-2} + \cdots$, $|a| < 1$
- $L^0 = 1$
- $L(cx_t) = cLx_t = cx_{t-1}$
- $L(x_t + y_t) = x_{t-1} + y_{t-1}$
- $(1-L)^i = \sum_{i=0}^{n} (-1)^i \binom{n}{i} L^i$
- p阶差分: $\Delta^p x_t = (1-L)^p x_t = \sum_{i=0}^{p} (-1)^i \binom{p}{i} x_{t-i}$
- k滞后差分: $\Delta_k x_t = x_t - x_{t-k} = (1 - L^k) x_t$

4. 随机游走

随机游走(random walks, 亦称随机徘徊): 假定$\{Z_t\}$为离散纯随机过程, 均值为μ, 方差为σ_Z^2, 如果

$$X_t = X_{t-1} + Z_t,$$

则过程$\{X_t\}$ 称为随机游走. 当$t = 1$, $X_1 = Z_1$, 有$X_t = \sum_{i=1}^{t} Z_i$. 因此, $E(X_t) = t\mu$, $Var(X_t) = t\sigma_Z^2$, 这意味着非平稳. 然而, 其一阶差分$\Delta X_t = X_t - X_{t-1} = Z_t$ 是一个纯随机过程, 它是平稳的.

5. 平滑

简单指数平滑(exponential smoothing)模型: 如果$\{X_i\}$ ($i = 0, \pm 1, \pm 2, \ldots$)为原始的时间序列, 那么指数平滑模型为:

$$Y_t = \alpha X_t + (1 - \alpha) Y_{t-1}, \quad 0 < \alpha < 1,$$

或者等价地表示为:

$$Y_t = \alpha \sum_{k=0}^{\infty} (1-\alpha)^k X_{t-k}.$$

还有更复杂的指数平滑模型, 包括线性趋势可加季节模型、线性趋势可乘季节模型、指数趋势可加季节模型、指数趋势可乘季节模型、减幅趋势可加季节模型、减幅趋势可乘季节模型等. 指数平滑类模型与ARMA类模型形成了当前拟合时间序列的主要方法.

除了指数平滑之外, 还有各种滤波方法, 把噪声去掉, 剩下人们想要的部分, 这也是平滑的方法. 这里对其数学细节不做赘述, 但有各种软件可以实行滤波. 此外, 还有许多非参数方法可对时间序列做平滑.

6. MA模型

移动平均过程(moving average processes): 假定均值为0, 方差为σ_Z^2的$\{Z_t\}$为离散纯随机过程, 如果

$$\begin{aligned} X_t &= Z_t + \beta_1 Z_{t-1} + \cdots + \beta_q Z_{t-q} \\ &= (1 + \beta_1 L + \cdots + \beta_q L^q) Z_t \\ &\equiv \theta(L) Z_t, \end{aligned}$$

则$\{X_t\}$称为q阶移动平均过程$MA(q)$. 考虑L为一个复变量, $\theta(L) = 0$称为该模型的特征方程. 显然

$$E(X_t) = 0, \ Var(X_t) = \sigma_Z^2 \sum_{i=0}^{q} \beta_i^2.$$

7. AR模型

自回归过程(autoregressive processes)$AR(p)$的定义为:

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + Z_t.$$

如果X_t的期望μ不等于0, 则上式可写为:

$$X_t = \alpha_0 + \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + Z_t,$$

这里$\alpha_0 = (1 - \alpha_1 - \cdots - \alpha_p)\mu$. 令$\phi(L) = (1 - \alpha_1 L - \cdots - \alpha_p L^p)$, 则该模型为$\phi(L) X_t = Z_t$. 能表明, 如果$Z_t$ 独立于$X_{t-1}, X_{t-2}, X_{t-3}, \ldots$, 那么该过程为平稳的, 当且仅当特征方程$\phi(L) = 0$ 的所有根在单位圆外.

8. **可逆性**

过程$\{X_t\}$称为可逆的(invertable), 如果随机扰动

$$Z_t = \sum_{j=0}^{\infty} \pi_j X_{t-j}$$

收敛, 这里$\sum |\pi_j| < \infty$. 也就是说, 该过程可以重新写成自回归过程的形式, 可能为有穷阶, 其系数形成收敛的和. 可逆性条件(invertibility)确保: 对于一个给定的acf, 有唯一的MA过程.

AR与MA过程互为镜像(duality). 如果序列是平稳的, 则$AR(p)$ 过程等价于收敛的$MA(\infty)$ 过程, 而且如果可逆性条件满足, 则$MA(q)$ 过程等价于收敛的$AR(\infty)$过程. 如果$MA(q)$过程的特征方程

$$\theta(L) = \beta_0 + \beta_1 L + \cdots + \beta_q L^q = 0$$

的所有根在单位圆外, 则该$MA(q)$过程为可逆的. 如果$AR(p)$过程的特征方程

$$\phi(L) = 1 - \alpha_1 L - \cdots - \alpha_p L^p = 0$$

的所有根在单位圆外, 则该$AR(p)$过程为平稳的.

9. **ARMA模型**

自回归移动平均过程$ARMA(p,q)$定义为:

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + Z_t + \beta_1 Z_{t-1} + \cdots + \beta_q Z_{t-q}$$

或者$\phi(L)X_t = \theta(L)Z_t$, 这里

$$\phi(L) = 1 - \alpha_1 L - \cdots - \alpha_p L^p,\ \theta(L) = 1 + \beta_1 L + \cdots + \beta_q L^q.$$

如果X_t的期望μ不等于0, 则上面定义可写为:

$$X_t = \alpha_0 + \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + Z_t + \beta_1 Z_{t-1} + \cdots + \beta_q Z_{t-q},$$

这里, $\alpha_0 = (1 - \alpha_1 - \cdots - \alpha_p)\mu$.

使得过程平稳的$\{\alpha_i\}$ 值满足$\phi(L) = 0$ 的根在单位圆外, 而使得过程可逆的$\{\beta_i\}$ 值满足$\theta(L) = 0$ 的根在单位圆外. ARMA比纯粹的AR或MA有较少的参数.

把ARMA写成纯MA过程: $X_t = \psi(L)Z_t$, 这里

$$\psi(L) = \theta(L)/\phi(L) = \sum \psi_i L^i.$$

把ARMA写成纯AR过程$\pi(L)X_t = Z_t$, 这里

$$\pi(L) = \phi(L)/\theta(L) = 1 - \sum_{i \geqslant 1} \pi_i L^i.$$

显然, $\pi(L)\psi(L)=1$, $\pi(L)\theta(L)=\phi(L)$, $\psi(L)\phi(L)=\theta(L)$.

10. ARIMA模型

如果X_t是不平稳的, 而

$$W_t=\Delta^d X_t=(1-L)^d X_t$$

是平稳的, 那么整合的ARMA模型$ARIMA(p,d,q)$ 为用平稳过程W_t 来替代不平稳的X_t在ARMA模型中的位置, 即

$$W_t=\alpha_1 W_{t-1}+\cdots+\alpha_p W_{t-p}+Z_t+\beta_1 Z_{t-1}+\cdots+\beta_q Z_{t-q},$$

或者

$$\phi(L)W_t=\theta(L)Z_t,$$

或者

$$\phi(L)(1-L)^d X_t=\theta(L)Z_t.$$

说明: $\phi(L)(1-L)^d$有d个根在单位圆上(如$L=1$), 意味着X_t 为非平稳的, 在单位圆上的根叫单位根(unit root). 作为特例, 随机游走(随机徘徊)$X_t=X_{t-1}+Z_t$, 可以写成$(1-L)X_t=Z_t$, 为$ARIMA(0,1,0)$.

如果时间序列X_t的d阶差分$\Delta^d X_t$是平稳过程, 则称X_t为d阶单整的, 记为$I(d)$, 也称序列X_t有d个单位根. 随机游走有一个单位根.

11. 季节模型

为了符号简洁, 本段使用下列记号:

$$\begin{aligned}
\phi_p(x) &= 1-\alpha_1 x-\alpha_2 x^2-\cdots-\alpha_p x^p,\\
\Phi_P(x) &= 1-C_1 x^s-C_2 x^{2s}-\cdots-C_P x^{Ps},\\
\theta_q(x) &= 1+\beta_1 x+\beta_2 x^2+\cdots+\beta_q x^q,\\
\Theta_Q(x) &= 1+T_1 x^s+T_2 x^{2s}+\cdots+T_Q x^{Qs}.
\end{aligned}$$

具有非平稳(通常的)阶数p,d,q, 季节阶数P,D,Q及周期s的Box-Jenkins 季节ARIMA 模型(可乘季节ARIMA 模型)为其差分序列$W_t=\Delta^d\Delta_s^D X_t$满足具有周期$s$的$ARMA(p,q)\times(P,Q)_s$模型. 一个一般的可乘季节ARIMA (SARIMA) 模型为:

$$\phi_p(L)\Phi_P(L)W_t=\theta_q(L)\Theta_Q(L)Z_t.$$

季节$SMA(Q)$模型为:

$$X_t=Z_t+T_1 Z_{t-s}+T_2 Z_{t-2s}+\cdots+T_Q Z_{t-Qs}$$

或

$$X_t=\Theta_Q(L)Z_t.$$

季节$SAR(P)$模型为:

$$X_t = C_1 X_{t-s} + C_2 X_{t-2s} + \cdots + C_P X_{t-Ps} + Z_t$$

或

$$\Phi_P(L) X_t = Z_t.$$

12. 结构时间序列模型/一元状态空间模型

结构时间序列模型(structural time series models)为针对一元时间序列的(线性高斯)状态空间模型(state-space models), 它以时间序列所分解的若干成分为基础. 那些成分序列以一组误差方差来识别. 最简单的模型为局部水平模型(local level model), 这里水平m_t也称为状态(state), 满足(状态方程或动态方程)

$$m_{t+1} = m_t + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2).$$

测量方程(观测方程)为:

$$x_t = m_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma_\epsilon^2),$$

该模型有两个参数: σ_ξ^2及σ_ϵ^2. 这是一个参数受限的$ARIMA(0,1,1)$模型.

另一个是局部线性趋势模型(local linear trend model), 有同样的测量方程, 但其动态方程中有称为斜率(slope)的随时间变化的n_t:

$$m_{t+1} = m_t + n_t + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2),$$

这里

$$n_{t+1} = n_t + \zeta_t, \quad \zeta_t \sim N(0, \sigma_\zeta^2).$$

有可能出现$\sigma_\zeta^2 = 0$的情况, 这就是局部水平模型了; 或者出现$\sigma_\xi^2 = 0$的情况, 意味着有光滑的趋势. 这是受限制的$ARIMA(0,2,2)$模型.

基本结构模型(basic structural model)为有可加季节成分的局部趋势模型. 因此, 测量方程为:

$$x_t = m_t + s_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma_\epsilon^2),$$

式中, s_t为季节成分, 其动态方程为:

$$s_{t+1} = -s_t - \cdots - s_{t-s+2} + w_t, \quad w_t \sim N(0, \sigma_w^2).$$

边界状态$\sigma_w^2 = 0$相应于确定但又任意的季节模式.

13. 多元时间序列模型

对于多元时间序列$\boldsymbol{X}_t = (x_{t1}, x_{t2}, ..., x_{tk})^\top$ $(t = 0, \pm 1, \pm 2, ...)$, 向量自回归模型$VAR(1)$为:

$$\boldsymbol{X}_t = \boldsymbol{\alpha} + \boldsymbol{\Phi}_1 \boldsymbol{X}_{t-1} + \boldsymbol{w}_t,$$

式中, $\boldsymbol{\Phi}_1$: $k \times k$; $E(\boldsymbol{w}_t \boldsymbol{w}_t^\top) = \boldsymbol{\Sigma}_w$; $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_k)^\top$ 为常数向量. 由于平稳序列总可以通过减去均值而转化为零均值序列, 因此通常假定$\boldsymbol{\alpha} = \mathbf{0}$. 向量自回归模型$VAR(p)$为:

$$\boldsymbol{X}_t = \boldsymbol{\alpha} + \sum_{j=1}^{p} \boldsymbol{\Phi}_j \boldsymbol{X}_{t-j} + \boldsymbol{w}_t, \ t = p+1, p+2, ..., n.$$

向量自回归移动平均过程$VARMA(p,q)$为:

$$\boldsymbol{X}_t = \boldsymbol{\alpha} + \sum_{j=1}^{p} \boldsymbol{\Phi}_j \boldsymbol{X}_{t-j} + \boldsymbol{w}_t + \sum_{k=1}^{q} \boldsymbol{\Theta}_k \boldsymbol{w}_{t-k},$$
$$\boldsymbol{\Phi}_p \neq \mathbf{0}, \ \boldsymbol{\Theta}_q \neq \mathbf{0}, \ \boldsymbol{\Sigma}_w > \mathbf{0}.$$

如果使用算子符号

$$\boldsymbol{\Phi}(L) = \boldsymbol{I} - \boldsymbol{\Phi}_1 L - \cdots - \boldsymbol{\Phi}_p L^p, \ \boldsymbol{\Theta}(L) = \boldsymbol{I} + \boldsymbol{\Theta}_1 L + \cdots + \boldsymbol{\Theta}_q L^q,$$

则$VARMA(p,q)$可以表示成

$$\boldsymbol{\Phi}(L) \boldsymbol{X}_t = \boldsymbol{\Theta}(L) \boldsymbol{w}_t.$$

由于种种困难, 在实际应用中, 一般都不考虑VARMA过程, 而仅仅考虑VAR过程.

带有输入项的向量自回归移动平均过程VARMAX为:

$$\boldsymbol{X}_t = \boldsymbol{\Gamma} \boldsymbol{u}_t + \sum_{j=1}^{p} \boldsymbol{\Phi}_j \boldsymbol{X}_{t-j} + \sum_{\ell=1}^{q} \boldsymbol{\Theta}_\ell \boldsymbol{w}_{t-\ell} + \boldsymbol{w}_t, \quad \boldsymbol{\Gamma} : \ k \times r,$$

式中, $\boldsymbol{u}_t$ 为$r \times 1$向量输入. 名称VARMAX中的"X"意味着它是外生向量过程.

13.1.2　常用的一元时间序列方法

1.　时间序列的组成和分解、差分及平滑

一个时间序列可能由趋势、季节、循环这三个成分中的某些或全部再加上随机成分构成. 如果能够把这些成分分解开来, 问题就可以简化. 当然, 并不是所有的序列都可以分解. 下面通过一个例子来介绍有关方法.

例 13.1　货币基数数据(AMBNS.csv)　这是在圣路易斯的美国联邦储备银行发布的货币基数.[1] 该数据是月度数据, 从1918年1月1日到2012年1月1日, 单位是10亿美元.

[1]数据来源: FRED, Federal Reserve Economic Data, Federal Reserve Bank of St. Louis: St. Louis Adjusted Monetary Base [AMBNS]; U.S. Department of Labor: Bureau of Labor Statistics; http://research.stlouisfed.org/fred2/series/AMBNS/downloaddata?cid=124; accessed Feb. 24, 2012.

去除趋势最简单的方法是一阶差分, 图13.1中的左图就是AMBNS数据全部数据曲线图, 图13.1中的右图是对其做一阶差分之后的点图, 可以看出, 一阶差分之后较早的趋势基本没有了, 但在最近几年有些波动很难消除.

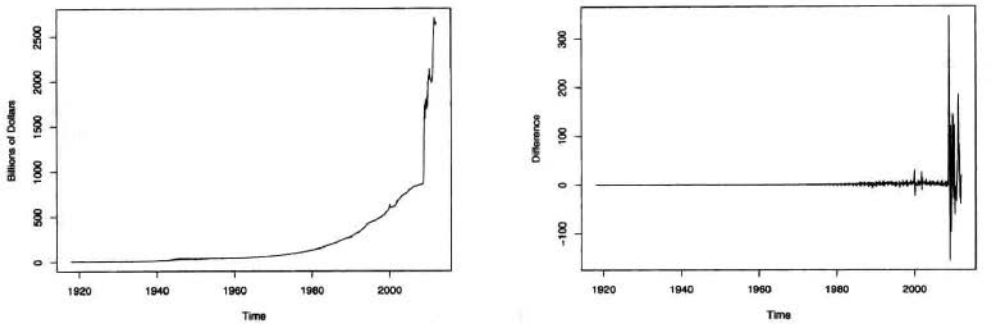


图 13.1 例13.1数据及其一阶差分图

下面是产生图13.1中两图的代码:

```
w=read.csv("AMBNS.csv")
w1=ts(w[,2],start = c(1918,1), frequency = 12)
par(mfrow=c(1,2))
ts.plot(w1,ylab="Billions of Dollars")
ts.plot(diff(w1),ylab="Difference")
```

当然, 并不是所有的趋势都能够被一阶差分消除. 对于季节性周期, 则可以用滞后等于季节长度的差分来消除. 也可以对货币基数数据做指数平滑, 其为Holt-Winters滤波的特例, 因此可用Holt-Winters滤波函数HoltWinters()来对1997年以后的数据做指数平滑, 代码为:

```
w2=ts(w1[949:1129],start = c(1997,1), frequency = 12)
b2 <- HoltWinters(w2, gamma = FALSE, beta = FALSE)
par(mfrow=c(1,2))
plot(b2) #1997以后的序列及拟合曲线图
ee2=b2$x-b2$fit[,1]
plot(ee2) #拟合误差图
```

图13.2中的两图为例13.1的1997年以后数据的指数平滑效果图, 其中左图是原数据及拟合曲线, 右图是拟合误差. 从左图看不出差别有多大, 但右图显示最后几年误差的波动很大. 相信读者很明白该数据最后几年变化较大的原因.

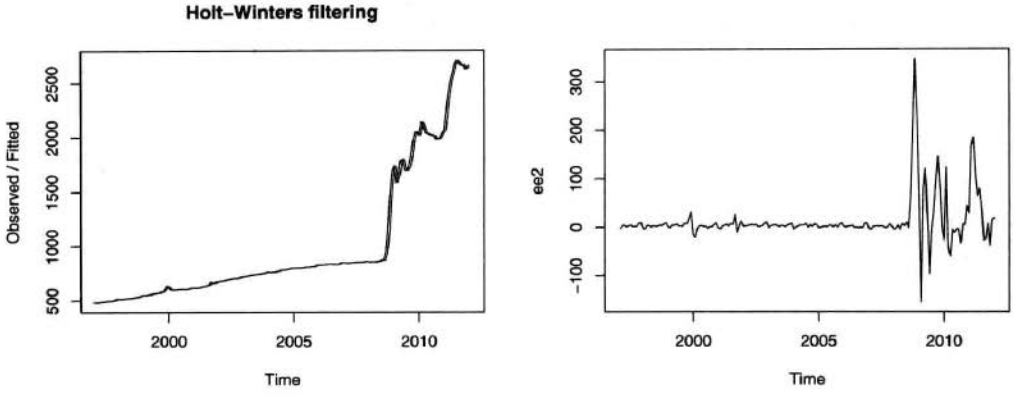


图 13.2 例13.1的1997年以后数据的指数平滑效果图

例 13.2 橙汁数据(Orange.csv) 这是1948年1月到2001年6月关于橙汁的每月观测数据[1], 为程序包Ecdat所带. 它一共有642个观测值, 3个变量: 冰冻橙汁厂家价格(priceoj)、成品的厂家价格(pricefg)、冰冻度—天数(fdd, 来自Orlando区域每日最低温度记录, 公式为$\sum_i \max(0 - T_i, 0)$, T_i为第i天的摄氏温度).

下面我们基于Loess方法对冰冻橙汁厂家价格做季节趋势分解(seasonal-trend decomposition). 使用的是程序包**stats**中的函数**stl()** (Cleveland et al., 1990). 图13.3中的左图为原始序列, 右图为画在一张图上的趋势、季节、冗余三部分.

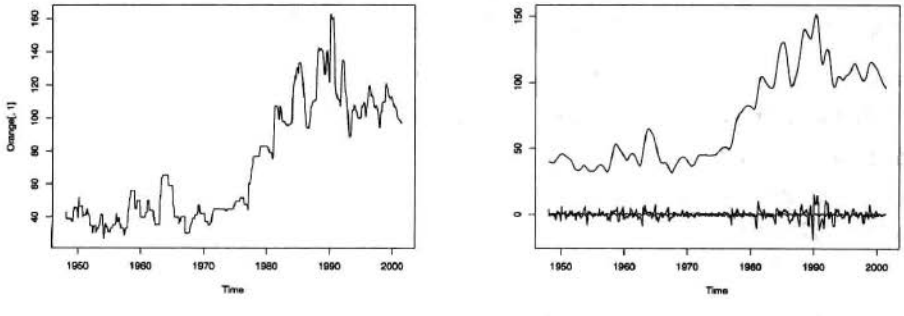


图 13.3 橙汁原始数据(左)以及数据的分解成分: 趋势、季节、冗余(右)

[1]U.S. Bureau of Labor Statistics for PPIOJ and PWFSA, National Oceanic and Atmospheric Administration (NOAA) of the U.S Department of Commerce for FDD.

下面是产生图13.3的代码:

```
data(Orange,package="Ecdat")
ts.plot(Orange[,1])#原始变量点图
a=stl(Orange[,1], "period") #对第一个变量(Orange[,1])分解
ts.plot(a$time.series[,1:3])#画出分解出来的三部分
```

2. ARMA模型和ARIMA模型

前面提到, 当时间序列均值和方差不变时, 有可能可以用平稳序列来近似, 即可以试试ARMA模型; 前面也提到, 如果在一些差分之后均值及方差不变, 可以试试ARIMA模型, 或者在差分后试试ARMA模型. 但是, 即使一个序列可以用ARMA模型, 如何选择它的阶数也是个问题. 一种方法就是用序列的acf和pacf条形图来判断其阶数, 虽然比较粗糙, 但总比没有强. 对于纯随机过程($ARMA(0,0)$), 由于观测值之间独立, acf和pacf的值很小, 它们的条形图没有什么突出的模式. 一个拟合得很好的时间序列的残差的acf和pacf条形图就应该如同纯随机过程的一样, 数值很小. 而对于一般的$ARMA(p,q)$序列阶数的确定, 则有表13.1中介绍的acf和pacf条形图直观判断法.

表 13.1 如何用pacf及acf条形图的拖尾和截尾判断ARMA模型

模型	$AR(p)$	$MA(q)$	$ARMA(p,q)$
pacf图形	第p个条后截尾	拖尾	头p个条无规律, 其后拖尾
acf图形	拖尾	第q个条后截尾	头q个条无规律, 其后拖尾

所谓"拖尾", 就是条形图以指数形式或周期形式衰减; 所谓"截尾", 就是很小而且没有什么模式. 如果acf和pacf的条形图中均没有截尾, 而且至少有一个图没有显示以指数形式或正弦形式衰减, 那么说明该序列不是平稳序列. 上述图形判别法不太准确, 只能作为参考. 稍微精确一些的做法就是对不同的p和q尝试各种ARMA模型, 比较它们的BIC等准则, 选择BIC较小的模型. 下面用一个例子来说明.

例 13.3 尼罗河数据(Nile.txt) 这是在阿斯旺所测量的1871—1970年尼罗河的年度流量. [1] 该数据可以从程序包`datasets`[2]得到.

[1] Durbin, J. and Koopman, S. J. (2001) *Time Series Analysis by State Space Methods.* Oxford University Press. http://www.ssfpack.com/DKbook.html.

[2] R Development Core Team (2011). R: A language and environment for statistical computing. *R Foundation for Statistical Computing.* Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

利用下面的语句来点出尼罗河数据的acf和pacf图(图13.4).

```
data(Nile,package="datasets")
layout(matrix(c(1,1,2,3),2,2,byrow=TRUE))
plot(Nile); acf(Nile); pacf(Nile)
```

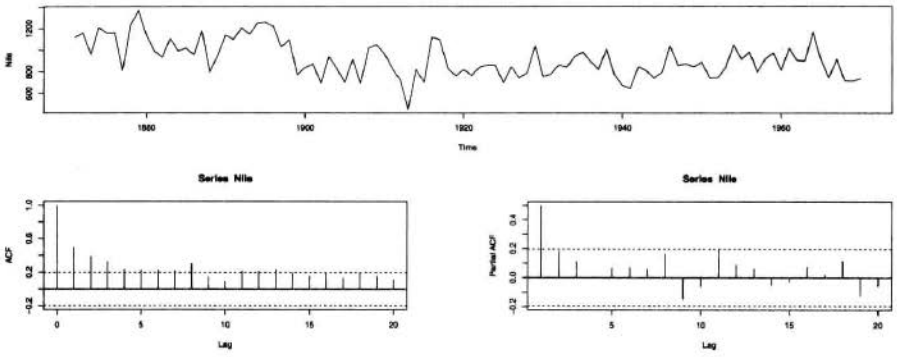


图 13.4 尼罗河数据的时间序列图(上)和acf图(下左)、pacf图(下右)

从图13.4的acf和pacf这两个条形图来看, 序列似乎是平稳的, 但很难看出阶数. 我们可以试试用程序包TSA[1]中的函数armasubsets()根据BIC准则来判断, 对于不同的p和q(这里最大的p和q均设为14) 显示出相应的BIC值, 并把结果画在图13.5中, 所用语句如下:

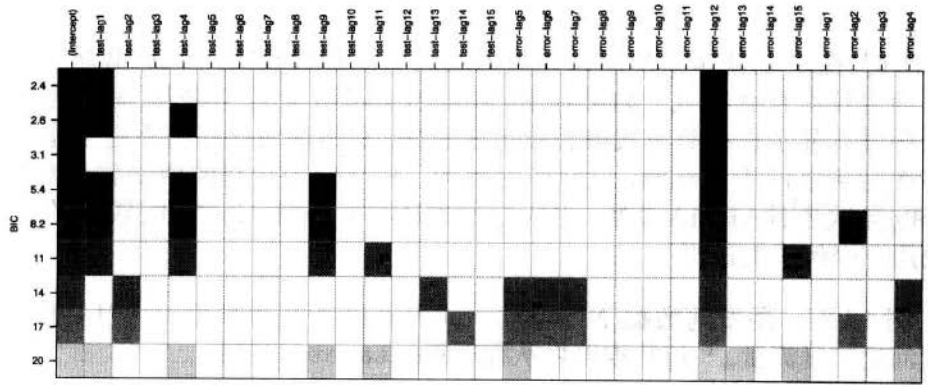


图 13.5 对于不同的p和q显示出相应的BIC值

[1]Kung-Sik Chan (2010). TSA: Time Series Analysis. R package version 0.98. http://CRAN.R-project.org/package=TSA.

```
library(TSA)
res=armasubsets(y=Nile,nar=15,nma=15,y.name="test",
    ar.method="ols")
plot(res)
```

从图13.5看出, 使得BIC最小的模型是$ARMA(1,11)$, 如果模型形式为

$$X_t = \alpha_1 X_{t-1} + \cdots + \alpha_p X_{t-p} + Z_t + \beta_1 Z_{t-1} + \cdots + \beta_q Z_{t-q},$$

那么仅有α_1及β_{11}两个系数显著. 我们用$ARMA(1,11)$模型来拟合这个数据, 语句为:

```
rn1=arima(Nile,c(1,0,11));rn1
```

打印出来的结果为:

```
Call:
arima(x = Nile, order = c(1, 0, 11))

Coefficients:
         ar1      ma1     ma2      ma3     ma4      ma5     ma6
      0.9443  -0.5546  0.0253  -0.0296  -0.2578  -0.1397  0.2750
s.e.  0.0675   0.1081  0.1332   0.1117   0.1133   0.1428  0.1095
          ma7     ma8      ma9     ma10    ma11  intercept
      -0.1890  0.3086  -0.2354  -0.3237  0.4661   930.1033
s.e.   0.1404  0.1340   0.1022   0.1511  0.1161    67.4978

sigma^2 estimated as 14980:  log likelihood=-626.93,aic=1279.86
```

从上面输出的系数估计的大小来看, β_{11}的确像图13.5反映的相对显著. 用下面的语句点出滞后为1 $\sim$ 30的Ljung-Box检验的p值(虚线为0.05水平线)、残差的acf和pacf函数图(图13.6). Ljung-Box检验的零假设为序列独立(对于某个滞后), 其p值小说明可能有相关性, 而对于不相关的观测值, 如纯随机过程, p值则应该很大. 从Ljung-Box检验及acf和pacf函数图来看, 这个拟合的残差有些像纯随机过程, 因此拟合还过得去. 语句为:

```
B=NULL;for( i in 1:30)
B=c(B,Box.test(rn1$resi, lag = i, type = "Ljung-Box")$p.value)
layout(matrix(c(1,1,2,3),2,2,byrow=TRUE))
plot(B,main="Ljung-Box tests", ylab="p-value",
    xlab="lag",pch=16,ylim=c(0,1))
abline(h=.05,lty=2)
```

```
acf(rn1$res);pacf(rn1$res)
```

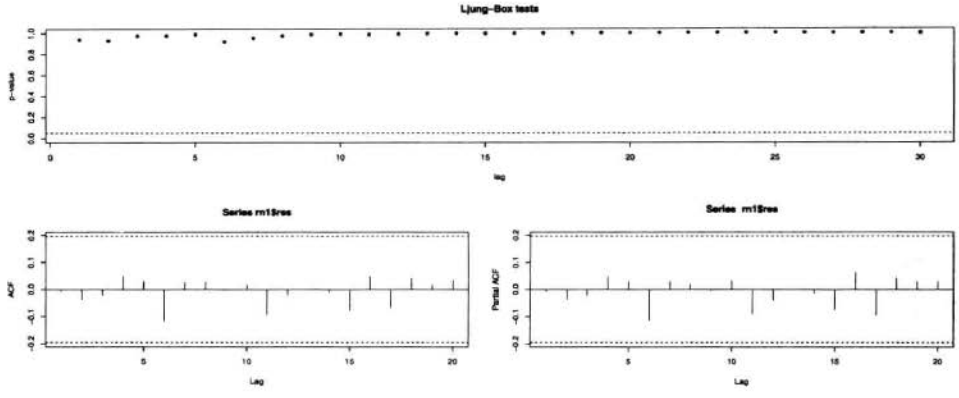


图 13.6 用ARMA拟合尼罗河数据残差的Ljung-Box检验的p值、acf图及pacf图

对于尼罗河数据, 用自动选择ARIMA参数的函数auto.arima(), 它包含在程序包forecast[1]之中. 利用下面语句做自动选择模型的计算和画图:

```
library(forecast)
data(Nile,package="datasets")
rn2=auto.arima(Nile);rn2
B=NULL;for( i in 1:30)
B=c(B,Box.test(rn2$resi, lag = i, type = "Ljung-Box")$p.value)
layout(matrix(c(1,1,2,3),2,2,byrow=TRUE))
plot(B,main="Ljung-Box tests", ylab="p-value",
    xlab="lag",pch=16,ylim=c(0,1))#Ljung-Box检验的p值图
abline(h=.05,lty=2)
acf(rn2$res);pacf(rn2$res)#acf图及pacf图
```

这里的函数auto.arima()选择了$ARIMA(1,1,1)$. 模型的确比较简单, 但相应残差的Ljung-Box检验的p值图、acf图及pacf图均不如图13.6. 这些图请读者按照上面代码自己产生.

关于自动选择模型的程序的说明: 很多软件都有自动选择模型的程序, 在R程序包中关于时间序列的ARMA模型的选择也有好几个, 但这些都有局限性. 软件不可能编得很复杂、很费时间, 因此不可能对所有的参数、所有可能

[1]Rob J. Hyndman with contributions from Slava Razbash and Drew Schmidt (2012). forecast: Forecasting functions for time series and linear models. R package version 3.20. http://CRAN.R-project.org/package=forecast.

的模型形式关于所有可能的准则做详尽的测试. 笔者相信, 而且有多次经验表明, 自动选择的模型可以作为有价值的参考, 但不见得比人工选择的要好.

下面考虑本小节一开始例13.1的货币基数数据, 截取货币基数的一部分数据(从1918年1月到2007年12月), 试试$ARIMA(0,1,8)(1,0,0)$模型, 这是从各种ARIMA模型中挑选出来的一个"较好"的猜测. 代码为(最后一行画出图13.7):

```
w=read.csv("AMBNS.csv");w1=ts(w[,2],start=c(1918,1),
    frequency=12)
w0=ts(w1[1:1080],start = c(1918,1), frequency = 12)
w2=ts(w1[949:1129],start = c(1997,1), frequency = 12)
b0=arima(w0,order=c(0,1,8),seasonal=list(order=c(1,0,0)))
bp72=predict(b0,72);plot(w2);lines(bp72$pre,lty=2)
par(mfrow=c(3,1))#下面是三个诊断点图
plot(b0$res, ylab="Residual");abline(h=5,lty=2)
abline(h=-5,lty=2)
acf(b0$res,main="ACF of residuals",ylim=c(-0.2,0.3));bb=NULL
for( i in 1:10)
bb=c(bb,Box.test(b0$res, lag = i, type = "Ljung-Box")$p.value)
plot(bb,main="Ljung-Box tests", ylab="p-value",xlab="lag",
    pch=16,ylim=c(0,1));abline(h=.05,lty=2)#Ljung-Box
```

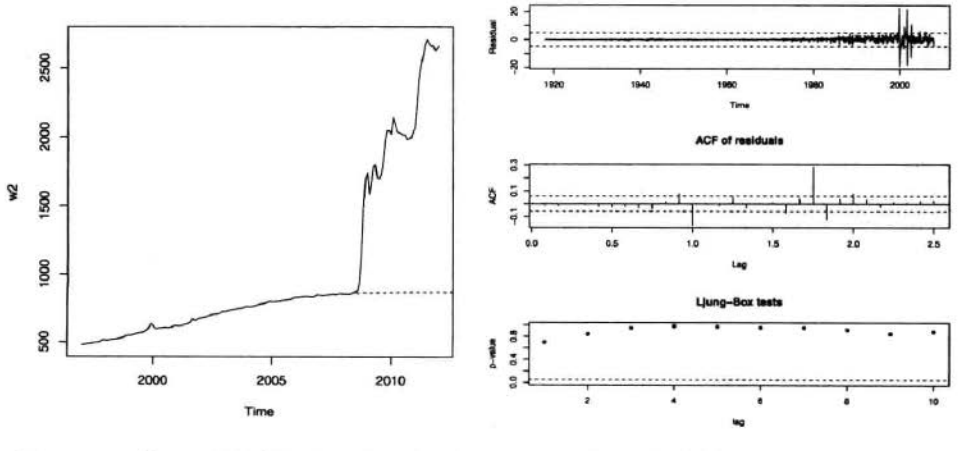


图 13.7 例13.1数据拟合后的部分序列(实线)及三年预测(虚线)图(左)和诊断图(右)

图13.7中的左图为例13.1序列放大(只取1997年后的序列)的点图(实线)及

根据上面拟合预测三年(2008—2010年)的图(虚线). 这个预测是灾难性的. 图13.7中的右三图是诊断图. 诊断图(上)为标准化残差图, 看得出来, 在2000年左右残差变化幅度很大; 诊断图(中)为残差的acf图, 表现不那么像纯随机过程; 诊断图(下)为滞后1 ~10的Ljung-Box检验p值的点图. Ljung-Box检验的p值小说明可能有相关性, 按照图中的滞后范围看不出显著的相关性, 但正如我们前面指出的, 不能拒绝零假设并不意味着零假设正确, 何况我们的滞后仅仅是有限的. 我们应该做更长时间的Ljung-Box检验. 可用下面代码求出对较长滞后($1 \sim 30$)做该检验的p值, 并产生图13.8.

```
B=NULL;for( i in 1:30)
B=c(B,Box.test(b0$resi, lag = i, type = "Ljung-Box")$p.value)
plot(B,main="Ljung-Box tests",
    ylab="p-value",xlab="lag",pch=16)
abline(h=.05,lty=2)
```

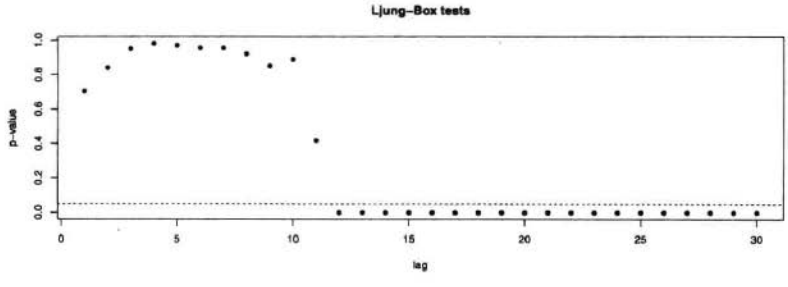


图 13.8 例13.1数据拟合后的较长滞后的Ljung-Box检验的p值(虚线为0.05水平线)

图13.8显示, 在滞后稍微大些时, p值基本上为0, 说明残差严重相关. 这说明用ARIMA模型来拟合比较"规范"的金融危机之前的数据就很不合适, 当然, 再用拟合结果来预测金融危机时候的情况(图13.7中的左图)更是极大的失策.

这里我们也试验了自动选择模型的函数`auto.arima()`. 对整个数据, 它建议用$ARIMA(4,2,5)(0,0,2)$, 但对于部分数据(从1918年1月到2007年12月), 该程序在出现警告情况下建议用$ARIMA(5,2,5)(0,0,2)$, 而且在拟合时出现了可能不收敛的问题.

这并不是一个成功的ARIMA模型拟合例子. 可能有人认为找一个更适合ARIMA模型的数据来做预测例子更好. 但是为了介绍方法和计算, 这个数据就足够了. 有什么理由不用尽管ARIMA拟合失败但却真实的数据呢? 只使用成功数据的教科书对学生会有某种误导.

3. 结构时间序列模型/一元状态空间模型

还是考虑尼罗河数据的例子, 只不过用局部水平结构时间序列来拟合, 这里测量方程为:

$$x_t = m_t + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma_\epsilon^2),$$

而动态方程(状态方程)为:

$$m_{t+1} = m_t + \xi_t, \quad \xi_t \sim N(0, \sigma_\xi^2).$$

我们将用程序包stats[1]中的函数StructTS()来得到σ_ϵ^2, σ_ξ^2, 以及$\{m_t\}$的估计. 具体代码如下:

```
data(Nile,package="datasets")
(fit <- StructTS(Nile, type = "level"))
plot(Nile)
#同期平滑得到的状态:
lines(fitted(fit), lty = 2)
#固定区间平滑得到的状态:
lines(tsSmooth(fit),lty=3,col=4)
legend("bottomleft", lty=1:3, c("Nile",
    "contemporaneous smoothing","fixed-interval smoothing"))
B=NULL;for( i in 1:30)
B=c(B,Box.test(residuals(fit),lag=i,type="Ljung-Box")$p.value)
layout(matrix(c(1,1,2,3),2,2,byrow=TRUE))
plot(B,main="Ljung-Box tests", ylab="p-value",
    xlab="lag",pch=16,ylim=c(0,1))
abline(h=.05,lty=2);acf(residuals(fit))
pacf(residuals(fit))
```

根据上面的计算, 得到$\hat{\sigma}_\epsilon^2 = 15099$及$\hat{\sigma}_\xi^2 = 1469$. 原始序列及两种平滑估计出来的$m_t$展示在图13.9中. 而残差的Ljung-Box检验以及acf和pacf条形图在图13.10的两图中, 将这两个图与用ARMA模型拟合得到的残差Ljung-Box检验和acf及pacf条形图(图13.6)比较, 可以看出结构时间序列的这个拟合并不比前面ARMA模型的拟合好.

[1] R Development Core Team (2011). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

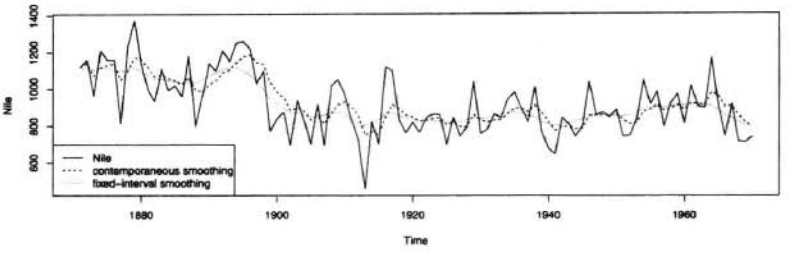


图 13.9 尼罗河时间序列及对其状态m_t的两个平滑估计

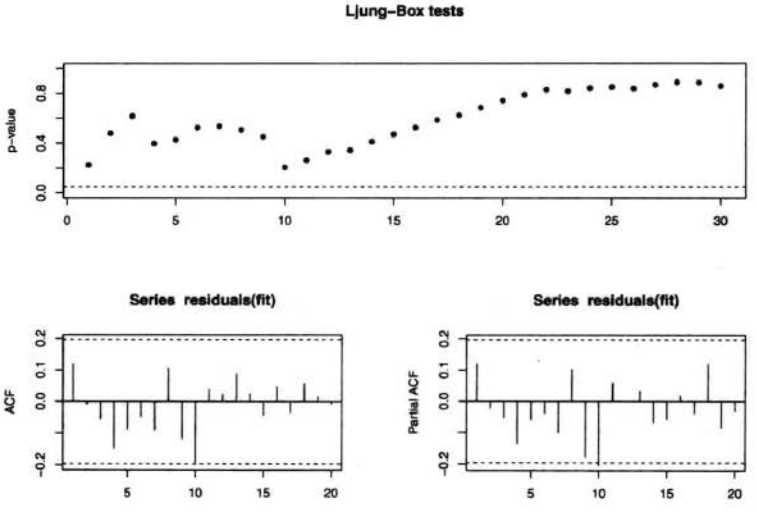


图 13.10 结构时间序列拟合例13.3残差的Ljung-Box检验p值(上)、acf及pacf条形图(下)

13.2 单位根检验、协整检验及Granger因果检验

前面说过, 用单变量时间序列本身来预测自己的未来是有风险的, 除非这个时间序列不受其他因素的影响. 因此, 人们考虑, 如果若干时间序列互相关联, 是否可以互相解释呢? 这些关系在未来是否不受其他因素的影响呢? 本节将讨论这个问题.

13.2.1 概述

本节的方法在计量经济学中是比较引人注目的. 有些人用这些方法做预测. 但是, 这些数学公式都是人们用有限的数学语言对现实问题的一种近似.

首先, 这里所有的方法都要求(实际上是假定)时间序列具有满足某些条件的线性表示, 并且满足某些分布假定, 甚至如Johansen方法中的更强的VAR表示, 没有这些假定, 所有的方法、渐近性质和结论都毫无意义. 想想看, 宇宙间有多少规律是严格线性的? 人们之所以处处使用线性模型, 或用线性模型近似, 仅仅是因为线性关系是我们或多或少可以控制的少数几个关系之一.

其次, 所有方法的结论都严重依赖于大样本性质, 这是因为独立同正态分布的条件几乎不可能满足, 也无法进行验证. 没有一位统计学家(无论他们证明了多少大样本定理)可以断定在实际中样本量达到多少才算是大样本. 而在小样本情况下, 所有这些结论可能根本不成立, 那些基于大样本定理的显著性检验也可能完全没有意义.

最后, 我们前面也说过, 一些时间序列可能是相关的, 它们在一定时期可能显示出较强的相关性. 但是, 用它们自己来预测自己的未来或者互相预测, 风险很大. 就拿宏观经济时间序列来说, 它们都受到经济危机、金融危机、政策变化、政权更迭、天灾人祸、疾病等偶然事件, 法律和规则的改变以及人们心态变化的影响, 而这些无法预料的因素根本无法加入到这几个只包含若干可测量的时间序列的模型之中, 某些短期预测往往不用时间序列也可以猜出来, 而长期预测往往是极端不可靠的. 单纯利用时间序列来做出推断, 甚至给出"政策建议", 不但是不科学的, 也是很不负责任的.

这里还会遇到利用不同的方法得到不同结论的问题. 这么多检验方法(这里只介绍几种)产生出不同的结论, 原因在于这些方法的数学模型不同以及这些模型及假定与事实不符. 不仅不同方法会导致不同结论, 同一种方法的不同角度也会导致不同的结论. 比如Engle-Granger检验, 很可能轮换回归得到的结论就不同, 不仅如此, 该方法有两个步骤, 第二个步骤肯定继承了第一个步骤的误差, 最终误差可能达到不可承受的地步.

此外, 在实际工作者使用这些方法时, 往往把"不能拒绝零假设"当成"零假设正确"或"接受零假设"的同义词, 这在逻辑上是完全错误的, 在统计上是绝对不允许的. "不能拒绝零假设"意味着拒绝的证据不足, 并不意味着零假设正确. 在拒绝零假设时, 至少给出了犯错误的大致概率, 即p值, 但任何人都无法给出"接受零假设"时犯错误的概率, 做出结论而又不给出该结论所包含的风险是极端不负责任的行为. 但实际工作者往往需要在显著性检验无法拒绝零假设时做出决策, 合乎逻辑的说法是: "在所有关于模型形式(比如某种线性表示)以及数据的**各种假定都成立的前提下**, 我没有足够的证据否定零假设, 因此**把零假设当成另一个附加的假定**."

13.2.2 单位根检验

如果一个时间序列是平稳的, 则没有任何预测价值, 因为平稳序列的均值不变, 任何预测都不会有什么有意义的结果, 所以人们只对非平稳序列感兴趣. 之所以对平稳序列进行研究, 也是因为希望可以通过差分等方式把非平稳序列转换成平稳的, 而后者是可以通过数学方式予以解释的.

如果由d次差分可以将一个非平稳序列转换成平稳的, 则称其为d阶单整的, 记为$I(d)$. 换句话说, 如果$\Delta^d X_t$为平稳的, 则称序列X_t有d个单位根. 最简单的情况为(这里带有截距)随机游走

$$X_t = c + X_{t-1} + u_t, \quad u_t \sim \text{iid}\,(0, \sigma^2),$$

这里, $u_t \sim \text{iid}\,(0, \sigma^2)$意味着$u_t$序列是独立同分布(不一定正态)的, 而且有均值0及方差σ^2. 对于随机游走, 显然, $\Delta X_t = c + u_t$为平稳过程, 即随机游走为1阶单整的($I(1)$), 它有一个单位根.

下面介绍常用的ADF单位根检验方法(Augmented Dickey-Fuller Test).

先考虑简单的$AR(1)$模型

$$X_t = \phi X_{t-1} + u_t, \tag{13.1}$$

它可以写成

$$X_t - X_{t-1} = (\phi - 1)X_{t-1} + u_t$$

或者

$$\Delta X_t = \gamma X_{t-1} + u_t, \tag{13.2}$$

式中, $\gamma = \phi - 1$. 实践中, 与其估计(13.1), 不如检验$H_0 : \gamma = 0 \Leftrightarrow H_a : \gamma \neq 0$. 在$H_0$下(意味着$\phi = 1$), 有单位根, 序列为非平稳的. 这个检验的$t$统计量不服从渐近$t$分布, 其相应的临界值由DF(Dickey-Fuller)检验的τ统计量给出. 但误差项往往出现序列相关性, 即u_t不是独立的, 这样就发展出ADF检验, 其回归方程包括那些一阶差分的滞后以确保误差项的独立同分布性质. 即有

$$\Delta X_t = \gamma X_{t-1} + \alpha_i \sum_{i=1}^{p} \Delta X_{t-i} + u_t,$$

还可以加入截距和时间趋势:

$$\Delta X_t = \beta_1 + \beta_2 t + \gamma X_{t-1} + \alpha_i \sum_{i=1}^{p} \Delta X_{t-i} + u_t. \tag{13.3}$$

基于式(13.3), ADF检验为$H_0 : \gamma = 0 \Leftrightarrow H_a : \gamma < 0$. 检验统计量为:

$$F_\gamma = \frac{\hat{\gamma}}{SE(\hat{\gamma})},$$

式中, $SE(\hat{\gamma})$为$\hat{\gamma}$的标准误差.

为了与数值计算相结合, 本节使用下面的例子.

例 13.4 芬兰数据(finland.csv) 该数据来自Johansen and Juselius (1990), 有4个时间序列变量, 它们是从1958年第二季度到1984年第三季度的货币供应量M1的对数(lrm1)、实际收入的对数(lny)、边际利率的对数(lnmr)、通货膨胀率(difp). 该数据在程序包urca[1]中提供.

下面就是对例13.4数据的ADF检验, 看其4个时间序列变量有没有单位根. 所用的检验函数是程序包urca提供的. 代码如下:

```
library(urca)
data(finland)
attach(finland)
lrm1.df=ur.df(lrm1,lags=5,type="trend");summary(lrm1.df)
lny.df=ur.df(lny,lags=5,type="trend");summary(lny.df)
lnmr.df=ur.df(lnmr,lags=5,type="trend");summary(lnmr.df)
difp.df=ur.df(difp,lags=5,type="trend");summary(difp.df)
```

ADF显著性水平为0.1的临界值为-3.13(值越小就越显著), 而这4个检验统计量的值分别为-2.2702, -1.5932, -3.1016, -2.7569, 因此没有足够的证据拒绝零假设, 也就是说, 没有证据说它们是平稳的. 因此假定它们都至少是$I(1)$的.

为了确定这些系列的单整阶数, 再对它们的差分序列做ADF单位根检验, 零假设是差分序列有单位根. 下面是代码:

```
lrm12=diff(lrm1);lrm12.df=ur.df(lrm12,lags=5,type="trend")
summary(lrm12.df)
lny2=diff(lny); lny2.df=ur.df(lny2,lags=5,type="trend")
summary(lny2.df)
lnmr2=diff(lnmr);lnmr2.df=ur.df(lnmr2,lags=5,type="trend")
summary(lnmr2.df)
difp2=diff(difp);difp2.df=ur.df(difp2,lags=5,type="trend")
summary(difp2.df)
```

这次, ADF显著性水平为0.01, 0.05及0.1的临界值分别为-3.99, -3.45, -3.13(值越小就越显著), 而上述差分的4个检验统计量分别为-4.89, -3.7416, -6.4769, -5.7403, 分别在0.01, 0.05, 0.01, 0.01的水平上显著, 这就拒绝了差分有单位根的零假设, 即它们都不太可能是$I(2)$的, 有可能是$I(1)$的.

[1] Pfaff, B. (2008) *Analysis of Integrated and Cointegrated Time Series with R.* Second Edition. Springer, New York. ISBN 0-387-27960-1.

13.2.3 协整检验

如果若干时间序列的线性组合的单整阶数小于其成分的单整阶数, 就称这些序列间存在协整(cointegration). 那些线性组合系数称为协整向量(cointegrating vector), 协整向量个数小于变量数目. 一个时间序列向量, 只有当它们是协整的时, 才有同时研究的价值. 协整意味着向量分量之间存在长期关系. 在短期中, 这些向量可能关系不那么显著, 分别由不同的动态过程所支配, 然而, 从长远来说, 协整把变量绑在一起.

只有当协整关系存在时, 研究多元时间序列才有意义, 这时可以用VAR模型或状态空间模型拟合数据. 对于多元时间序列$\boldsymbol{X}_t = (X_{t1}, X_{t2}, ..., X_{tk})^\top$ $(t = 0, \pm1, \pm2, ...)$, 带有输入项(哑元$\boldsymbol{D}_t$)的向量自回归模型$VAR(p)$为:

$$\boldsymbol{X}_t = \boldsymbol{\alpha} + \sum_{j=1}^{p} \boldsymbol{\Phi}_j \boldsymbol{X}_{t-j} + \Upsilon \boldsymbol{D}_t + \boldsymbol{u}_t, \ t = p+1, p+2, ..., n.$$

如果使用符号

$$\boldsymbol{\Gamma}_i = -(\boldsymbol{I} - \boldsymbol{\Phi}_1 - \cdots - \boldsymbol{\Phi}_i) \ (i = 1, ..., p-1); \ \boldsymbol{\Pi} = -(\boldsymbol{I} - \boldsymbol{\Phi}_1 - \cdots - \boldsymbol{\Phi}_p).$$

则$VAR(p)$可以重新表示成

$$\Delta \boldsymbol{X}_t = \boldsymbol{\Gamma}_1 \Delta \boldsymbol{X}_{t-1} + \cdots + \boldsymbol{\Gamma}_{p-1} \Delta \boldsymbol{X}_{t-p+1} + \boldsymbol{\Pi} \boldsymbol{X}_{t-p} + \Upsilon \boldsymbol{D}_t + \boldsymbol{\alpha} + \boldsymbol{u}_t, \quad (13.4)$$

称为长期向量误差修正模型(long-run vector error correction model, long-run VECM), 其中矩阵$\boldsymbol{\Pi}$包含了累积的长期影响. 这描述了序列之间的长期关系. 如果令

$$\boldsymbol{\Gamma}_i^* = -(\boldsymbol{\Phi}_{i+1} + \cdots + \boldsymbol{\Phi}_p), \ i = 1, ..., p-1,$$

则$VAR(p)$还可以表示成

$$\Delta \boldsymbol{X}_t = \boldsymbol{\Gamma}_1^* \Delta \boldsymbol{X}_{t-1} + \cdots + \boldsymbol{\Gamma}_{k-1}^* \Delta \boldsymbol{X}_{t-p+1} + \boldsymbol{\Pi} \boldsymbol{X}_{t-1} + \Upsilon \boldsymbol{D}_t + \boldsymbol{\alpha} + \boldsymbol{u}_t, \quad (13.5)$$

称为暂时的(transitory或short-run)误差修正模型.

本小节考虑三种协整检验, 一种是Engle-Granger (两步)方法, 一种是Pillips-Ouliaris方法, 最后一种为Johansen方法.

1. Engle-Granger协整检验

这个检验的思想很简单: 先用这些时间序列变量互相做普通最小二乘(OLS)回归, 再通过单位根检验它们的残差是否为$I(0)$, 如果是, 则这些变量可能存在协整关系. 我们通过例13.4来说明. 首先对这些序列互相做OLS回归:

```
lrm1=ts(lrm1,start=c(1958,2),end=c(1984,3),frequency=4)
lny=ts(lny,start=c(1958,2),end=c(1984,3),frequency=4)
```

```
lnmr=ts(lnmr,start=c(1958,2),end=c(1984,3),frequency=4)
difp=ts(difp,start=c(1958,2),end=c(1984,3),frequency=4)

flcons=window(cbind(lrm1,lny,lnmr,difp),start=c(1958,2),
    end=c(1984,3))
#各个变量轮流作因变量:
lrm1.eq=summary(lm(lrm1~lny+lnmr+difp,data=flcons));lrm1.eq
lny.eq=summary(lm(lny~lrm1+lnmr+difp,data=flcons));lny.eq
lnmr.eq=summary(lm(lnmr~lrm1+lny+difp,data=flcons));lnmr.eq
difp.eq=summary(lm(difp~lrm1+lny+lnmr,data=flcons));difp.eq
#下面是残差序列:
error.lrm1=ts(resid(lrm1.eq),start=c(1958,3),
    end=c(1984,3),frequency=4)
error.lny=ts(resid(lny.eq),start=c(1958,3),
    end=c(1984,3),frequency=4)
error.lnmr=ts(resid(lnmr.eq),start=c(1958,3),
    end=c(1984,3),frequency=4)
error.difp=ts(resid(difp.eq),start=c(1958,3),
    end=c(1984,3),frequency=4)
```

从这些回归的输出(没有在这里显示)可以看出, 有些变量之间是有些关系的, 但由于OLS的检验条件不一定符合, 不足为据, 因此必须对残差做进一步研究. 下面对这些残差做ADF单位根检验, 这里的零假设是残差序列存在单位根, 即它们之间不存在协整.

```
df.lrm1=ur.df(error.lrm1,lags=0,type="none");summary(df.lrm1)
df.lny=ur.df(error.lny,lags=0,type="none"); summary(df.lny)
df.lnmr=ur.df(error.lnmr,lags=0,type="none");summary(df.lnmr)
df.difp=ur.df(error.difp,lags=0,type="none");summary(df.difp)
```

这里, ADF检验的显著性为0.01, 0.05, 0.1的临界值分别为-2.58, -1.95, -1.62, 而这四个残差的检验统计量的值为-3.5256, -3.904, -5.1339, -8.7288, 都在0.01的水平上显著, 这就是说, 可以拒绝零假设, 得出结论说残差不存在单位根, 为$I(0)$序列, 因而这些序列之间很可能存在协整关系.

有了协整关系, 我们希望得到长期的误差修正模型(ECM), 对于例13.4, 它可以是这样的(非矩阵)形式:

$$\begin{aligned}\Delta \mathrm{lrm1}_t = {} & \beta_0 + \beta_1 \Delta \mathrm{lny}_t + \beta_2 \Delta \mathrm{lnmr}_t + \beta_3 \Delta \mathrm{difp}_t \\ & + \alpha(\mathrm{lrm1}_{t-1} - \beta_0 - \beta_1 \mathrm{lny}_{t-1} - \beta_2 \mathrm{lnmr}_{t-1} - \beta_3 \mathrm{difp}_{t-1}) + u_t.\end{aligned}$$

这些系数可以用简单线性模型估计:

```
lrm12=diff(lrm1)
lny2=diff(lny)
lnmr2=diff(lnmr)
difp2=diff(difp)
leq2=lag(error.lrm1)
ecm=summary(lm(lrm12~lny2+lnmr2+difp2+leq2))#OK
ecm$coef[,1] #得到ECM系数
```

得到上述长期ECM系数的估计: $\hat{\beta}_0 = 0.00285$, $\hat{\beta}_1 = 0.61041$, $\hat{\beta}_2 = 0.35955$, $\hat{\beta}_3 = -0.44133$, $\hat{\alpha} = -0.18900$. 当然, 这些估计是比较粗糙的. 但可以看出, 最后一项即误差项的系数估计$\hat{\alpha}$为负数, 这就是"误差修正"的含义, 描述了长期效应收敛的快慢.

2. Pillips-Ouliaris协整检验

前面的Engle-Granger协整检验利用长期方程的残差来做ADF检验, 而Pillips-Ouliaris协整检验引进两种残差检验: 方差率检验(variance ratio test)和多元迹检验. 这两个检验的零假设都是"序列不存在协整", 因此如果检验显著, 则可能存在协整.

考虑VAR模型的拟合回归方程

$$\boldsymbol{Z}_t = \hat{\boldsymbol{\Phi}}\boldsymbol{Z}_{t-1} + \hat{\boldsymbol{u}}_t,\ t = 1, 2, ..., n,$$

式中, $\boldsymbol{Z}_t = (\boldsymbol{Y}_t, \boldsymbol{X}_t)^\top$; $\boldsymbol{Y}_t$为一个序列; 而$\boldsymbol{X}_t$为时间序列向量; $\hat{\boldsymbol{\Phi}}$为估计的系数. 方差率检验统计量为:

$$\hat{P}_u = \frac{n\hat{\omega}_{11.2}}{\frac{1}{n}\sum_{t=1}^{n}\hat{e}_t},$$

式中, $\hat{e}_t$为长期回归拟合方程$Y_t = \hat{\beta}X_t + \hat{e}_t$的残差项, 而

$$\hat{\omega}_{11.2} = \hat{\omega}_{11} - \hat{\boldsymbol{\omega}}_{21}^\top \hat{\boldsymbol{\Omega}}_{22}^{-1}\hat{\boldsymbol{\omega}}_{21}.$$

而且

$$\hat{\boldsymbol{\Omega}} = \frac{1}{n}\sum_{t=1}^{n}\hat{\boldsymbol{u}}_t\hat{\boldsymbol{u}}_t^\top + \frac{1}{n}\sum_{s=1}^{l} w_{sl}\left(\sum_{t=s+1}^{n}\hat{\boldsymbol{u}}_t\hat{\boldsymbol{u}}_{t-s}^\top + \hat{\boldsymbol{u}}_{t-s}\hat{\boldsymbol{u}}_t^\top\right),$$

式中, $w_{sl} = 1 - s/(l-1)$ (窗宽$l < n$);

$$\hat{\boldsymbol{\Omega}} = \begin{pmatrix} \hat{\omega}_{11} & \hat{\boldsymbol{\omega}}_{21}^\top \\ \hat{\boldsymbol{\omega}}_{21} & \hat{\boldsymbol{\Omega}}_{22} \end{pmatrix}.$$

而多元迹统计量为:

$$\hat{P}_z = n\,\mathrm{tr}(\hat{\boldsymbol{\Omega}}\boldsymbol{M}_{zz}^{-1}),$$

这里$\boldsymbol{M}_{zz}^{-1} = \frac{1}{n}\sum_{t=1}^{n} \boldsymbol{z}_t\boldsymbol{z}_t^{\top}$.

对于例13.4的数据, 方差率检验的代码为:

```
flcons=window(cbind(lrm1,lny,lnmr,difp),
    start=c(1958,2), end=c(1984,3))
pu.test=summary(ca.po(flcons,demean="const",type="Pu"))
pu.test
```

统计量的值为26.0919, 而显著性水平为0.1的临界值为39.6949, 因此不显著(这里统计量越大越显著). 多元迹检验代码为:

```
pz.test=summary(ca.po(flcons,demean="const",type="Pz"))
pz.test
```

统计量的值为184.3248, 而显著性水平为0.01的临界值为153.4504, 因此很显著. 按照多元迹检验, 可能存在协整, 而根据方差率检验, 没有证据说明存在协整. 这两个检验的结论相互矛盾.

3. Johansen 方法

Johansen 方法(Johansen procedure 或Johansen test)允许多个协整关系, 因此比Engle - Granger检验的应用更加广泛, Engle - Granger检验是基于对单个协整关系残差的ADF单位根检验. 有两种Johansen检验: 基于迹或者特征值, 而且推断可能有区别. 回顾前面的式(13.4): k维时间序列向量的$VAR(p)$可以表示成

$$\Delta\boldsymbol{X}_t = \boldsymbol{\Gamma}_1\Delta\boldsymbol{X}_{t-1} + \cdots + \boldsymbol{\Gamma}_{p-1}\Delta\boldsymbol{X}_{t-p+1} + \boldsymbol{\Pi}\boldsymbol{X}_{t-p} + \Upsilon\boldsymbol{D}_t + \boldsymbol{\alpha} + \boldsymbol{u}_t.$$

矩阵$\boldsymbol{\Pi}$包含着关于模型的长期性质. 如果它的秩为0, 那么系统就不是协整的(假定所有在$\boldsymbol{X}_t$中的变量至少为单整$I(1)$的). 如果$\boldsymbol{\Pi}$的秩为k(满秩), 那么$\boldsymbol{X}_t$的变量为平稳的. 如果矩阵$\boldsymbol{\Pi}$有秩r, 这里$0 < r < k$, 则$\boldsymbol{\Pi}$可以分解为两个不同的$(k \times r)$矩阵$\boldsymbol{a}$和$\boldsymbol{b}$, 使得$\boldsymbol{\Pi} = \boldsymbol{a}\boldsymbol{b}^{\top}$, 即在$\boldsymbol{b}$中包含$r$个协整向量. $\boldsymbol{b}$的每一列为协整向量意味着$\boldsymbol{b}^{\top}\boldsymbol{X}_t \sim I(0)$.

Johansen方法基本上是从不受约束的VAR来估计矩阵$\boldsymbol{\Pi}$, 再看是否能够拒绝由$\boldsymbol{\Pi}$的降秩形式给出的约束. Johansen方法包含如下步骤:

- 确保时间序列向量的单整水平至少为$I(1)$;
- 选择滞后阶数;
- 选择系统的确定的成分(诸如包含截距、斜率等项的问题);
- 确定协整向量数目.

其中第四步是确定$\boldsymbol{\Pi}$的秩, 即协整向量的个数. 需要做迹检验或最大特征值检验.

最大特征值检验 为简化符号, 这里令

$$\boldsymbol{Z}_{0t}=\Delta\boldsymbol{X}_t,\ \boldsymbol{Z}_{1t}=(\Delta\boldsymbol{X}_{t-1},...,\Delta\boldsymbol{X}_{t-p+1},\boldsymbol{D}_t,1)^\top,\ \boldsymbol{Z}_{pt}=\boldsymbol{X}_{t-p},$$

再令$\boldsymbol{\Gamma}$包含参数$(\boldsymbol{\Gamma}_1,...,\boldsymbol{\Gamma}_{p-1},\Upsilon,\boldsymbol{\alpha})$, 模型(13.4)就成为

$$\boldsymbol{Z}_{0t}=\boldsymbol{\Gamma}\boldsymbol{Z}_{1t}+\boldsymbol{a}\boldsymbol{b}^\top\boldsymbol{Z}_{pt}+\boldsymbol{u}_t.$$

令

$$\boldsymbol{M}_{ij}=\frac{1}{n}\sum_{t=1}^{n}\boldsymbol{Z}_{it}\boldsymbol{Z}_{jt}^\top,\quad i,j=0,1,p.$$

记

$$\boldsymbol{S}_{ij}=\boldsymbol{M}_{ij}-\boldsymbol{M}_{i1}\boldsymbol{M}_{11}^{-1}\boldsymbol{M}_{1j},\quad i,j=0,1,p.$$

假定特征值$\hat{\lambda}_1,\hat{\lambda}_2,...,\hat{\lambda}_k$ $(\hat{\lambda}_1>\hat{\lambda}_2>\cdots>\hat{\lambda}_k>0)$为方程

$$|\lambda\boldsymbol{S}_{kk}-\boldsymbol{S}_{k0}\boldsymbol{S}_{00}^{-1}\boldsymbol{S}_{0k}|=0$$

的特征值. 要做的检验为:

$$H_0:\ \mathrm{rank}(\boldsymbol{\Pi})=r\ \Leftrightarrow\ H_a:\ \mathrm{rank}(\boldsymbol{\Pi})=r+1.$$

检验统计量为:

$$\lambda_{\max}(r,r+1)=-n\ln(1-\hat{\lambda}_{r+1}).$$

迹检验 想法是, 当增加比r个更多的λ_i, $\mathrm{tr}(\boldsymbol{\Pi})$是否会增加. $\lambda_i=0$等价于相应的迹统计量为0. 这就导致了随r递增的约束模型的似然比检验, 要做的检验为:

$$H_0:\ \mathrm{tr}(\boldsymbol{\Pi})\leqslant r\ \Leftrightarrow\ H_a:\ \mathrm{tr}(\boldsymbol{\Pi})>r.$$

检验统计量为:

$$\lambda_{\mathrm{tr}}(r)=-n\sum_{i=r+1}^{k}\ln(1-\hat{\lambda}_i).$$

使用Johansen方法对例13.4的数据进行迹检验的代码如下:

```
summary(ca.jo(data.frame(lrm1,lny,lnmr,difp),
    type="trace",ecdet="const")) #迹检验
```

程序输出下面检验结果:

```
Values of teststatistic and critical values of test:
          test 10pct  5pct  1pct
```

```
r <= 3 |  7.76  7.52  9.24 12.97
r <= 2 | 18.90 17.85 19.96 24.60
r <= 1 | 51.67 32.00 34.91 41.07
r = 0  | 95.78 49.65 53.12 60.16
```

对该输出的解读见表13.2. 按照这个结果, 有可能有$r = 2$个协整向量.

表 13.2 Johanson 迹检验结果

H_0	H_a	检验统计量	0.1临界值	0.05临界值	0.01临界值	结论
$r \leqslant 3$	$r > 3$	7.76	7.52	9.24	12.97	0.05不显著
$r \leqslant 2$	$r > 2$	18.90	17.85	19.96	24.60	0.05不显著
$r \leqslant 1$	$r > 1$	51.67	32.00	34.91	41.07	0.01水平显著
$r = 0$	$r > 0$	95.78	49.65	53.12	60.16	0.01水平显著

使用Johansen方法对例13.4的数据进行特征值检验的代码如下:

```
summary(ca.jo(data.frame(lrm1,lny,lnmr,difp),
    type="eigen",ecdet="const"))#特征值检验
```

程序输出下面检验结果:

```
Values of teststatistic and critical values of test:
          test 10pct  5pct  1pct
r <= 3 |  7.76  7.52  9.24 12.97
r <= 2 | 11.15 13.75 15.67 20.20
r <= 1 | 32.77 19.77 22.00 26.81
r = 0  | 44.11 25.56 28.14 33.24
```

对该输出的解读见表13.3. 按照这个结果, 有可能有$r = 2$个协整向量. 这两个检验的结果一致.

表 13.3 Johanson 特征值检验结果

H_0	H_a	检验统计量	0.1临界值	0.05临界值	0.01临界值	结论
$r \leqslant 3$	$r > 3$	7.76	7.52	9.24	12.97	0.05水平不显著
$r \leqslant 2$	$r > 2$	11.15	13.75	15.67	20.20	0.1水平不显著
$r \leqslant 1$	$r > 1$	32.77	19.77	22.00	26.81	0.01水平显著
$r = 0$	$r > 0$	44.11	25.56	28.14	33.24	0.01水平显著

13.2.4 Granger 因果检验

考虑两个时间序列X和Y, 由Granger(1969)提出的关于X和Y的Granger因果检验(Granger causality testing) 是用来评估X过去的观测值对于预测Y(其过去值已经建模)是否有用的. 零假设为: X过去的观测值(比如p个)对于预测Y值没用.

该检验是用p个过去的Y的值对p个过去的X的值回归而成. 统计量的分布为F分布. 下面公式对于理解其概念可能有帮助, 假定Y有自回归模型$AR(p)$(这里p可以根据AIC, BIC等准则确定):

$$Y_t = a_0 + a_1 Y_{t-1} + \cdots + a_p Y_{t-p} + \epsilon_{1t}.$$

再进行增加X的自回归

$$Y_t = a_0 + a_1 Y_{t-1} + \cdots + a_p Y_{t-p} + b_0 + b_1 X_{t-1} + \cdots + b_p X_{t-p} + \epsilon_{2t}.$$

如果系数$b_1, ..., b_p$都显著不为0或者ϵ_{2t}的方差显著小于ϵ_{1t}的方差, 则称变量X为Y的"Granger原因"(X Granger-(G)-causes Y). 这个检验虽然在原理和计算上都很简单, 但在计量经济学研究中很有用.

就我们的芬兰数据来说, 要对各个变量进行Granger 因果检验, 可以使用程序包MSBVAR[1], 为了得到其两两变量之间(选择$p = 6$)检验的F统计量值及p值, 只要输入下面代码即可:

```
library(urca);data(finland)
granger.test(finland, p=6)
```

输出为:

```
             F-statistic      p-value
lny -> lrm1    2.5550011 0.0251138779
lnmr -> lrm1   2.2508396 0.0456979173
difp -> lrm1   0.6537563 0.6869746013
lrm1 -> lny    1.6687760 0.1382613189
lnmr -> lny    2.1307762 0.0577097766
difp -> lny    0.5504900 0.7683299652
lrm1 -> lnmr   2.5841420 0.0237037209
lny -> lnmr    4.5006864 0.0005128672
difp -> lnmr   1.7440821 0.1203087148
lrm1 -> difp   1.5787451 0.1629162902
```

[1]Patrick Brandt (2012). MSBVAR: Markov-Switching, Bayesian, Vector Autoregression Models. R package version 0.7-2. http://CRAN.R-project.org/package=MSBVAR.

```
lny -> difp     2.1123702 0.0598009010
lnmr -> difp    0.8472695 0.5370558900
```

可以看出, 有些变量之间是显著的, 互相间有Granger因果关系, 比如lnmr与lrm1之间的检验p值在每个方向都小于一般人选取的显著性水平0.05.

值得注意的是, Granger 因果关系并不一定是真正的因果关系, 因为变量X和Y很可能都被另外的第三个过程影响. 由于Granger检验是为两个变量而设计的, 因此, 该检验不适用于包含许多(多于两个)变量的情况.

13.3 VAR模型、VARX模型与状态空间模型

上一节讨论了单位根、协整、Granger因果检验等问题, 主要目的是确定这些时间序列之间是否存在长期的关联关系, 但究竟是怎样的关系, 则需要用一些模型来拟合数据. 本节要介绍向量自回归模型(VAR模型)、带有输入序列(可以是多个序列)及输出序列的VARX模型以及一系列状态空间(state space)模型. 还要用这些模型拟合例13.4的数据并做出预测.

13.3.1 VAR模型的拟合与预测

这里用的程序包是vars[1], 该程序包解VAR的函数为VAR(), 其对应的模型是

$$\boldsymbol{y}_t = \boldsymbol{A}_1\boldsymbol{y}_{t-1} + \cdots + \boldsymbol{A}_p\boldsymbol{y}_{t-p} + \boldsymbol{C}\boldsymbol{D}_t + \boldsymbol{u}_t,$$

式中, $\boldsymbol{y}_t$为$K \times 1$的内生变量; $\boldsymbol{u}_t$为同样维数的扰动项. 系数矩阵$\boldsymbol{A}_1, \cdots, \boldsymbol{A}_p$为$K \times K$的. 在函数选项中可以加常数项、趋势项、中心化的季节变量、外生变量($\boldsymbol{C}\boldsymbol{D}_t$). 函数缺省值有常数项, 而没有季节和外生变量. 对于例13.4数据, 代码为:

```
library(vars)
data(finland,package="urca") #finland数据
VAR(finland, p = 2, type = "none")
VAR(finland, p = 2, type = "const")
VAR(finland, p = 2, type = "trend")
VAR(finland, p = 2, type = "both")
```

下面只给出代码中type = "none"的输出:

[1]Bernhard Pfaff (2008). VAR, SVAR and SVEC Models: Implementation Within R Package vars. *Journal of Statistical Software* 27(4). URL http://www.jstatsoft.org/v27/i04/.

```
VAR Estimation Results:
=======================

Estimated coefficients for equation lrm1:
=========================================
Call:
lrm1=lrm1.l1+lny.l1+lnmr.l1+difp.l1+lrm1.l2+lny.l2+lnmr.l2
    +difp.l2

lrm1.l1 lny.l1 lnmr.l1 difp.l1 lrm1.l2 lny.l2 lnmr.l2 difp.l2
0.5082 -0.1629 -0.0867 -0.3503  0.3748 0.2647 -0.2273  0.0701

Estimated coefficients for equation lny:
========================================
Call:
lny=lrm1.l1+lny.l1+lnmr.l1+difp.l1+lrm1.l2+lny.l2+lnmr.l2
    +difp.l2

lrm1.l1 lny.l1 lnmr.l1 difp.l1 lrm1.l2 lny.l2 lnmr.l2 difp.l2
-0.2407 0.5174 -0.0421 -0.3785  0.2189 0.5090 -0.1479  0.2284

Estimated coefficients for equation lnmr:
=========================================
Call:
lnmr=lrm1.l1+lny.l1+lnmr.l1+difp.l1+lrm1.l2+lny.l2+lnmr.l2
    +difp.l2

lrm1.l1 lny.l1 lnmr.l1 difp.l1 lrm1.l2 lny.l2 lnmr.l2 difp.l2
-0.2654 0.0828  0.8118  0.1347  0.2929 -0.0916 -0.2326 0.0701

Estimated coefficients for equation difp:
=========================================
Call:
difp=lrm1.l1+lny.l1+lnmr.l1+difp.l1+lrm1.l2+lny.l2+lnmr.l2
    +difp.l2
```

```
lrm1.l1 lny.l1 lnmr.l1 difp.l1 lrm1.l2 lny.l2 lnmr.l2 difp.l2
 0.0363 -0.0282 0.0102 0.2098 -0.0193  0.0176 -0.0183  0.3422
```

可以看出, 输出展示逐个序列的拟合形式, 也就是没有给出矩阵形式. 由于选的滞后数目为p=2, 因此每个变量有两个滞后的系数. 比如上面最后一行代表:

$$\begin{aligned}\mathrm{difp}_t &= 0.0363\,\mathrm{lrm1}_{t-1} - 0.0282\,\mathrm{lny}_{t-1} + 0.0102\,\mathrm{lnmr}_{t-1} + 0.2098\,\mathrm{difp}_{t-1}\\ &\quad - 0.0193\,\mathrm{lrm1}_{t-2} + 0.0176\,\mathrm{lny}_{t-2} - 0.0183\,\mathrm{lnmr}_{t-2} + 0.3422\,\mathrm{difp}_{t-2}.\end{aligned}$$

如果要对VAR(finland,p=2,type="both")拟合的模型做未来的预测(设定置信度为0.95以求出置信带), 则可以用下面代码:

```
z=VAR(finland,p=2,type="both")
zp=predict(z,n.ahead =8,ci = 0.95)
plot(zp)
```

用代码zp$fcst可输出所有的预测值, 包括95%置信带. 还可输出预测的图形(图13.11).

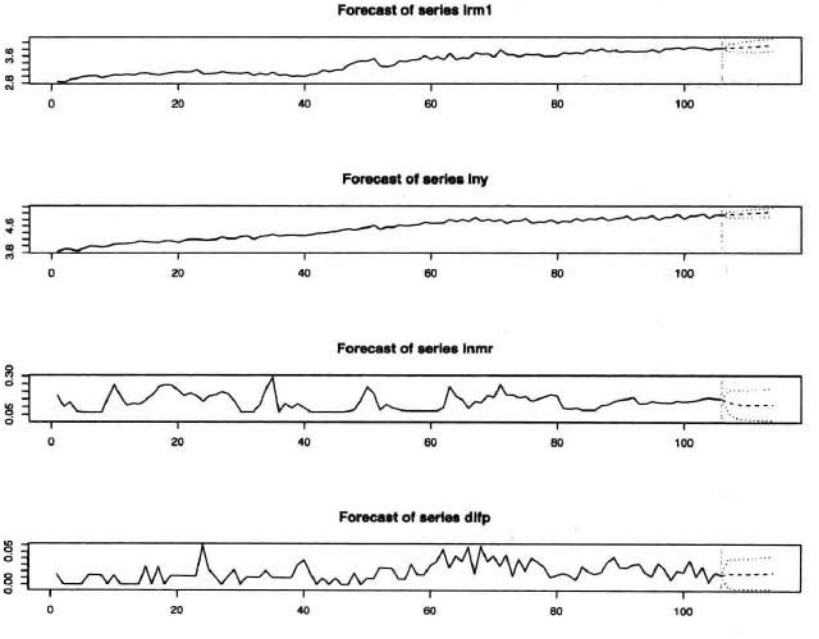


图 13.11 VAR拟合模型对未来的预测

图13.11中各个图右侧的虚线部分为预测值, 上下虚线之间的区域为95%置信带.

13.3.2 VARX模型的拟合与预测

这里用的程序包是dse.[1] 使用该程序包可以实现VARX模型及状态空间模型的拟合. 使用程序包dse的输出是矩阵形式, 相应的带有输入序列的向量自回归移动平均模型(VARMAX)可以用下式表达

$$\boldsymbol{A}(L)\boldsymbol{Y}_t = \boldsymbol{B}(L)\boldsymbol{e}_t + \boldsymbol{C}(L)\boldsymbol{X}_t, \tag{13.6}$$

式中, $\boldsymbol{Y}_t$是k维输出变量(处于因变量的地位), 也就是说, 有k个时间序列作为因变量; $\boldsymbol{X}_t$为m维输入变量(自变量), 由m个时间序列组成; $\boldsymbol{e}_t$为k维不可观测的扰动过程; $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$为适当维的后移滞后算子$L$的矩阵. VARX模型为VARMAX模型的特例, 其$\boldsymbol{B}(L) = I$. 下面用例13.4的芬兰数据来说明对这些系数矩阵的估计.

使用程序包dse的第一步是把数据变成该程序包的数据格式. 代码如下:

```
data(finland,package="urca") #finland数据
library(dse);attach(finland)
#把第一个变量(M1)作为输出, 另外三个作为输入:
fld <- TSdata(input= finland[,2:4],output= finland[, 1])
#把数据标为时间序列:
fld <-tframed(fld,list(start=c(1958,2), frequency=4))
#把变量名标上(说明:旧版本无法取代默认名字:Series 1,Series 2,...):
seriesNamesInput(fld) <- c("lny","lnmr","difp")
seriesNamesOutput(fld) <- "lrm1"
```

利用最小二乘估计拟合VARX模型:

```
fld.ls <- estVARXls(fld,max.lag=2)#默认的max.lag=6
print(fld.ls)
stability(fld.ls)
rr=checkResiduals(fld.ls)
par(mfrow=c(1,2));acf(rr$re);pacf(rr$re)
```

输出的拟合矩阵为:

```
neg. log likelihood= -81.20899
```

[1]Gilbert, P. D. (2006 or later) Brief User's Guide: Dynamic Systems Estimation distributed with the dse package. http://cran.r-project.org/web/packages/dse/vignettes/dse-guide.pdf.

```
A(L) =
1-0.733L1-0.173L2

B(L) =
1

C(L) =
0.487-0.410L1     0.245-0.329L1     -0.770+0.091L1
```

这意味着拟合的式(13.6)为:

$$(1-0.73L-0.17L^2)\mathrm{lrm1}$$
$$=(0.49-0.41L)\mathrm{lny}+(0.25-0.33L)\mathrm{lnmr}+(-0.77+0.09L)\mathrm{difp}$$

或者

$$\begin{aligned}&\mathrm{lrm1}_t-0.73\,\mathrm{lrm1}_{t-1}-0.17\,\mathrm{lrm1}_{t-2}\\=\;&0.49\,\mathrm{lny}_t-0.41\,\mathrm{lny}_{t-1}+0.25\,\mathrm{lnmr}_t\\&-0.33\,\mathrm{lnmr}_{t-1}-0.77\,\mathrm{difp}_t+0.09\,\mathrm{difp}_{t-1}.\end{aligned}$$

图13.12给出了用VARX模型拟合例13.4数据所得残差的acf图和pacf图.

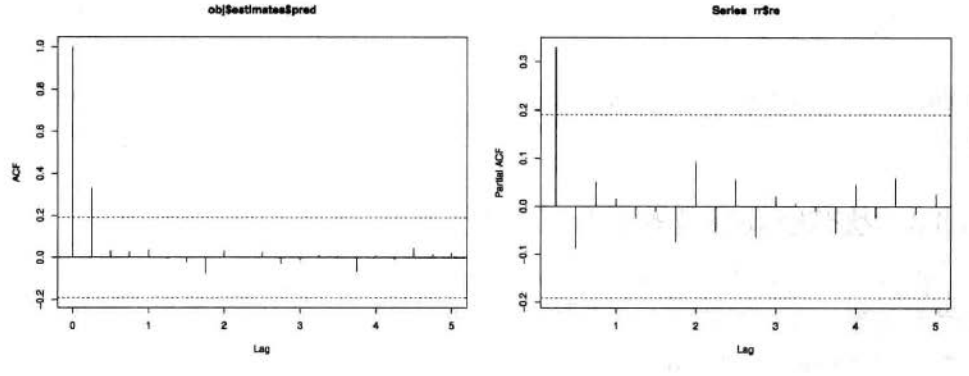


图 13.12 用VARX模型拟合例13.4数据所得残差的acf图(左)和pacf图(右)

如果要利用到1982年第二季度为止的数据建模, 并利用原来全部数据来预测1982年第三季度到1984年第三季度的输出(M1), 则可以用下面语句(包括画图):

```
fld.ls2 <- estVARXls(window(fld,end=c(1982,2)),max.lag=2)
```

```
f.p=forecast(fld.ls2, conditioning.inputs=fld$input)
tfplot(f.p)
```

得到从1982年第三季度到1984年第三季度的输出(M1)预测值(下面的输出)以及利用模型对其他时间输出的预测值.

```
         Qtr1     Qtr2     Qtr3     Qtr4
1982                      3.798115 3.836499
1983 3.758110 3.799563 3.803339 3.819459
1984 3.770789 3.817838 3.806252
```

注意, 软件输出中对于建模中没有包括的部分的预测叫forecast, 而对于训练集本身的预测叫pred(prediction). 预测的部分在图13.13中用虚线展示(实线是真实数值).

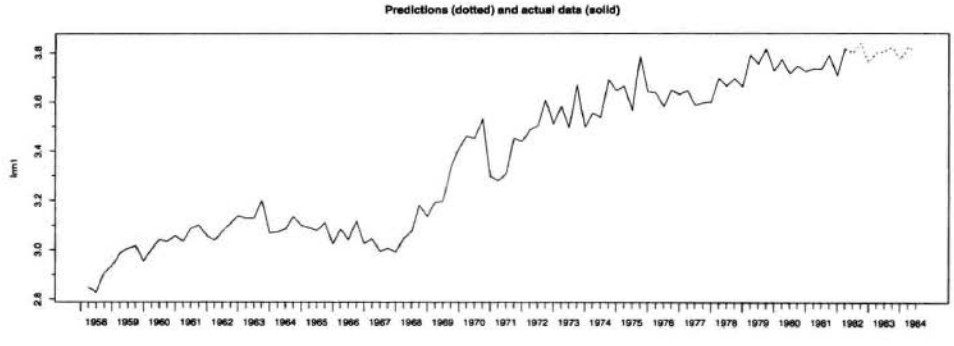


图 13.13 用基于到1982年第二季度为止的数据估计的VARX模型预测以后9个季度的输出

13.3.3 状态空间模型的拟合与预测

状态空间模型是另一种多元时间序列模型, 是VARX之外的另一种包含输入序列及输出序列的模型. 一个独立于时间的线性状态空间(state space)的创新形式(innovations form)为:

$$\boldsymbol{Z}_t = \boldsymbol{F}\boldsymbol{Z}_{t-1} + \boldsymbol{G}\boldsymbol{X}_t + \boldsymbol{K}\boldsymbol{e}_{t-1},$$
$$\boldsymbol{Y}_t = \boldsymbol{H}\boldsymbol{Z}_t + \boldsymbol{e}_t,$$

式中, $\boldsymbol{Z}_t$为不可观测的n维状态向量; $\boldsymbol{F}$为状态转移矩阵; $\boldsymbol{G}$为输入矩阵; $\boldsymbol{H}$为输出矩阵; $\boldsymbol{K}$为Kalman收益. 第一个方程通常称为状态转移方程(state transition equation), 第二个称为测量方程(measurement equation). 这些概念和结

构方程模型的一些概念有些类似, 这里$\boldsymbol{Z}_t$是隐变量, 而$\boldsymbol{Y}_t$和$\boldsymbol{X}_t$是可观测变量. 另一个更一般的模型为非创新形式(non-innovations form):

$$\boldsymbol{Z}_t = \boldsymbol{F}\boldsymbol{Z}_{t-1} + \boldsymbol{G}\boldsymbol{X}_t + \boldsymbol{Q}\boldsymbol{n}_t,$$
$$\boldsymbol{Y}_t = \boldsymbol{H}\boldsymbol{Z}_t + \boldsymbol{R}\boldsymbol{e}_t,$$

式中, $\boldsymbol{n}_t$为系统噪声; $\boldsymbol{Q}$为系统噪声矩阵; $\boldsymbol{R}$为输出(测量)噪声矩阵.

下面用程序包dse中的状态空间模型拟合例13.4数据. 和前面一样, 仍然把变量lrm1作为输出变量$\boldsymbol{Y}_t$, 而把其余三个作为输入变量$\boldsymbol{X}_t$. 使用如下代码(estSSfromVARX()是用VARX模型来估计状态空间模型的方法):

```
fld.ss <- estSSfromVARX(fld,max.lag=3)
print(fld.ss)
stability(fld.ss)
rs=checkResiduals(fld.ss)
par(mfrow=c(1,2));acf(rs$re);pacf(rs$re)
```

输出的拟合矩阵为:

```
neg. log likelihood= -58.54483

 F =
     [,1] [,2]    [,3]
[1,]    0    0 -0.0678
[2,]    1    0  0.2391
[3,]    0    1  0.7257

 G =
        [,1]   [,2]    [,3]
[1,]  0.0721 -0.162  0.3296
[2,] -0.4238 -0.191  0.0517
[3,]  0.4371  0.204 -0.9037

 H =
     [,1] [,2] [,3]
[1,]    0    0    1

 K =
        [,1]
[1,] -0.0678
```

```
[2,]  0.2391
[3,]  0.7257
```

这给出了状态空间模型中的矩阵的估计:

$$\hat{\boldsymbol{F}} = \begin{pmatrix} 0 & 0 & -0.068 \\ 1 & 0 & 0.239 \\ 0 & 1 & 0.726 \end{pmatrix}, \quad \hat{\boldsymbol{G}} = \begin{pmatrix} 0.072 & -0.162 & 0.330 \\ -0.424 & -0.191 & 0.052 \\ 0.437 & 0.204 & -0.904 \end{pmatrix},$$

$$\hat{\boldsymbol{H}} = (0,0,1), \quad \hat{\boldsymbol{K}} = \begin{pmatrix} -0.068 \\ 0.239 \\ 0.726 \end{pmatrix}.$$

如果用基于到1982年第二季度为止的数据估计的状态空间模型预测以后9个季度的输出, 可用下面语句:

```
fld.ss <- estSSfromVARX(window(fld,end=c(1982,2)),
    fld,max.lag=3)
f.s.p=forecast(fld.ss,conditioning.inputs=fld$input)
tfplot(f.s.p)
```

图13.14的虚线部分为预测的值, 这个图和VARX模型预测图(图13.13)不完全一样.

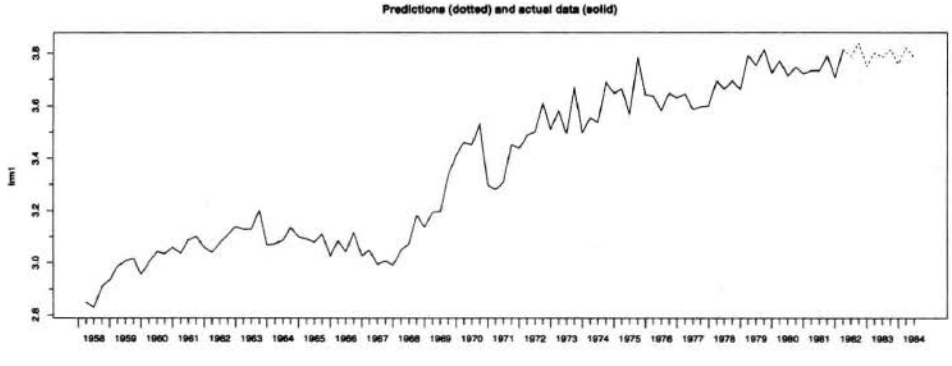


图 13.14 用状态空间模型预测1982年第二季度以后9个季度的输出

13.3.4 模型的比较

程序包dse有一个自动在各种状态空间模型中选择"最好"模型的函数estBlackBox()(黑匣子). 下面将要利用这个黑匣子模型来拟合数据, 并和前面的VARX及状态空间模型进行比较. 这里需要比较的模型包括VARX模

型(用estVARXls()函数)、状态空间模型(用estSSfromVARX()函数), 及黑匣子模型(用estBlackBox()函数). 为了得到黑匣子模型的估计, 可以运行下面的代码:

```
fld.bb <- estBlackBox(fld, max.lag=6)
print(fld.bb)
stability(fld.bb)
rb=checkResiduals(fld.bb)
par(mfrow=c(1,2));acf(rb$re);pacf(rb$re)
```

结果(在lag=6之内)选择了阶数为2的状态空间模型, 输出的估计矩阵为:

```
neg. log likelihood= -81.20899

 F =
       [,1]   [,2]
[1,]  0.542 -0.682
[2,] -0.406  0.192

 G =
       [,1]   [,2]    [,3]
[1,] -0.471 -0.253  0.6713
[2,] -0.456 -0.396 -0.0366

 H =
     [,1] [,2]
[1,] -1.14 0.11

 K =
       [,1]
[1,] -0.605
[2,]  0.387
```

即

$$\hat{\boldsymbol{F}} = \begin{pmatrix} 0.542 & -0.682 \\ -0.406 & 0.192 \end{pmatrix}, \quad \hat{\boldsymbol{G}} = \begin{pmatrix} -0.471 & -0.253 & 0.6713 \\ -0.456 & -0.396 & -0.0366 \end{pmatrix},$$

$$\hat{\boldsymbol{H}} = (-1.14, 0.11), \quad \hat{\boldsymbol{K}} = \begin{pmatrix} -0.605 \\ 0.387 \end{pmatrix}.$$

用下面语句对以fld.ls, fld.ss, fld.bb为结果的三个模型做比较:

```
informationTests(fld.ls, fld.ss,fld.bb)
tfplot(fld.ls, fld.ss,fld.bb,start=c(1960,1))
```

得到下面关于三个模型(按照次序1,2,3)各种指标的输出(表13.4)以及原输出序列及三个模型向前一步预测值的图(图13.15).

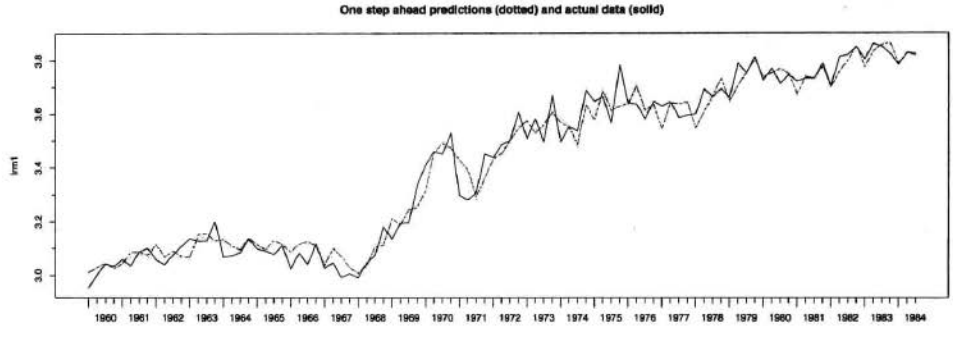


图 13.15 例13.4数据原输出序列及三个模型向前一步预测值

表 13.4 模型比较的各种指标

			based on no.of parameters					based on theoretical parameter dim.				
	PORT	−ln(L)	AIC	BIC	GVC	RICE	FPE	AIC	BIC	GVC	RICE	FPE
1	13.6	−81.2	−146.4	−125.1	−145.8	−145.1	−146.4					
2	14.2	−58.5	−87.1	−47.1	−84.7	−81.8	−86.9	−87.1	−47.1	−84.7	−81.8	−86.9
3	13.6	−81.2	−134.4	−97.1	−132.4	−129.9	−134.3	−142.4	−115.8	−141.4	−140.3	−142.4

表13.4中各种指标的英文原文为:

- PORT: Portmanteau test
- AIC: neg. Akaike Information Criterion
- GVC: Generalized Cross Validation
- FPE: Final Prediction Error
- −ln(L): neg. log likelihood
- BIC: neg. Bayes Information Criterion
- RICE: Rice Criterion

从三个模型基于参数个数(输出中标以"based on no.of parameters"的一栏) 的一些准则看, 用函数estVARXls()得到的VARX模型似乎是最好的.

13.4 非线性时间序列

13.4.1 引言

本节介绍如何从图形区别出一个线性模型与一个非线性模型, 但对这里涉及的众多知识不做详尽解释, 希望有兴趣的读者参阅动力系统、混沌理论、差分方程、非线性时间序列等方面的文献.

先看两个数据例子, 一个是模拟的$ARIMA(1,1,1)$时间序列数据; 另一个是程序包tsDyn[1]所带的lynx数据.

例 13.5 人造$ARIMA(1,1,1)$数据 该数据是用下面语句模拟而成的, 为$ARIMA(1,1,1)$模型, 有参数$\alpha = 0.2, \beta = -0.1$. 这是一个线性时间序列, 是为了与非线性时间序列进行比较而产生的.

```
library(TSA);set.seed(0)
y=arima.sim(list(order=c(1,1,1),ar=0.2,ma=-0.1),n=120)
```

例 13.6 猞猁数据(lynx) 猞猁数据是1821—1934年在加拿大每年捕获的猞猁数目(参见Brockwell & Davis, 1991; Campbell & Walker, 1977).

为了比较这两个数据, 看原来的序列图(图13.16).

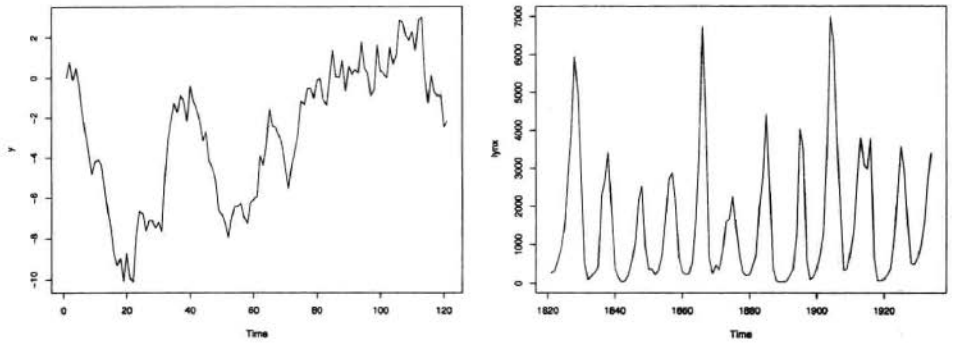


图 13.16 例13.5数据(左)和例13.6数据(右)的时间序列图

从图13.16可以看出, 猞猁数据出现震荡, 但又不是周期性震荡, 这可能是一个非线性随机动态系统(nonlinear stochastic dynamics).图13.16是用下面代码产生的:

[1]Antonio, Fabio Di Narzo, Jose Luis Aznarte and Matthieu Stigler (2009). tsDyn: Time series analysis based on dynamical systems theory. Matthieu Stigler (2010). tsDyn: Threshold cointegration: overview and implementation in R.

```
library(tsDyn);x=log10(lynx);plot(y);plot(lynx)
```

通过观察两个序列分别关于不同滞后的相图(phase portrait plot), 即对$m = 1, 3$, x_t对x_{t-m}的散点图(图13.17), 也可以发现它们有很大区别. 猞猁数据的相图(图13.17的右侧两图)与人造线性时间序列数据相图(图13.17的左侧两图)很不一样, 左边两图杂乱无序地集中在对角线方向, 而右边两图显然在循环转圈. 这至少说明右边的图反映了一些有别于线性时间序列的特征, 也就是非线性特征.

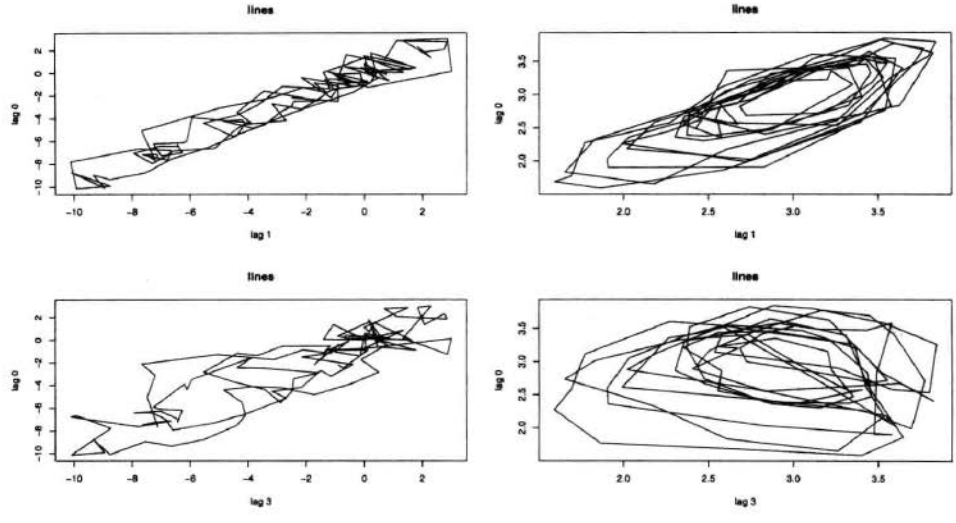


图 13.17　例13.5数据(左)和例13.6数据(右)关于不同滞后的相图

产生图13.17的代码为:

```
library(tsDyn);
par(mfrow=c(2,2))
autopairs(y,lag=1,type="lines");autopairs(x,lag=1,type="lines")
autopairs(y,lag=3,type="lines");autopairs(x,lag=3,type="lines")
```

我们还可以用局部线性AR时间序列模型来预测这两个序列未来的值, 并对它们点出相图(图13.18), 代码如下:

```
y.new=llar.predict(y,n.ahead=40,m=3,eps=2,onvoid="enlarge",r=5)
lag.plot(y.new,labels=FALSE,lag=4)
x.new=llar.predict(log(lynx),n.ahead=40,m=3,eps=2,
    onvoid="enlarge",r=5)
lag.plot(x.new, labels=FALSE,lag=4)
```

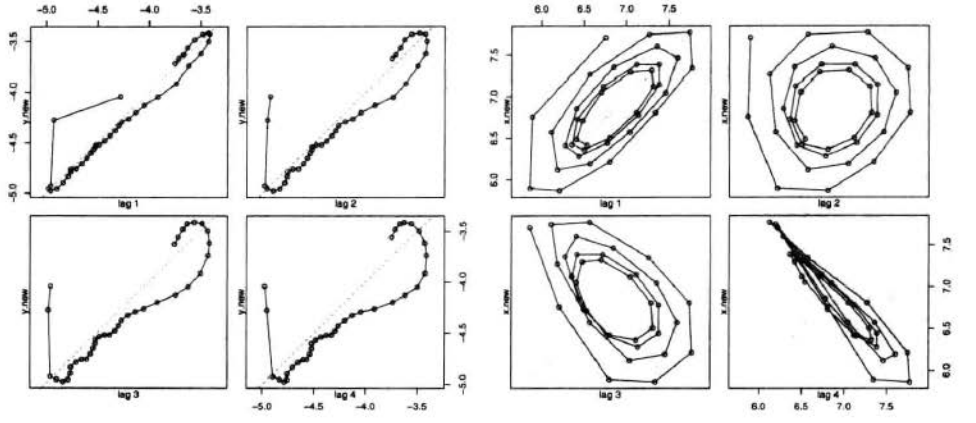


图 13.18 用局部线性AR模型预测40个值的相图

图13.18也显示出右边4个猞猁序列的图与左边的人造线性序列完全不同, 其左边4图对应例13.5序列, 右边4图对应例13.6序列.

一般的离散单变量(不一定是线性的)m阶时间序列$\{X_t\}_{t\in T}$ (称为NLAR(m), NonLinear AutoRegressive of order m)有下面的形式:

$$X_{t+s} = f(X_t, X_{t-d}, ..., X_{t-(m-1)d}; \boldsymbol{\theta}) + \epsilon_{t+s},$$

式中, $\{\epsilon_t\}_{t\in T}$为白噪声; 而f为$R^m \to R$的映射; m为嵌入维数(embedding dimension); d为滞后时间; s为预测步数. 向量$\boldsymbol{\theta}$为模型f的待估计参数, 需要通过观测序列$\{X_t\}_{t\in N}$来估计.

13.4.2 线性AR模型

前面已经简单讨论过线性时间序列, 作为前面一般序列特例的线性AR模型为:

$$\begin{aligned} X_{t+s} = \phi + \phi_1 X_t + \phi_2 X_{t-d} + \phi_3 X_{t-2d} + \cdots \\ + \phi_{m-1} X_{t-(m-2)d} + \phi_m X_{t-(m-1)d} + \epsilon_{t+s}. \end{aligned}$$

用线性AR模型($d = 1, m = 3$)拟合猞猁数据, 代码为($d = 1$为默认值):

```
library(tsDyn);x <- log10(lynx)
x.ar=linear(x, m=2);summary(x.ar)#这里的m相当于公式中的m-1
```

输出的系数估计为: $(\phi, \phi_1, \phi_2) = (1.0576005, 1.3842377, -0.7477757)$, 即估计的模型为:

$$X_t = 1.06 + 1.38X_{t-1} - 0.75X_{t-2} + \epsilon_t.$$

而且, $\sigma_\epsilon^2 = 0.05072$. 根据上述拟合模型, 预测未来10年的结果可用如下语句:

```
(x.ar.p=predict(x.ar, n.ahead=10))
```

输出下面10个数的预测数列:

```
Time Series:
Start = 1935
End = 1944
Frequency = 1
 [1] 3.384622 3.102350 2.821052 2.642745 2.606274
 [6] 2.689122 2.831076 2.965623 3.045717 3.055977
```

13.4.3 自门限自回归模型

自门限自回归模型(self threshold autoregressive model, SETAR)严格说来是分段线性模型, 当时间序列的值在不同的期间时, 用不同的线性AR模型来描述. 两段(一个门限参数)的SETAR模型为:

$$X_{t+s} = \begin{cases} \phi_{10} + \phi_{11}X_t + \cdots + \phi_{1mL}X_{t-(mL-1)d} + \epsilon_{t+s}, & Z_t \leqslant \text{th}; \\ \phi_{20} + \phi_{21}X_t + \cdots + \phi_{2mH}X_{t-(mH-1)d} + \epsilon_{t+s}, & Z_t > \text{th}. \end{cases}$$

式中, Z_t为门限变量(threshold variable); th为门限值. 在SETAR模型中Z_t可以是$\{X_t, X_{t-d}, ..., X_{t-(m-1)d}\}$中的一个. 可以通过门限滞后$0 \leqslant \delta \leqslant m-1$来定义$Z_t$, 使得

$$Z_t = X_{t-\delta d}.$$

还可以更灵活地定义Z_t为不同滞后的时间序列值的线性组合, 即

$$Z_t = \beta_1 X_t + \beta_2 X_{t-d} + \cdots + \beta_m X_{t-(m-1)d}.$$

类似地, 三段(两个门限参数)的SETAR模型为:

$$X_{t+s} = \begin{cases} \phi_{10} + \phi_{11}X_t + \cdots + \phi_{1mL}X_{t-(mL-1)d} + \epsilon_{t+s}, & Z_t \leqslant \text{th}_1; \\ \phi_{20} + \phi_{21}X_t + \cdots + \phi_{2mM}X_{t-(mM-1)d} + \epsilon_{t+s}, & \text{th}_1 < Z_t \leqslant \text{th}_2; \\ \phi_{30} + \phi_{31}X_t + \cdots + \phi_{3mH}X_{t-(mH-1)d} + \epsilon_{t+s}, & Z_t > \text{th}_2. \end{cases}$$

1. 一个门限参数的模型

我们可以用程序包tsDyn中的函数setar()实现用SETAR模型拟合猞猁数据, 可以选定$\delta = 1$(thDeley=1)及$m = 2$, 但两段模型的自回归阶数(mL和mH)必须在函数中确定, 但这也可以用另一个函数selectSETAR()来根据AIC选择. 语句为:

```
selectSETAR(x, m=2)
```

部分输出为:

```
   thDelay mL mH       th pooled-AIC
1        0  2  2 2.557507  -25.48913
2        0  2  2 2.587711  -24.43721
3        0  2  2 2.556303  -23.45258
4        0  2  2 2.553883  -23.28460
5        0  2  2 2.582063  -23.06773
```

因而, 可以选择mL=2, mH=2, 即使用下面的拟合语句:

```
x.setar=setar(x, m=2, mL=2, mH=2, thDelay=1)
summary(x.setar)
```

得到下面的输出:

```
Non linear autoregressive model
SETAR model ( 2 regimes)
Coefficients:
Low regime:
    phiL.1     phiL.2    const L
 1.2642793 -0.4284292  0.5884369

High regime:
   phiH.1    phiH.2   const H
 1.599254 -1.011575  1.165692

Threshold:
-Variable: Z(t) = + (0) X(t)+ (1)X(t-1)
-Value: 3.31
Proportion of points in low regime: 69.64% High regime: 30.36%
```

于是拟合的模型为:

$$X_{t+1}=\begin{cases}0.588+1.264X_t-0.428X_{t-1}+\epsilon_{t+1}, & X_{t-1}\leqslant 3.31;\\ 1.166+1.599X_t-1.012X_{t-1}+\epsilon_{t+1}, & X_{t-1}>3.31.\end{cases}$$

对于上述拟合模型, 预测未来10年的结果可用如下语句:

```
(x.setar.p=predict(x.setar, n.ahead=10))
```

输出下面10个数的预测数列:

```
Time Series:
Start = 1935
End = 1944
Frequency = 1
 [1] 3.348576 2.949075 2.494675 2.478933 2.653709
 [6] 2.881419 3.094429 3.266175 3.392051 3.477612
```

2. **两个门限参数的模型**

还可以用SETAR($\delta = 1, m = 2$, 三部分模型的自回归阶数均为2)来拟合猞猁数据(未进行参数选择), 代码为:

```
x.setar3=setar(x,m=2,mL=2,mH=2,mM=2,thDelay=1,nthresh=2)
summary(x.setar3)
```

得到下面输出:

```
Non linear autoregressive model
SETAR model ( 3 regimes)
Coefficients:
Low regime:
    phiL.1     phiL.2    const L
 1.3980502 -0.5729485  0.5729163

Mid regime:
    phiM.1     phiM.2    const M
 1.2149740 -0.6995948  1.5613169

High regime:
   phiH.1    phiH.2   const H
 1.599254 -1.011575  1.165692

Threshold:
-Variable: Z(t) = + (0) X(t)+ (1)X(t-1)+ (0)X(t-0)
-Value: 2.612 3.310
Proportion of points in low regime: 35.71%
Middle regime: 33.93%   High regime: 30.36%
```

这时, 拟合的模型为:

$$X_{t+1}=\begin{cases}0.573+1.398X_t-0.573X_{t-1}+\epsilon_{t+1}, & X_{t-1}\leqslant 2.612;\\ 1.561+1.215X_t-0.700X_{t-1}+\epsilon_{t+1}, & 2.612<X_{t-1}\leqslant 3.310;\\ 1.166+1.599X_t-1.012X_{t-1}+\epsilon_{t+1}, & X_{t-1}>3.310.\end{cases}$$

对于上述拟合模型, 预测未来10年的结果可用如下语句:

```
x.setar3.p=predict(x.setar3, n.ahead=10)
```

输出下面10个数的预测数列:

```
Time Series:
Start = 1935
End = 1944
Frequency = 1
 [1] 3.455664 3.289612 3.140546 3.075603 3.100984
 [6] 3.177257 3.252169 3.289825 3.283168 3.248737
```

3. Hansen检验

Hansen(1999)提出了零假设为线性模型, 备选假设为门限模型的利用自助法分布的检验. 对猞猁数据实行这个检验的代码如下(备选假设为两段和三段的门限AR模型, 因此实际上是两个检验):

```
Hansen.x=setarTest(lynx, m=1, nboot=1000)
summary(Hansen.x);plot(Hansen.x)
```

利用上面代码产生了和检验有关的图13.19及下面的输出:

```
Test of linearity against setar(2) and setar(3)

         Test  Pval
1vs2 20.05733 0.002
1vs3 42.81466 0.002

Critical values:
          0.9     0.95    0.975     0.99
1vs2 10.77135 12.43518 15.41961 16.84954
1vs3 20.71762 24.01091 26.37393 31.43030

SSR of original series:
               SSR
```

```
AR        137160013
SETAR(2) 116484232
SETAR(3)  99471266

Threshold of original series:
          th1  th2
SETAR(2) 3465   NA
SETAR(3) 1676 3465

Number of bootstrap replications:  1000
Asymptotic bound:  729.5347
```

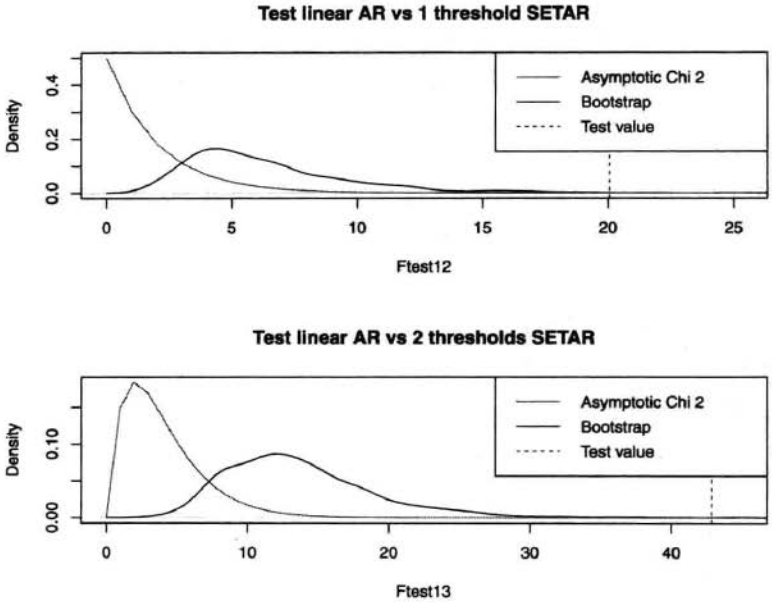


图 13.19 与Hansen检验有关的图

13.4.4 Logistic 平滑过渡自回归模型

Logistic 平滑过渡自回归模型(logistic smooth transition autoregressive model, LSTAR)为:

$$X_{t+s} = (\phi_{10} + \phi_{11}X_t + \cdots + \phi_{1L}X_{t-(L-1)d})(1 - G(Z_t, \gamma, \text{th})) + (\phi_{20} + \phi_{21}X_t + \cdots + \phi_{2H}X_{t-(H-1)d})G(Z_t, \gamma, \text{th}) + \epsilon_{t+s},$$

式中, $G(Z_t,\gamma,\text{th})$为logistic分布函数, 其位置参数(location)为th, 尺度参数(scale)为$1/\gamma$, 即

$$G(Z_t,\gamma,\text{th})=\frac{1}{1+\exp[-(Z_t-\text{th})\gamma]}=\frac{1}{2}+\frac{1}{2}\tanh\frac{(Z_t-\text{th})\gamma}{2},$$

其密度函数为:

$$f(Z_t,\gamma,\text{th})=\frac{\gamma\exp[-(Z_t-\text{th})\gamma]}{\{1+\exp[-(Z_t-\text{th})\gamma]\}^2}=\frac{\gamma}{4}\text{sech}^2\frac{(Z_t-\text{th})\gamma}{2}.$$

LSTAR模型的各个变量和SETAR类似. 给定th和γ后, 该模型也是线性的. 估计LSTAR模型时, 需要为所有待估计参数(ϕ,γ,th)确定初始值. 估计时先通过线性回归确定ϕ, 然后确定th, γ使得残差平方和最小, 重复这两步, 直到收敛.

用LSTAR模型拟合猞猁数据, 这里也有关于mL,mH参数选择的函数(当然, 用默认值也可以执行)selectLSTAR(), 还是确定m=2, thDelay=1. 用下面代码选择参数:

```
selectLSTAR(x, m=2)
```

得到输出:

```
  thDelay mL mH       AIC       BIC       th      gamma
1       1  2  1 -358.5196 -339.3662 3.379743   8.740303
2       1  2  2 -356.6440 -334.7544 3.344847  10.305586
3       1  1  1 -355.9970 -339.5798 3.585194   2.819021
4       1  1  2 -354.9086 -335.7552 4.272686   2.717408
5       0  2  2 -349.9262 -328.0366 2.567991 100.000109
```

根据上面输出的AIC和BIC值来做的判断并不一致. 这里还是根据AIC选择了mL=2,mH=1, 并用来建模. 代码如下:

```
x.lstar=lstar(x, m=2, mL=2,mH=1,thDelay=1)
summary(x.lstar)
```

得到输出:

```
Optimized values fixed for regime 2  :
    gamma=8.70099, th=3.380043; SSE=4.342669

Non linear autoregressive model

LSTAR model
Coefficients:
```

```
Low regime:
   const1    phi1.1    phi1.2
 0.450298  1.252161 -0.354821

High regime:
    const2     phi2.1
-2.1162244  0.4513928

Smoothing parameter: gamma = 8.701

Threshold
Variable: Z(t) = + (0) X(t) + (1) X(t-1)

Value: 3.38

Fit:residuals variance=0.03809,  AIC=-359, MAPE=5.597%

Non-linearity test of full-order LSTAR model
against full-order AR model
 F = 12.446 ; p-value = 1.3815e-05
```

拟合的模型为:

$$X_{t+1} = (0.450 + 1.252X_t - 0.355X_{t-1})(1 - G(X_{t-1}, 8.701, 3.38)) + (-2.116 + 0.451X_t)G(X_{t-1}, 8.701, 3.38) + \epsilon_{t+1}.$$

输出中还有一个相对于线性AR模型的F检验, p值为0.0000138, 说明这个模型比线性模型优越.

对于上述拟合模型, 预测未来10年的结果可用如下语句:

```
x.lstar.p=predict(x.lstar, n.ahead=10)
```

输出下面10个数的预测数列:

```
Time Series:
Start = 1935
End = 1944
Frequency = 1
 [1] 3.345630 2.909123 2.564024 2.612981 2.811630
 [6] 3.042702 3.257380 3.416870 3.426117 3.197871
```

13.4.5 神经网络模型

具有D个隐藏层节点(只考虑一个隐藏层)而且激活函数为g的神经网络模型(neural network model)定义为:

$$x_{t+s} = \beta_0 + \sum_{j=1}^{D} \beta_j g\left(\gamma_{0j} + \sum_{i=1}^{m} \gamma_{ij} x_{t-(i-1)d}\right),$$

在确定使用多少隐藏层节点时, 可用下面语句:

```
set.seed(8)
selectNNET(x, m=3, size=1:10)
```

得到输出:

```
   size       AIC       BIC
1     6 -372.2861 -287.4639
2     8 -357.9101 -245.7260
3     9 -351.2467 -225.3816
4     5 -348.6748 -277.5337
5     1 -336.8290 -320.4118
6     4 -336.7884 -279.3282
7     7 -335.7535 -237.2503
8     2 -326.8290 -296.7308
9     3 -316.8290 -273.0498
10   10 -246.8289 -107.2828
```

这里根据上面输出的AIC和BIC值所做的判断也不一致, 可以选择6个节点:

```
x.nnet=nnetTs(x, m=2,size=6);summary(x.nnet)
```

得到输出:

```
NNET time series model
a 2-6-1 network with 25 weights
options were - linear output units

Fit:
residuals variance = 0.04052,  AIC = -315, MAPE = 5.873%
```

当然, 对于神经网络模型完全没有必要写出拟合公式, 但可以用拟合的模型进行预测, 预测未来10年的结果可用以下代码:

```
(x.nnet.p=predict(x.nnet, n.ahead=10))
```

输出为:

```
Time Series:
Start = 1935
End = 1944
Frequency = 1
 [1] 3.391782 3.042936 2.701032 2.609172 2.757570
 [6] 2.991986 3.221451 3.377600 3.393032 3.241303
```

13.4.6 可加AR模型

可加AR模型(additive autoregressive model, AAM)是一个广义可加模型(generalized additive model, GAM), 定义为:

$$x_{t+s} = \mu + \sum_{i=1}^{m} s_i(x_{t-(i-1)d}),$$

式中, s_i为由处罚三次回归样条(penalized cubic regression splines)所表示的光滑函数. 拟合猞猁数据可以用下面语句:

```
x.aar=aar(x, m=3);summary(x.aar)
```

而对未来10年的预测则可用语句:

```
(x.aar.p=predict(x.aar, n.ahead=10))
```

输出为:

```
Time Series:
Start = 1935
End = 1944
Frequency = 1
 [1] 3.345630 2.909123 2.564024 2.612981 2.811630
 [6] 3.042702 3.257380 3.416870 3.426117 3.197871
```

13.4.7 模型的比较

可以通过AIC和MAPE(mean absolute percent error, 均方绝对误差率)在上述各种模型中挑选最好的, 代码如下(利用前面运算结果):

```
mod <- list()
mod[["linear"]]=x.ar
```

```
mod[["setar"]]=x.setar
mod[["setar3"]]=x.setar3
mod[["lstar"]]=x.lstar
mod[["nnetTs"]]=x.nnet
mod[["aar"]]=x.aar
sapply(mod, AIC);sapply(mod, MAPE)
```

输出为(上面是AIC, 下面是MAPE):

```
  linear     setar    setar3     lstar    nnetTs      aar
-333.8737 -358.3740 -357.5254 -358.5189 -315.4810 -320.5343
  linear     setar     setar3    lstar     nnetTs    aar
0.068019  0.056486 0.05443646 0.055974   0.058734 0.056662
```

从上面输出可以看出, LSTAR和SETAR3差不多. 一个AIC最小, 一个MAPE最小.

13.4.8 门限协整

1. 向量误差修正模型

分析非平稳时间序列时, 协整的概念是非常重要的, 前面介绍的是线性协整(linear cointegration), 其基本思想是, 即使两个或更多的变量是非平稳的, 也有可能存在它们的平稳的线性组合. 这些变量能够被解释为有一个稳定的关系, 即长远的平衡状态(a long-run equilibrium), 可以用向量的误差修正模型(vector error-correction model, VECM)来表示, 并共享一个共同的随机趋势.

然而, 这个定义意味着对长远平衡状态的微小偏移将导致立即的误差修正机制. 门限协整(threshold cointegration)推广了线性协整, 它允许只在超过临界门限时才调整, 即考虑了可能的交易花费和价格的粘滞; 也允许调整的非对称性, 即对正负偏移不以同样方式修正.

回顾前面的长期线性VECM(式(13.4)):

$$\Delta \boldsymbol{X}_t = \boldsymbol{\Gamma}_1 \Delta \boldsymbol{X}_{t-1} + \cdots + \boldsymbol{\Gamma}_{p-1} \Delta \boldsymbol{X}_{t-p+1} + \boldsymbol{\Pi} \boldsymbol{X}_{t-p} + \boldsymbol{\Upsilon} \boldsymbol{D}_t + \boldsymbol{\alpha} + \boldsymbol{u}_t,$$

其中, 矩阵$\boldsymbol{\Pi}$包含了累积的长期影响. 在$k = 2, p = 1$, 而且没有输入项的情况下, 上式也可以写成

$$\begin{bmatrix} \Delta X_{t1} \\ \Delta X_{t2} \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix} \begin{bmatrix} \Delta X_{t-1,1} \\ \Delta X_{t-1,2} \end{bmatrix} + \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} ECT_{-1} + \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} u_{t1} \\ u_{t2} \end{bmatrix},$$

式中, 误差修正项

$$ECT_{-1} = (1, -\beta)\begin{bmatrix} X_{t-1,1} \\ X_{t-1,2} \end{bmatrix}.$$

在门限自回归模型实际上是分段自回归模型时, 上面公式的误差修正项也分成相应的段. 以三段门限为例, 上式可转换成下面的门限VECM(threshold VECM, TVECM)形式, 其中, 最上面的式子相应于低段(用误差修正项的下标L标识), 中间的式子相应于中段(用误差修正项的下标M标识), 最下面的式子相应于高段(用误差修正项的下标H标识),

$$\begin{bmatrix} \Delta X_{t1} \\ \Delta X_{t2} \end{bmatrix} = \begin{cases} \begin{bmatrix} a_{1L} \\ a_{2L} \end{bmatrix} ECT_{L,t-1} + \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix} \begin{bmatrix} \Delta X_{t-1,1} \\ \Delta X_{t-1,2} \end{bmatrix} + \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} u_{t1} \\ u_{t2} \end{bmatrix}; \\ \begin{bmatrix} a_{1M} \\ a_{2M} \end{bmatrix} ECT_{M,t-1} + \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix} \begin{bmatrix} \Delta X_{t-1,1} \\ \Delta X_{t-1,2} \end{bmatrix} + \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} u_{t1} \\ u_{t2} \end{bmatrix}; \\ \begin{bmatrix} a_{1H} \\ a_{2H} \end{bmatrix} ECT_{H,t-1} + \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix} \begin{bmatrix} \Delta X_{t-1,1} \\ \Delta X_{t-1,2} \end{bmatrix} + \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} u_{t1} \\ u_{t2} \end{bmatrix}. \end{cases}$$

2. 向量误差修正模型的估计

这一类模型也完全可以使用程序包**tsDyn**中的函数来估计参数, 包括协整系数向量参数β的估计. 下面用该程序包所带的美国利率期限结构数据(zeroyld)来说明.

例 13.7　美国利率期限结构数据(zeroyld)　该数据共有从1947年到1991年的482个观测值及2个变量: 12个月短期利率期限(short.run)及120个月长期利率期限(long.run).

式(13.7)展示了用TVECM拟合该数据得到的结果模型, 式中第一行相应于th $\in (-\infty, -1.31]$, 第二行相应于th $\in (-1.31, 0.77]$, 第三行相应于th $\in (0.77, \infty)$.

$$\begin{bmatrix} \Delta X_{t1} \\ \Delta X_{t2} \end{bmatrix} = \begin{cases} \begin{bmatrix} 0.23 \\ 1.14 \end{bmatrix} ECT_{-1} + \begin{bmatrix} 0.56 \\ 1.95 \end{bmatrix} + \begin{bmatrix} 0.22 & -0.13 \\ 0.77 & 0.02 \end{bmatrix} \begin{bmatrix} \Delta X_{t-1,1} \\ \Delta X_{t-1,2} \end{bmatrix}; \\ \begin{bmatrix} -0.01 \\ 0.02 \end{bmatrix} ECT_{-1} + \begin{bmatrix} 0.01 \\ -0.01 \end{bmatrix} + \begin{bmatrix} 0.10 & 0.07 \\ 0.33 & 0.24 \end{bmatrix} \begin{bmatrix} \Delta X_{t-1,1} \\ \Delta X_{t-1,2} \end{bmatrix}; \\ \begin{bmatrix} -0.05 \\ 0.11 \end{bmatrix} ECT_{-1} + \begin{bmatrix} 0.07 \\ -0.09 \end{bmatrix} + \begin{bmatrix} -0.11 & 0.05 \\ 0.09 & -0.03 \end{bmatrix} \begin{bmatrix} \Delta X_{t-1,1} \\ \Delta X_{t-1,2} \end{bmatrix}. \end{cases} \tag{13.7}$$

使用TVECM还得到构成误差修正项ECT的协整向量: $(1, -\beta) = (1, -1.047592)$, 以及搜寻门限的图13.20(包括与线性门限值的比较).

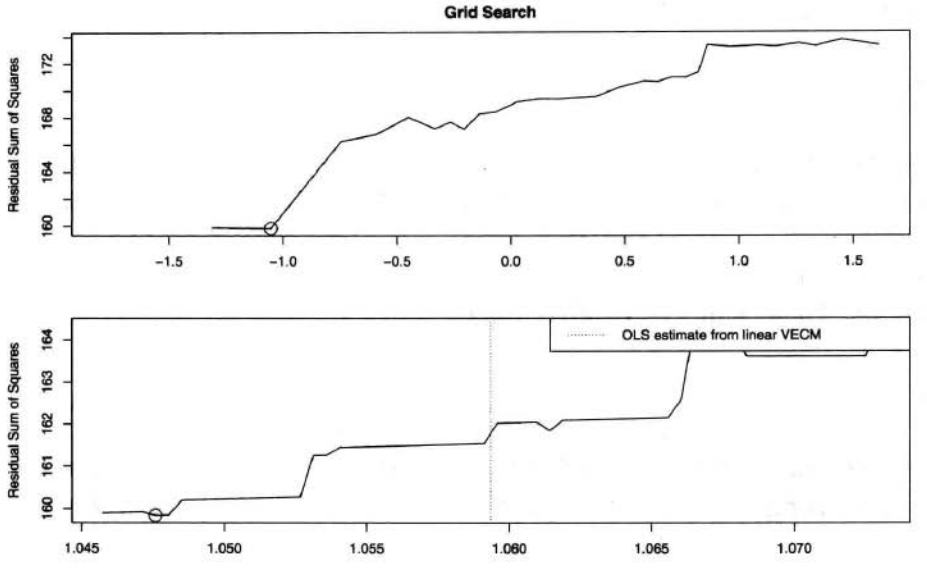


图 13.20 搜寻门限的图(包括与线性门限值的比较)

这里, 使用TVECM拟合例13.7所用的语句为:

```
data(zeroyld)
tvecm=TVECM(zeroyld, nthresh=2,lag=1, ngridBeta=60,
        ngridTh=30, plot=TRUE,trim=0.05)
summary(tvecm)
```

输出为:

```
#Model TVECM

Full sample size: 482   End sample size: 480
Number of variables: 2  Number of estimated parameters 24
AIC -2222.993   BIC -2114.475   SSR 155.5288

Cointegrating vector: (1, - 1.047592 )
$Bdown
                    ECT              Const       short.run t-1
Equation short.run 0.23(0.0209)  0.56(0.0042)  0.22(0.2194)
Equation long.run  1.14(1.1e-11) 1.95(2.2e-09) 0.77(0.0070)
                    long.run t-1
Equation short.run -0.1314(0.1646)
```

```
Equation long.run  0.0204(0.8946)

$Bmiddle
                    ECT              Const      short.run t-1
Equation short.run -0.01(0.7920) 0.01(0.6923)  0.10(0.3455)
Equation long.run  0.020(0.7283) -0.01(0.8530) 0.33(0.0580)
                    long.run t -1
Equation short.run 0.0690(0.2106)
Equation long.run  0.2368(0.0086)

$Bup
                    ECT              Const     short.run t -1
Equation short.run -0.05(0.4422) 0.07(0.4250) -0.11(0.2911)
Equation long.run  0.11(0.3008)  -0.09(0.5044) 0.09(0.6232)
                    long.run t -1
Equation short.run 0.0464(0.5253)
Equation long.run  -0.0287(0.8097)

Threshold
Values: -1.312 0.766
Percentage of Observations in each regime 6.7% 56.9% 36.5%
```

3. 向量误差修正模型的检验

这里检验的零假设为线性协整(linear cointegration), 备选假设为门限协整(threshold cointegration). 该检验来自Hansen and Seo (2002). 该检验应用了回归和残差自助法抽样.

对于例13.7美国利率期限结构数据(zeroyld), 如果取自助法抽样数目为100, 该检验代码为:

```
z=TVECM.HStest(zeroyld, lag=1, intercept=TRUE, nboot=100)
summary(z);plot(z)
```

得到与检验统计量有关的图13.21及下面输出.

```
# Test of linear versus threshold cointegration
    of Hansen and Seo(2002)

Test Statistic: 20.59942
```

```
    (Maximized for threshold value: -0.04805437)
P-Value:         0.05      ( Fixed regressor bootstrap)

Critical values:
     0.90%    0.95%    0.99%
 19.15006 20.44568 22.28197
Number of bootstrap replications:         100

Cointegrating value(estimated under restricted linear model):
    -1.022065
```

这个输出不仅给出了检验的p值0.05, 而且给出了协整值$\beta = -1.022065$. 这说明可能存在门限协整.

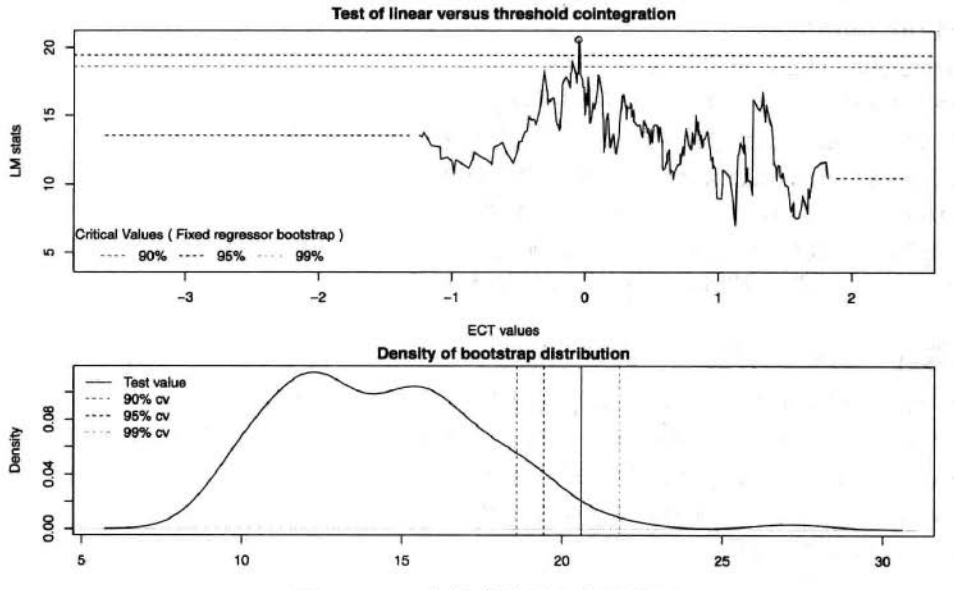


图 13.21 与检验统计量有关的图

附录 练习: 熟练使用R软件

实践1(最初几步):

```
x=1:100#把1,2,...,100这个整数向量赋值到x
(x=1:100) #同上, 只不过显示出来
sample(x,20) #从1,2,...,100中随机不放回地抽取20个值作为样本
set.seed(0);sample(1:10,3)#先设随机种子再抽样
#从1,2,...,200000中随机不放回地抽取10000个值作为样本:
z=sample(1:200000,10000)
z[1:10]#方括号中为向量z的下标
y=c(1,3,7,3,4,2)
z[y]#以y为下标的z的元素值
(z=sample(x,100,rep=T))#从x有放回地随机抽取100个值作为样本
(z1=unique(z))
length(z1)#z中不同元素的个数
xz=setdiff(x,z) #x和z之间的不同元素--集合差
sort(union(xz,z))#对xz及z的并的元素从小到大排序
setequal(union(xz,z),x) #xz及z的并的元素与x是否一样
intersect(1:10,7:50) #两个数据的交
sample(1:100,20,prob=1:100)#从1:100中不等概率随机抽样,
#上一语句各数字被抽到的概率与其值大小成比例
```

实践2(一些简单运算):

```
pi*10^2 #能够用?"*"来看基本算术运算方法, pi是圆周率
"*"(pi,"^"(10,2)) #和上面一样, 有些烦琐, 是吧! 没有人这么用
pi*(1:10)^-2.3#可以对向量求指数幂
x = pi * 10^2 x print(x) #和上面一样
(x=pi *10^2) #赋值带打印
pi^(1:5) #指数也可以是向量
print(x, digits= 12)#输出x的12位数字
```

实践3(关于R对象的类型等):

```
x=pi*10^2
class(x) #x的class
typeof(x) #x的type
class(cars)#cars是一个R中自带的数据
typeof(cars) #cars的type
```

```
names(cars)#cars数据的变量名字
summary(cars) #cars的汇总
head(cars)#cars的头几行数据, 和cars[1:6,]相同
tail(cars) #cars的最后几行数据
str(cars)#也是汇总
row.names(cars) #行名字
attributes(cars)#cars的一些信息
class(dist~speed)#公式形式,"~"左边是因变量,右边是自变量
plot(dist~speed,cars)#两个变量的散点图
plot(cars$speed,cars$dist) #同上
```

实践4(包括简单自变量为定量变量及定性变量的回归):

```
ncol(cars);nrow(cars) #cars的行列数
dim(cars) #cars的维数
lm(dist ~ speed, data = cars)#以dist为因变量,speed为自变量做OLS
cars$qspeed =cut(cars$speed, breaks=quantile(cars$speed),
    include.lowest = TRUE) #增加定性变量qspeed, 四分位点为分割点
names(cars) #数据cars多了一个变量
cars[3]#第三个变量的值, 和cars[,3]类似
table(cars[3])#列表
is.factor(cars$qspeed)
plot(dist ~ qspeed, data = cars)#点出箱线图
#拟合线性模型(简单最小二乘回归):
(a=lm(dist ~ qspeed, data = cars))
summary(a)#回归结果(包括一些检验)
```

实践5(简单样本描述统计量):

```
x <- round(runif(20,0,20), digits=2)#四舍五入
summary(x) #汇总
min(x);max(x) #极值, 与range(x)类似
median(x)  # 中位数(median)
mean(x)    # 均值(mean)
var(x)     #方差(variance)
sd(x)      # 标准差(standard deviation),为方差的平方根
sqrt(var(x)) #平方根
rank(x)    # 秩(rank)
order(x)#升幂排列的x的下标
order(x,decreasing = T)#降幂排列的x的下标
```

```
x[order(x)] #和sort(x)相同
sort(x)      #同上: 升幂排列的x
sort(x,decreasing=T)#sort(x,dec=T) 降幂排列的x
sum(x);length(x)#元素和及向量元素个数
round(x) #四舍五入,等于round(x,0),而round(x,5)为留到小数点后5位
fivenum(x)  # 五数汇总, quantile
quantile(x) # 分位点 quantile (different convention)有多种定义
quantile(x, c(0,.33,.66,1))
mad(x) # "median average distance":
cummax(x)#累积最大值
cummin(x)#累积最小值
cumprod(x)#累积积
cor(x,sin(x/20)) #线性相关系数 (correlation)
```

实践6(简单图形):

```
x=rnorm(200)#将200个随机正态数赋值到x
hist(x, col = "light blue")#直方图(histogram)
rug(x) #在直方图下面加上实际点的大小
stem(x)#茎叶图
x <- rnorm(500)
y <- x + rnorm(500) #构造一个线性关系
plot(y~ x) #散点图
a=lm(y~x) #做回归
abline(a,col="red")#或者abline(lm(y~x),col="red")散点图加拟合线
print("Hello World!")
paste("x 的最小值= ", min(x)) #打印
demo(graphics)#演示画图(点击Enter来切换)
```

实践7(复数运算和求函数极值):

```
(2+4i)^-3.5+(2i+4.5)*(-1.7-2.3i)/((2.6-7i)*(-4+5.1i))#复数运算
#下面构造一个10维复向量, 实部和虚部均为10个标准正态样本点:
(z <-complex(real=rnorm(10), imaginary =rnorm(10)))
complex(re=rnorm(3),im=rnorm(3))#3维复向量
Re(z) #实部
Im(z) #虚部
Mod(z) #模
Arg(z) #辐角
choose(3,2) #组合
```

```
factorial(6)#排列6!
#解方程:
f=function(x) x^3-2*x-1
uniroot(f,c(0,2))#迭代求根
#如果知道根为极值
f=function(x) x^2+2*x+1 #定义一个二次函数
optimize(f,c(-2,2))#在区间(-2,2)内求极值
```

实践8(字符型向量和因子型变量):

```
a=factor(letters[1:10]);a #letters:小写字母向量,LETTERS:大写
a[3]="w" #不行! 会给出警告
a=as.character(a) #转换一下
a[3]="w" #可以了
a;factor(a) #两种不同的类型
#定性变量的水平:
levels(factor(a))
sex=sample(0:1,10,r=T)
sex=factor(sex);levels(sex)
#改变因子的水平:
levels(sex)=c("Male","Female");levels(sex)
#确定水平次序:
sex=ordered(sex,c("Female","Male"));sex
levels(sex)
```

实践9(数据输入输出):

```
x=scan()#屏幕输入, 可键入或粘贴, 多行输入在空行后按Enter键
1.5 2.6 3.7 2.1 8.9 12 -1.2 -4

x=c(1.5,2.6,3.7,2.1,8.9,12,-1.2,-4)#等价于上面代码
w=read.table(file.choose(),header=T)#从列表中选择有变量名的数据
setwd("f:/mydata")#建立工作路径
(x=rnorm(20))#给x赋值20个标准正态数据值
#(注:有常见分布的随机数、分布函数、密度函数及分位数函数)
write(x,"test.txt")#把数据写入文件(路径要对)
y=scan("test.txt");y#扫描文件数值数据到y
y=iris;y[1:5,];str(y) #iris是R自带数据
write.table(y,"test.txt",row.names=F)#把数据写入文本文件
w=read.table("test.txt",header=T)#读带有变量名的数据
```

```
str(w) #汇总
write.csv(y,"test.csv")#把数据写入csv文件
v=read.csv("test.csv")#读入csv数据文件
str(v) #汇总
data=read.table("clipboard")#读入剪贴板的数据
```

实践10(序列等):

```
(z=seq(-1,10,length=100))#从-1到10等间隔的100个数组成的序列
z=seq(-1,10,len=100)#和上面写法等价
(z=seq(10,-1,-0.1)) #10到-1间隔为-0.1的序列
(x=rep(1:3,3))  #三次重复1:3
(x=rep(3:5,1:3)) #自己看，这又是什么呢?
x=rep(c(1,10),c(4,5))
w=c(1,3,x,z);w[3]#把数据(包括向量)组合(combine)成一个向量
x=rep(0,10);z=1:3;x+z #向量加法(如果长度不同，R给出警告和结果)
x*z#向量乘法
rev(x)#颠倒次序
z=c("no cat","has ","nine","tails") #字符向量
z[1]=="no cat" #双等号为逻辑等式
z=1:5
z[7]=8;z #什么结果? 注:NA为缺失值(not available)
z=NULL
z[c(1,3,5)]=1:3;
z
rnorm(10)[c(2,5)]
z[-c(1,3)]#去掉第1、3元素
z=sample(1:100,10);z
which(z==max(z))#给出最大值的下标
```

实践11(矩阵):

```
x=sample(1:100,12);x #抽样
all(x>0);all(x!=0);any(x>0);(1:10)[x>0]#逻辑符号的应用
diff(x) #差分
diff(x,lag=2) #差分
x=matrix(1:20,4,5);x #矩阵的构造
x=matrix(1:20,4,5,byrow=T);x#矩阵的构造，按行排列
t(x) #矩阵转置
x=matrix(sample(1:100,20),4,5)
```

```
2*x
x+5
y=matrix(sample(1:100,20),5,4)
x+t(y) #矩阵之间相加
(z=x%*%y) #矩阵乘法
z1=solve(z) #用solve(a,b)可以解方程ax=b
z1%*%z #应该是单位向量，但浮点运算不可能得到干净的0
round(z1%*%z,14)  #四舍五入
b=solve(z,1:4); b #解联立方程
```

实践12(矩阵续):

```
nrow(x);ncol(x);dim(x)#行列数目
x=matrix(rnorm(24),4,6)
x[c(2,1),]#第2和第1行
x[,c(1,3)] #第1和第3列
x[2,1] #第[2,1]元素
x[x[,1]>0,1] #第1列大于0的元素
sum(x[,1]>0) #第1列大于0的元素的个数
sum(x[,1]<=0) #第1列不大于0的元素的个数
x[,-c(1,3)]#没有第1、3列的x.
diag(x)  #x的对角线元素
diag(1:5) #以1:5为对角线元素,其他元素为0的对角线矩阵
diag(5) #5维单位矩阵
x[-2,-c(1,3)]#没有第2行，第1、3列的x
x[x[,1]>0&x[,3]<=1,1]#第1列>0并且第3列<=1的第1列元素
x[x[,2]>0|x[,1]<.51,1]#第1列<.51或者第2列>0的第1列元素
x[!x[,2]<.51,1]#第1列中相应于第2列>=.51的元素
apply(x,1,mean)#对行(第一维)求均值
apply(x,2,sum)#对列(第二维)求和
x=matrix(rnorm(24),4,6)
x[lower.tri(x)]=0;x #得到上三角阵,
#为得到下三角阵，用x[upper.tri(x)]=0)
```

实践13(高维数组):

```
x=array(runif(24),c(4,3,2));x
#上面用24个服从均匀分布的样本点构造4乘3乘2的三维数组
is.matrix(x)
dim(x)#得到维数(4,3,2)
```

```
is.matrix(x[1,,])#部分三维数组是矩阵
x=array(1:24,c(4,3,2))
x[c(1,3),,]
x=array(1:24,c(4,3,2))
apply(x,1,mean)#可以对部分维做求均值运算
apply(x,1:2,sum)#可以对部分维做求和运算
apply(x,c(1,3),prod)#可以对部分维做求乘积运算
```

实践14(矩阵与向量之间的运算):

```
x=matrix(1:20,5,4) #5乘4矩阵
sweep(x,1,1:5,"*")#把向量1:5的每个元素乘到每一行
sweep(x,2,1:4,"+")#把向量1:4的每个元素加到每一列
x*1:5
#下面把x标准化,即每一元素减去该列均值,除以该列标准差
(x=matrix(sample(1:100,24),6,4));(x1=scale(x))
(x2=scale(x,scale=F))#自己观察并总结结果
(x3=scale(x,center=F))#自己观察并总结结果
round(apply(x1,2,mean),14)#自己观察并总结结果
apply(x1,2,sd)#自己观察并总结结果
round(apply(x2,2,mean),14);apply(x2,2,sd)#自己观察并总结结果
round(apply(x3,2,mean),14);apply(x3,2,sd)#自己观察并总结结果
```

实践15(缺失值, 数据的合并):

```
airquality #有缺失值(NA)的R自带数据
complete.cases(airquality)#判断每行有没有缺失值
which(complete.cases(airquality)==F) #有缺失值的行号
sum(complete.cases(airquality)) #完整观测值的个数
na.omit(airquality) #删去缺失值的数据
#附加, 横或竖合并数据: append,cbind,rbind
x=1:10;x[12]=3
(x1=append(x,77,after=5))
cbind(1:5,rnorm(5))
rbind(1:5,rnorm(5))
cbind(1:3,4:6);rbind(1:3,4:6) #去掉矩阵重复的行
(x=rbind(1:5,runif(5),runif(5),1:5,7:11))
x[!duplicated(x),]
unique(x)
```

实践16(list):

```
#list可以是任何对象(包括list本身)的集合
z=list(1:3,Tom=c(1:2,a=list("R",letters[1:5]),w="hi!"))
z[[1]];z[[2]]
z$T
z$T$a2
z$T[[3]]
z$T$w
```

实践17(条形图和表):

```
x=scan()#30个顾客在五个品牌中的挑选
3 3 3 4 1 4 2 1 3 2 5 3 1 2 5 2 3 4 2 2 5 3 1 4 2 2 4 3 5 2

barplot(x) #不合题意的图
table(x) #制表
barplot(table(x)) #正确的图
barplot(table(x)/length(x)) #比例图(和上图形状一样)
table(x)/length(x)
```

实践18(形成表格):

```
library(MASS)#载入程序包MASS
quine #MASS所带数据
attach(quine)#把数据变量的名字放入内存
#下面语句产生从该数据得到的各种表格
table(Age)
table(Sex, Age); tab=xtabs(~ Sex + Age, quine); unclass(tab)
tapply(Days, Age, mean)
tapply(Days, list(Sex, Age), mean)
detach(quine) #attach的逆运行
```

实践19(如何写函数):

```
#下面这个函数是按照定义(编程简单, 但效率不高)求n以内的素数
ss=function(n=100){z=2;
for (i in 2:n)if(any(i%%2:(i-1)==0)==F)z=c(z,i);return(z) }
fix(ss) #用来修改任何函数或编写一个新函数
ss() #计算100以内的素数
t1=Sys.time() #记录时间点
```

```
ss(10000) #计算10000以内的素数
Sys.time()-t1 #计算费了多少时间
system.time(ss(10000))#计算执行ss(10000)所用时间
#函数可以不写return,这时最后一个值为return的值
#为了输出多个值最好使用list输出
```

实践20(画图):

```
x=seq(-3,3,len=20);y=dnorm(x)#产生数据
w= data.frame(x,y)#合并x,成为数据w
par(mfcol=c(2,2))#准备画四个图的地方
plot(y ~ x, w,main="正态密度函数")
plot(y ~ x,w,type="l", main="正态密度函数")
plot(y ~ x,w,type="o", main="正态密度函数")
plot(y ~ x,w,type="b",main="正态密度函数")
par(mfcol=c(1,1))#取消par(mfcol=c(2,2))
```

实践21(色彩和符号等的调整):

```
plot(1,1,xlim=c(1,7.5),ylim=c(0,5),type="n") #画出框架
#在plot命令后面追加点(如要追加线可用lines函数):
points(1:7,rep(4.5,7),cex=seq(1,4,l=7),col=1:7, pch=0:6)
text(1:7,rep(3.5,7),labels=paste(0:6,letters[1:7]),
    cex=seq(1,4,l=7),col=1:7)#在指定位置加文字
points(1:7,rep(2,7), pch=(0:6)+7)#点出符号7到13
text((1:7)+0.25, rep(2,7), paste((0:6)+7))#加符号号码
points(1:7,rep(1,7), pch=(0:6)+14) #点出符号14到20
text((1:7)+0.25, rep(1,7), paste((0:6)+14)) #加符号号码
#关于符号形状、大小、颜色以及其他画图选项的说明可用"?par"来查看
```

参考文献

[1] Agrawal, R., Imielinski, T., and Swami, A. (1993) Mining associations between sets of items in large databases. *ACM SIGMOD Int'l Conf. on Management of Data,* Washington D.C., May 1993.

[2] Ambler, G. and Royston, P. (2001) Fractional polynomial model selection procedures: Investigation of type I error rate. *Journal of Statistical Simulation and Computation*, 69: 89 - 108.

[3] Agresti, A. (2002) *Categorical Data.* Second edition. Wiley.

[4] Arenas, C. and Cuadras, C.M. (2002) Some recent statistical methods based on distances. *Contributions to Science*, 2, 183 - 191.

[5] Badsberg, J.H. (1995) *An Environment for Graphical Models.* PhD thesis, Aalborg University.

[6] Baltagi, B. H. and Pinnoi, N. (1995) Public capital stock and state productivity growth: Further evidence. *Empirical Economics,* 20: 351 - 359.

[7] Baltagi, Badi H. (2003) *Econometric Analysis of Panel Data.* 2nd ed. John Wiley and Sons.

[8] Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations (with discussion). *Journal of the Royal Statistical Society*, B, 26: 211 - 252.

[9] Breiman, L. (2001) Statistical modeling: The two cultures. *Statistical Science,* Vol. 16, No. 3: 199-231.

[10] Breiman, L. (2001) Random Forests. *Machine Learning*, 45(1): 5-32.

[11] Breslow, N.E. (1984) Extra-Poisson variation in log-linear models. *Applied Statistics,* 33: 38 - 44.

[12] Brockwell, P. J. and Davis, R. A. (1991) *Time Series and Forecasting Methods.* Second edition. Springer. Series G (page 557).

[13] Buehlmann, P. and Hothorn, T. (2007) Boosting algorithms: Regularization, prediction and model fitting (with discussion). *Statistical Science,* 22(4): 477-505.

[14] Buntine, W. (1991) Theory refinement of Bayesian networks. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 52 - 60, 1991.

[15] Campbell, M. J.and Walker, A. M. (1977) A survey of statistical work on the Mackenzie River series of annual Canadian lynx trappings for the years 1821 - 1934 and a new analysis. *Journal of the Royal Statistical Society series,* A, 140: 411 - 431.

[16] Chambers, John M. and Hastie, Trevor J. eds. (1992) *Statistical Models in S.* Wadsworth and Brooks/Cole, Pacific Grove, CA 1992, pp. 46 - 47.

[17] Chatfield, C. (2004) *The Analysis of Time Series.* 6th ed. Chapman & Hall/CRC.

[18] Cleveland, R. B., Cleveland, W. S., McRae, J.E. and Terpenning, I. (1990) STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics,* 6: 3 - 73.

[19] Cooper, G. F. and Herskovits, E. (1992) A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309 - 348, 1992.

[20] Cox, D. and Oakes, D. (1984) *Analysis of Survival Data.* Chapman & Hall, London.

[21] *Consumer Reports,* April, 1990, pp. 235 - 288.

[22] Cornwell, C. and Rupert, P. (1988) Efficient estimation with panel data: An empirical comparison of instrumental variables estimators. *Journal of Applied Econometrics,* 3: 149 - 155.

[23] de Campos, Luis M. (2006) A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research*, 7: 2149-2187.

[24] Diaconis, P. and Efron, B. (1983) Occam's two razors: The sharp and the blunt. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining* (R Agrawai and P. Stolorz, eds.) 37-43, AAAI Press, Menlo Park, CA.

[25] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004) Least angle regression. *Annals of Statistics*, 32: 407 - 451.

[26] Flake, G.W. and Lawrence, S. (2002) Efficient SVM regression training with SMO.*Machine Learning*, 46: 271-290.

[27] Fleming, T. and Harrington, D. (1991) *Counting Processes and Survival Analysis*. Wiley, New York.

[28] Fox, J. and Andersen, R. (2006) Effect displays for multinomial and proportional-odds logit models. *Sociological Methodology*, 36: 225 - 255.

[29] Frank, A. & Asuncion, A. (2010) UCI machine learning repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[30] Friedman, J. (2008) Fast sparse regression and classification. *Technical report*. Standford University.

[31] Garthwaite, P.H. (1994) An interpretation of partial least squares. *J. Amer. Statist. Assoc.*, 89: 122-127.

[32] Glass, D.V. (1954) *Social Mobility in Britain*. Glencoe: Free Press.

[33] Golub, G.H., Heath, M. and Wahba, G. (1979) Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics,* Vol. 21, No. 2: 215-223.

[34] Granger, C.W.J. (1969) Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37: 424-438.

[35] Greene, W.H. (2003) *Econometric Analysis*. 5th edition. Upper Saddle River, NJ: Prentice Hall.

[36] Griffiths, T.L. and Yuille, A. (2006) A primer on probabilistic inference. In *Trends in Cognitive Sciences*. Supplement to special issue on Probabilistic Models of Cognition, 10(7): 1 - 11.

[37] Hansen (1999). Testing for linearity. *Journal of Economic Surveys,* Vol. 13, No. 5, December 1999, pp. 551-576(26). Available at: http://www.ssc.wisc.edu/ bhansen/papers/cv.htm.

[38] Hansen, B. and Seo, B. (2002) Testing for two-regime threshold cointegration in vector error-correction models. *Journal of Econometrics,* 110: 293 - 318.

[39] Hastie, T., Tibshirani, R. and Friedman, J. (2001) *The Elements of Statistical Learning.* Springer-Verlag.

[40] Heckerman, D., Geiger, D. and Chickering, D. M. (1995) Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197 - 243.

[41] Hedeker, D. and Gibbons, R. D. (1994) A random-effects ordinal regression model for multilevel analysis. *Biometrics,* 50: 933 - 944.

[42] Herriges, J. A. and Kling, C. L. (1999) Nonlinear income effects in random utility models. *Review of Economics and Statistics,* 81: 62-72.

[43] Hogg, R. V., and Ledolter, J. (1992) *Applied Statistics for Engineers and Physical Scientists.* Second Edition. Macmillan, New York. Exercise 1.5-6.

[44] Irigoien, I. and Arenas, C. (2008) INCA: New statistic for estimating the number of clusters and identifying atypical units. *Statistics in Medicine*, 27(15): 2948 - 2973.

[45] Jérôme Pagès. Multiple Factor Analysis: Main features and application to sensory data. http://factominer.free.fr/docs/PagesAFM.pdf.

[46] Johansen, S. and Juselius, K. (1990) Maximum likelihood estimation and inference on cointegration - with applications to the demand for money. *Oxford Bulletin of Economics and Statistics,* 52, 2: 169 - 210.

[47] Jordan, M.I., Ghahramani, Z., Jaakkola, T.S. and Saul, L.K. (1998) An introduction to variational methods for graphical models. In *Learning in Graphical Models.* M.I. Jordan, ed. Kluwer Academic Publishers, Dordrecht.

[48] Kalbfleisch, J. D. and Prentice, R. L. (1980) *The Statistical Analysis of Failure Time Data.* John Wiley & Sons.

[49] Koenker, R. and Bassett, G. (1982) Robust tests of heteroscedasticity based on regression quantiles. *Econometrica*, 50: 43 - 61.

[50] Lambert, D. (1992) Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics*, 34: 1-14.

[51] Lauritzen, S.L. and Spiegelhalter, D.J. (1988) Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society.* Series B, 50(2): 157 - 224.

[52] Lê, S., Josse, J. and Husson, F. (2008) FactoMineR: An R package for multivariate analysis. *Journal of Statistical Software,* 25 (1): 1-18.

[53] Lê, S., Josse, J. and Husson, F. (2011) *Exploratory Multivariate Analysis by Example using R.* Chapman & Hall/CRC.

[54] Ledolter, J. and Tiao, G. C. (1979) Statistical methods for ambient air pollutants with special reference to the Los Angeles Catalyst Study (LACS) Data. *Environmental Science and Technology,* 13: 1233-1240.

[55] Massart, D. L., Vandeginste, B. G. M., Buydens, L. M. C., de Jong, S., Lewi, P. J., Smeyers-Verbeke, J. (1998) *Handbook of Chemometrics and Qualimetrics: Part B.* Elsevier. Tables 35.1 and 35.4.

[56] Mullahy, J. (1986) Specification and testing of some modified count data models. *Journal of Econometrics*, 33: 341-365.

[57] Munnel, A. (1990) Why has productivity growth declined? Productivity and public investment. *New England Economic Review,* 3 - 22.

[58] Murtaugh, P.A., Dickson, E.R., Van Dam, G.M., Malinchoc, M. Grambsch, P.M., Langworthy, A.L., Gips, C.H. (1994) Primary biliary cirrhosis: Prediction of short-term survival based on repeated patient visits. *Hepatology*, 20(1.1): 126-34.

[59] Nolan, D. and Speed, T. (2000) *Stat Labs: Mathematical Statistics through Applications.* Springer.

[60] Pearl, J. (1987) Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence*, 32(2): 245 - 258.

[61] Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems.* Morgan and Kaufman, San Mateo.

[62] Pearl, J. and Russel, S. (2001) Bayesian networks. Report (R-277), November 2000. In *Handbook of Brain Theory and Neural Networks.* M. Arbib ed. MIT Press, Cambridge, pp. 157 - 160.

[63] R Development Core Team (2011). R: A language and environment for statistical computing. *R Foundation for Statistical Computing,* Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

[64] Reisby, N., Gram, L. F., Bech, P., Nagy, A., Petersen, G. O., Ortmann, J., et al. (1977) Imipramine: Clinical effects and pharmacokinetic variability. *Psychopharmacology,* 54: 263 - 272.

[65] Ripatti, S. and Palmgren, J. (2000) Estimation of multivariate frailty models using penalized partial likelihood. *Biometrics,* 56:1016-1022.

[66] Rizopoulos, D., Verbeke, G. and Lesaffre, E. (2009) Fully exponential Laplace approximations for the joint modelling of survival and longitudinal data. *Journal of the Royal Statistical Society,* B, 71: 637-654.

[67] Sela, Rebecca J., and Simonoff, Jeffrey S.(2011) RE-EM Trees: A data mining approach for longitudinal and clustered data. *Machine Learning.*

[68] Smyth, G. K. (1989) Generalized linear models with varying dispersion. *J. R. Statist. Soc.,* B, 51: 47 - 60.

[69] Therneau, T. and Grambsch, P. (2000) *Modeling Survival Data: Extending the Cox Model.* Springer-Verlag, New York.

[70] Therneau, T., Grambsch, P. and Pankratz,V.S. (2003) Penalized survival models and frailty. *J Computational and Graphical Statistics,* 12:156-175.

[71] Tukey, J. (1962) The future of data analysis. *Annals of Mathematical Statistics,* 33 (1): 13.

[72] Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S.* 4th ed. Springer.

[73] Verma, T., and Pearl, J. (1988) Influence diagrams and d-separation, UCLA cognitive systems laboratory. *Technical Report,* 880052 (R-101), March 1988.

[74] Wold, S., Wold, H., Dunn, W.J. and Ruhe, A.(1984) The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM J. Sci. Stat. Comput.*, 5: 735-743.

[75] Wheaton, B., Muthén, B., Alwin, D.F. and Gene, F. S. (1977) Assessing reliability and stability in panel models. In *Sociological methodology*. ed. D. Heise, San Francisco: Jossey-Bass.

[76] Wulfsohn, M. and Tsiatis, A. (1997) A joint model for survival and longitudinal data measured with error. *Biometrics,* 53: 330-339.

[77] Zhang, N.L. and Poole, D. (1996) Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5: 301 - 328.

高等院校研究生用书

TEXTBOOKS FOR GRADUATES

管理研究方法：原理与应用	刘　军
会计研究方法论	吴　溪
实证审计理论	陈汉文　韩洪灵
公司财务实证研究：重点文献导读	陆正飞　岳　衡　祝继高
财务会计与资本市场实证研究：重点文献导读	陆正飞　姜国华　张　然
会计信息与证券投资实证研究：重点文献导读	陆正飞　姜国华　张　然
实证会计理论的研究与进展：经典文献赏析	许家林　王昌锐　胡　伟
现代统计分析方法与应用（第三版）	何晓群
时间序列分析：方法与应用	易丹辉
结构方程模型：方法与应用	易丹辉
复杂数据统计方法——基于R的应用（第三版）	吴喜之

人大经管图书在线　http://www.rdjg.com.cn
了解图书出版信息　下载教学辅助资料

ISBN 978-7-300-21
9 787300 215945
定价：36.00元